

ऊर्जा संतुलन तकनीकों के माध्यम से कुपोषण मार्कर आकलन
के लिए निर्णय समर्थन प्रणाली का विकास

Development of Decision Support System for
Malnutrition Marker Assessment through Energy
Balance Techniques

द्वारा/BY

वनजा वी

VANAJA V

मास्टर ऑफ़ साइंस
MASTER OF SCIENCE

इन/IN

कंप्यूटर एप्लीकेशन
COMPUTER APPLICATION



भाकृअनुप-भारतीय कृषि सांख्यिकी अनुसंधान संस्थान
ICAR-INDIAN AGRICULTURAL STATISTICS RESEARCH
INSTITUTE

भाकृअनुप-भारतीय कृषि अनुसंधान संस्थान
ICAR-INDIAN AGRICULTURAL RESEARCH INSTITUTE

नई दिल्ली - 110012

NEW DELHI – 110012

2022

ऊर्जा संतुलन तकनीकों के माध्यम से कुपोषण मार्कर आकलन
के लिए निर्णय समर्थन प्रणाली का विकास

**Development of Decision Support System for
Malnutrition Marker Assessment through Energy
Balance Techniques**

द्वारा/BY

वनजा वी
VANAJA V

Thesis submitted to the Faculty of Post-Graduate School,
ICAR-Indian Agricultural Research Institute, New Delhi,
In partial fulfillment of the requirements for the degree of

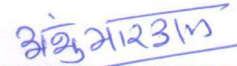
**MASTER OF SCIENCE
IN
COMPUTER APPLICATIONS**

Approved by:

Chairperson: Dr. Mukesh Kumar



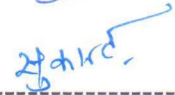
Co-Chairperson: Dr. Anshu Bharadwaj



Member: Dr. Shashi Bhushan Lal



Member: Dr. Sukanta Dash



Member: Dr. Pratibha Joshi





भाकृअनुप-भारतीय कृषि सांख्यिकी अनुसंधान संस्थान
लाइब्रेरी एवेन्यू, पूसा, नई दिल्ली-110012
ICAR-INDIAN AGRICULTURAL STATISTICS RESEARCH INSTITUTE
Library Avenue, Pusa, New Delhi-110012



डॉ. मुकेश कुमार
प्रधान वैज्ञानिक
Dr. Mukesh Kumar
Principal Scientist

CERTIFICATE

*This is to certify that the work incorporated in the thesis entitled “**Development of Decision Support System for Malnutrition Marker Assessment through Energy Balance Techniques**” submitted in partial fulfillment of the requirement for the degree of **MASTER OF SCIENCE** in **COMPUTER APPLICATIONS** of the P. G. School, ICAR-Indian Agricultural Research Institute, New Delhi is a record of bona fide research work carried out by **Ms. Vanaja V**, Roll No. 21441, under my guidance and supervision and no part of this dissertation has been submitted for any other degree or diploma.*

All assistance and help received during the course of this investigation has been properly acknowledged by her.

Place: New Delhi
Date: 07 October, 2022


(Mukesh Kumar)
Chairperson
Advisory committee

ACKNOWLEDGEMENTS

This thesis is the fruitful outcome of the knowledge gained over the entire period of my M.Sc. study at ICAR-Indian Agricultural Statistics Research Institute (I.A.S.R.I), New Delhi during which I have been in touch with a great number of people whose contribution in varied yet myriad ways has led to the research and making of the thesis which deserves special mention. It is a pleasure to convey my gratitude to all of them by way of my humble acknowledgements.

First and foremost with reverence, I want to express deepest sense of gratitude to **Dr. Mukesh Kumar**, Principal Scientist, Division of Computer Application, ICAR-Indian Agricultural Statistics Research Institute (I.A.S.R.I), New Delhi and Chairman of my Advisory Committee for his initiative, benevolence, endurance, constructive criticism and constant monitoring during the period of my investigation and also during preparation of this thesis. Above all and most needed, he provided me constant support and encouragement in various ways. I consider myself blessed to have the privilege of being guided by him. I am really indebted to him.

It is great privilege for me to express my esteem and profound sense of gratitude to **Dr. Anshu Bharadwaj**, Principal Scientist, Division of Computer Application, ICAR-I.A.S.R.I, New Delhi and Member of my Advisory Committee for her valuable suggestions and help.

I would also like to express my sincere thanks to **Dr. Shashi Bhushan Lal**, Principal Scientist, Division of Computer Application, ICAR-I.A.S.R.I and Member of my Advisory Committee. I am really grateful to him for the support and encouragement he provided during research work.

I would also like to express my sincere thanks to **Dr. Sukanta Dash**, Scientist, Discipline of Agricultural Statistics, ICAR-I.A.S.R.I and Member of my Advisory Committee. I am really grateful to him for the support and encouragement he provided during the course of my studies.

I would also like to express my sincere thanks to **Dr. Pratibha Joshi**, Scientist, Division of Agricultural Extension, ICAR-I.A.S.R.I and Member of my Advisory Committee. I am really grateful to her for the support and encouragement she provided during the whole research work. I will always be thankful to her for immense support for making me understand the concepts clearly.

I am deeply indebted to **Dr. Alka Arora**, Professor, Division of Computer Application, ICAR-I.A.S.R.I, New Delhi for the help and guidance provided by her.

I am deeply indebted to **Dr. Rajender Parsad**, Director, ICAR-I.A.S.R.I for the help and infrastructures provided by him.

I am thankful to my all batchmates **Pavana, Vivek, Akash, Tanvi, Chandini, Soutrik, Madhu, Kabilan, Ankith, Rabsan, Bappa, Devendra and Anitha** for their friendly approach and moral support throughout my entire M.Sc. tenure.

I do not have any words to express the help and support given by my seniors and Subhashis sir for all their affectionate support and help.

I take this opportunity to appreciate the help rendered by the staff of TAC, ICAR-I.A.S.R.I Library for their help.

I am indebted **Dr. A.K. Singh**, Director ICAR-I.A.R.I, New Delhi and Dean and Joint Director (Education), ICAR-I.A.R.I, for providing facilities to carry out this research work.

I like to thank all the staff of PG School for their helpful attitude and cooperation, throughout the period of study. Last but not the least I am thankful to ICAR for the financial assistance provided to me in the form of Fellowship during the tenure of my study. Finally, I would like to thank everybody who was imperative to the successful realization of this thesis, as well as articulate my apology that I could not mention personally one by one.

Date: 07 october, 2022

Place: New Delhi-110012



(Vanaja V)

CONTENTS

Chapter	Title	Page. No
I.	Introduction	1-5
1.1	Background	1-2
1.2	Anthropometry	2-3
1.3	Energy Balance	3
1.4	Agriculture and Information Technology (IT)	3-4
1.5	Decision Support System (DSS)	4
1.6	Motivation and Problem Identification	4-5
1.7	Objectives	5
II.	Review of Literature	6-10
2.1	General Perspective	6
2.2	Previous Works Done	6
2.2.1	Anthropometry and Energy Balance Techniques for Nutritional Status	6-8
2.2.2	Mobile Apps/DSS in Agriculture	8-10
III.	Materials and Methods	11-29
3.1	Application Architecture	11-12
3.1.1	Client-Side Interface Layer	12-13
3.1.2	Application Layer	13-14
3.1.3	Database Layer	14

3.1.4	Architecture of Application	15
3.1.5	Work Flow of the DSS	15
3.2	Programming Environments	16
3.2.1	eXtensible Markup Language (XML)	16
3.2.2	Java Programming Language	16-17
3.3	Application Development Environments	17
3.3.1	Android Studio IDE	17
3.3.2	Eclipse IDE	17-18
3.3.3	Spring Boot Tool	18
3.4	Physical Fitness Index (PFI)	19
3.5	VO ₂ Max	19-20
3.6	Body Mass Index (BMI)	20
3.7	Mid Upper Arm Circumference (MUAC)	21
3.8	Calf Circumference (CC)	21
3.9	Skin Fold Thickness	22
3.10	Minimum Dietary Diversity	23
3.11	Total Energy Expenditure (TEE)	23-24
3.12	Resting Energy Expenditure (REE)	24
3.13	Basal Metabolic Rate (BMR)	25
3.14	Integration of Existing Application NutriGuide	25
3.15	Database Designing	25
3.15.1	MySQL	25-26

3.15.2	Database Structure	26-29
3.16	Apache Tomcat Server	29
IV.	Results	30-44
4.1	Features of the Application	30
4.2	User Interface of DSS	31
4.2.1	Home Page	31
4.2.2	User Registration	32
4.2.3	User sign In	33
4.2.4	Farmer Registration and Information Download	34
4.2.4.1	Registration of New Farmer	34
4.2.4.2	Download Farmers Details	35
4.2.5	Physical Fitness	36
4.2.5.1	Physical Fitness Index (PFI)	37
4.2.5.2	VO ₂ Max	38
4.2.6	Anthropometric Measures	39
4.2.6.1	Body Mass Index (BMI)	39
4.2.6.2	Mid Upper Arm Circumference (MUAC)	40
4.2.6.3	Calf Circumference (CC)	40
4.2.6.4	Skin Fold Parameters	41
4.2.7	Minimum Dietary Diversity	41
4.2.8	Energy Balance Techniques	42
4.2.8.1	Basal Metabolic Rate (BMR)	42

4.2.8.2	Resting Energy Expenditure (REE)	43
4.2.8.3	Total Energy Expenditure (TEE)	43
4.2.9	Result	44
V.	Discussion	45-46
VI.	Summary and Conclusion	47
	Abstract	
	सार	
	Bibliography	i- iv
	Appendix	i-xxxiii

LIST OF TABLES

Table No.	Title	Page No.
1	Range of PFI	19
2	Relative VO ₂ Max range	20
3	BMI Classification	20
4	MUAC Classification	21
5	CC Classification	21
6	Body Density Equations	22
7	Different Food Groups	23
8	Pal Values for Different Activities	24
9	User Details	26
9	Farmer Details	26
10	PFI Details	27
11	VO ₂ Max Details	27
12	BMI Details	27
15	MUAC Details	27
16	CC Details	28
17	Skinfold Thickness Details	28
18	BMR Details	28
19	REE Details	28
20	TEE Details	29

21	MDD Details	29
----	-------------	----

LIST OF FIGURES

Figure No.	Title	Page No.
1	Android operating system architecture	12
2	Android activity life cycle	13
3	ER Diagram of the Database	14
4	3-tier Architecture of the Application	15
5	Work Flow of the DSS	15
6	Splash screen and home page	31
7	User registration	32
8	Login page	33
9	Farmer registration page	34
10	Downloaded farmers' details excel sheet snapshot	35
11	Moving to Calculation screen	36
12	Calculation of PFI	37
13	Calculation of VO ₂ Max	38
14	Calculation of BMI	39
15	Categorizing MUAC	40
16	Categorizing CC	40
17	Calculation of skinfold thickness	41
18	Calculation of Minimum Dietary Diversity	41
19	Calculation of BMR	42
20	Calculation of REE	43
21	Calculation of TEE	43

22	Snapshot of result page of particular farmer	44
----	--	----

CHAPTER I

INTRODUCTION

1.1 Background

The growth of the country's human resources requires adequate nutrition. It is the cornerstone of human development, especially for women, adolescents, and children, and it contributes to socioeconomic advancement, the eradication of poverty, and the security of the country's health. The health issues affecting children and women are influenced by the interconnected social, biological, and cultural factors. In India, one in three women of reproductive age are underweight, with a Body Mass Index of less than 18.5 kg/m² (BMI). We are all aware that undernutrition during pregnancy and childbirth perpetuates the intergenerational cycle of hunger (UNICEF). A fundamental cornerstone of any individual's health is adequate nutrition, which is especially critical for women because inadequate nutrition not only affects women's own health but also the health of their children. Children of malnourished women are more likely to face ill health throughout their lives thus making vicious cycle of malnutrition.

Malnutrition is a multifaceted issue that necessitates multisector responses. The results demonstrate that enhancing agricultural productivity can have a favourable effect on nutritional outcomes. India is experiencing a severe malnutrition crisis, according to the Global Nutrition Report 2018. By reducing affordability disparities for food and critical commodities like wheat, rice, and sugar in methods like the Public Distribution System, Governmental Schemes, and initiatives for nutritional security in India for everyone.

Malnutrition refers to deficiencies, excesses, or imbalances in a person's energy and/or nutrient intake. The term malnutrition covers two broad groups of conditions. One is 'undernutrition'—which includes stunting (low height for age), wasting (low weight for height), underweight (low weight for age) in the children with age group of 1-5 years, or chronic energy deficiency in women and/or micronutrient deficiencies or insufficiencies (a lack of important vitamins and minerals). The other is overweight, obesity, and diet-related non-communicable diseases (such as heart disease, stroke, diabetes, and cancer) (WHO). In this brief, the term malnutrition will refer to conditions of nutritional deficiency, including undernutrition and micronutrient deficiencies, though malnutrition actually also relates to problems of nutritional

excess (Rae Galloway 2002). Nutritional status can be assessed using many screening methods including 24 hour recall method of food intake, food frequency questionnaire, energy consumption etc. In the preliminary screening method, anthropometric measurements are one among them, where other methods are time consuming. Anthropometric measures such as body mass index (BMI) and waist circumference are widely used to assess nutritional status. In addition to that, the assessment of 24-hour energy expenditure (24-EE, in kcal/ d) is a requirement for establishing caloric prescriptions for patients. Energy expenditure can be determined by indirect calorimetric measurements like total energy expenditure (TEE), resting energy expenditure (REE) and basal metabolic rate (BMR). These are also called energy balance techniques.

Mobile-based information and modelling systems for the purpose of aiding the decision-making activities, helping to understand the problem, exploring various alternative courses for actions, and their impacts, and facilitating sensitivity analysis. The mobile-based applications provide the information to stakeholders on time and at minimum cost which helps in making timely decisions about their problems. DSS can be used to make timely decisions. A Decision Support System (DSS) is an application that assists managerial decision-makers in solving semi-structured and unstructured problems using data and models. The proposed DSS is an idea for nutrition and health assessment using anthropometry and energy balance techniques which can be used anywhere, at any number, and at any time.

1.2 Anthropometry

There are several screening methods for evaluation of nutritional status, which include biochemical, clinical, dietary assessment and anthropometric measurements. Anthropometry is study of proportions and measurements of human body. It is an inexpensive, non-invasive and less time-consuming method of assessing nutritional and health status of individual. It provides information on different component of body structure(especially muscular and fat components). Body mass index (BMI) and Mid upper arm circumference (MUAC) are widely used and significant anthropometric measurements. BMI is a good indicator and used to know a person's weight is normal for his/her height. Low BMI indicates under-weight for particular height. BMI alone cannot be used as nutrition indicator.

On the other, MUAC is also an important malnutrition screening tool and independent of height and weight. It indicates the arm muscle and sub-cutaneous fat, which are important to determine the survival in starvation and fat and fat free mass. It requires less equipment and easy to perform. It is a good predictor of mortality in children. Other anthropometric measures like calf

circumference and skinfold parameters can also be used to indicate malnutrition. Calf circumference is related to muscle mass of whole body. Skinfold thickness is measured at four different positions (biceps, triceps, subscapular and suprailiac) using skinfold calliper. It is used to access the body fat from the total body density and measure sub-cutaneous fat.

The use of anthropometry allows for the detection of nutritional status and the direction of care and therapy. It is possible to estimate a population's nutritional status, which can help with programme and policy decision-making.

1.2 Energy Balance

Energy balance is simple ($\text{Energy Balance} = \text{Energy Intake} - \text{Energy Expenditure}$), accurate and reliable measurements. It is used especially if the measurements are time-consuming and labour intensive. Energy requirements can be estimated using total energy expenditure (TEE). TEE is the amount of energy used by human body in one day adjusted to the amount of activity (sedentary, moderate or very active). Three components to TEE in humans are basal metabolic rate (BMR), thermic effect of food and the energy expenditure of activity. BMR is the amount of energy required by the body at rest to perform life-sustaining functions. It accounts for 60 percent of total daily energy expenditure and predicted by lean body mass. Thermic effect of food is increase in energy expenditure associated with digestion, absorption and storage, which accounts for 10percent of BMR. Resting energy expenditure (REE) is energy spent by the body during non-active period. It represents 60-70 percent of TEE (Total Energy Expenditure). Decisions about the adequate dietary intake can be taken by studying the energy requirements.

1.3 Agriculture and Information Technology (IT)

Now a days, computers have become integrated part of agriculture and related activities. Technological advancements have had a significant impact on the agri-production system, which was originally intended for other industries such as construction (Zhang *et al*, 2002). Computers are used to perform various activities (like monitoring, assessment, and information management) and to analyse large amounts of data, so that it will be easy to make decisions based on previous responses. Computers helps in communications, where large amount of data of transferred in less time. Programs have been created to mimic human logic by IT experts, these projects are called Decision Support Systems. To meet the demands of ever-increasing global population, agriculture has grown over time. However, current traditional methods will not be able to meet the

requirements as the estimated population by 2050 will be 9.8 billion. Therefore, we need IT in agriculture, so that we can combat future challenges. It will make easy for researchers to look into innovations, which may help the farmers and helps in assuring long-term viability of agriculture.

1.5 Decision Support System (DSS)

DSS stands for "Decision Support System (DSS)", coined by Gorry and Mortan and was first used in 1971. Each term has its own definition, which is as follows: (Sharda et al, 1988).

- Decision: It emphasises decision-making power in various scenarios.
- Support: Providing computer-based support necessitates computer-assisted decision making with sufficient “structure”.
- System: emphasising and assisting with various sorts of issue solving, as well as making relevant recommendations in the appropriate setting.

Three basic elements constituting a DSS architecture are:

- The stored database
- The working model
- The Graphical User Interface (GUI)

A Decision support system is a class of computerized systems that support decision-making activities. It is a decision-supporting interactive computer system that performs tasks by using data and models. A tool known as a computer-based modelling system, a decision support system is used to help decision-making processes (in visualisation and analysis of a problem, suggesting alternative activities, and analysing their effects on the system). It helps the stakeholders in taking timely decisions with minimum cost, which can be employed anywhere, at any number, and at any time. Decision Support Systems are either computerized or with human interaction, or a combination of both.

1.6 Motivation and Problem Identification:

Several DSS has been developed in different agricultural activities and for nutritional assessment but there is no integrated system for evaluating anthropometric data, mapping nutritional status, and energy requirements. In this context, the development of DSS is being proposed for Malnutrition Assessment Using Energy Balance Techniques. The proposed DSS would help to assess the nutritional and energy expenditure status using different anthropometric measures and energy balance techniques. The dietitians, nutritionists, or researchers can use it according to their convenience and can easily fetch or get their required information regarding nutritional status and

energy expenditure. It also helps to advise the farmer on how to enhance their diet to improve their physical condition.

1.7 Objectives

- (i) To design and develop Decision Support System for Malnutrition Assessment Using Energy Balance Techniques
- (ii) To integrate existing Application “Nutri Guide”
- (iii) To validate developed DSS with sample data

CHAPTER II

REVIEW OF LITERATURE

2.1 General Perspective

Mobile phones are one of the most often used devices in today's world. Nowadays, everyone uses a mobile phone, including both children and adults. They are incredibly helpful and help us in many different ways. Mobile phones make life more convenient and easier for us every day.

Different anthropometric and energy balance approaches can be utilized to acquire the results and categorize the physical health condition, and a mobile-based decision support system can be used to evaluate the nutritional status and energy expenditure of different genders and age groups. We will also receive general meal recommendations from this application for optimal nutrition.

2.2 Previous Work done

2.2.1 Review of Literature on Anthropometry and Energy Balance Techniques for Nutritional Status

Durnin and Womersley (1974) conducted a study on 481 men and women, whose age ranges from 16-72 years. Total body fat was assessed from the body density and its estimation from skinfold thickness. Skinfold thickness measurements were taken on four positions of the body (biceps, triceps, subscapular, and suprailliac). It was concluded that skinfold thickness have linear relationship with body density. For the estimation of the body density, Linear regression equations were also computed.

Mifflin *et al.* (1990) derived predictive equations for resting energy expenditure (REE) from data on 498 healthy subjects (including females (n = 247) and males (n = 251)), aged 19-78 y (45 ± 14 y, $I \pm SD$) and normal-weight (n = 264) and obese (n = 234) individuals. REE was measured by indirect calorimetry. To derive relationships between REE and weight, height, and age for both men and women ($R^2 = 0.71$) multiple-regression analyses were employed.

Wang *et al.* (2000) stated that for quantifying body size and proportions anthropometry is a simple reliable method. Anthropometrics can predict regional fat and skeletal fat along with total body fat. Equations for predicting body fat percent include more than 19 skinfold positions, 17 circumference sites, 11 width sites, and 9 length sites. The parameters that influence the anthropometric predictions are sex, age ethnicity, and site. Repeatability of 2% for circumference

measurements and 10% for skinfold measurements are necessary for anthropometrists to get certified.

Carbone *et al.* (2012) states that Long-term negative energy balances reduce body mass because both skeletal muscle and fat mass are lost. Numerous negative effects, such as a lowered basal metabolic rate, a slower protein turnover, a decline in physical performance, and an increased risk of injury are linked to reductions in skeletal muscle mass. In reaction to a negative energy balance, skeletal muscle mass decreases because of unbalanced rates of muscle protein synthesis and breakdown. The physiological mechanisms causing the loss of skeletal muscle during energy restriction are not well understood, yet. Recent research has shown that dietary protein intake over the current daily recommended intake of 0.8 g of protein per kilogram of body weight may reduce the loss of skeletal muscle mass by altering the intracellular control of muscle anabolism and proteolysis.

Hand *et al.* (2013) Four hundred and thirty (218 women and 212 men) completed all baseline measurements. There were statistically significant differences by sex uncovered for most anthropomorphic, physiological, and behavioural variables. Self-reported weight changes (mean \pm SD) over the previous year were 0.92 ± 5.24 kg for women and 21.32 ± 6.1 kg for men. Resting metabolic rate averages by sex were 2.88 ± 0.35 ml/kg/min for women and 3.05 ± 0.33 ml/kg/min for men. Concluded that Resting metabolic rates well below the standard estimate suggest that the evaluation of the components of total energy expenditure will be impactful for our understanding of the roles of energy intake and expenditure on changes in energy utilization and storage.

Bhattacharya *et al.* (2019) conducted a study to determine a new Composite Score using anthropometric measurements. Then compared the composite score with other methods like body mass index (BMI) and mid-upper arm circumference (MUAC) classification, to test the significance of the method. Anthropometric data from 780 adult Oraon (Male = 387, Female = 393) labourers of Alipurduar district of West Bengal, India were collected. Conventional methods (like BMI and MUAC) were used to assess the nutritional status. In the results it was found that the frequency of undernutrition was 45.9% according to the BMI category, 56.7% according to the MUAC category, and 51.8% according to the newly computed Composite Score. Concluded that by calculating the Composite Score of the anthropometric measurements nutritional status in the population can be identified more correctly and it is non-invasive.

Tonder *et al.* (2019) studied to expand on the limited South African malnutrition prevalence data

and investigate the viability of mid-upper-arm circumference (MUAC) as a malnutrition screening tool. A cross-sectional, multi-center, descriptive design was used. Direct measurements of BMI and MUST were not obtained in 38% and 43% of patients respectively, whilst MUAC was 100% obtainable. MUAC cut-offs for undernutrition at <23 cm (BMI < 16 kg/m²) and <24 cm (BMI < 18.5 kg/m²), respectively were determined, for the study of population groups. Concluded that MUAC is a feasible method to predict adult malnutrition.

Ross *et al.* (2020) studied that Visceral adipose tissue (VAT) is a main target for lifestyle-based treatments intended to reduce or manage health risk due to the undeniable link between it and cardiometabolic risk. Furthermore, strong evidence supports the notion that liver fat (LF) is positively linked to an elevated health risk and that its reduction is linked to a better metabolic profile. Less is known about the separate relationships between decreased VAT, LF, and cardiometabolic risk. In this narrative review, we present the evidence that suggests whether a negative energy balance brought on by either an increase in energy expenditure (aerobic activity) or a decrease in energy intake (hypocaloric diet) are useful techniques for lowering both VAT and LF.

2.2.2 Review of Literature for Mobile Apps or Decision Support System

C-DAC, Hyderabad (2016) launched a mobile app named ‘Dietary Guidelines for Indians’, which translates the scientific nutrition knowledge into dietary advice based on Recommended Dietary Allowances (RDA) for Indians given by the National Institute of Nutrition. The app includes 15 guidelines and its explanation, helps a person to lead a healthy life by following the guidelines.

ICMR – National Institute of Nutrition, Hyderabad (2017) has developed country’s first ‘nutritional atlas’. Nutritional atlas gives data and information related to nutritional status in groups of population at state and national levels. It provides a bird-eye view of deficiencies related to nutrition, disorders and commonness in various parts of the nation. It acts like a information management tool and helps the users to get time-wise trends on parameters like undernutrition, obesity, overweight and other diseases (communicable and non-communicable).

National Institute of Nutrition – Bio-Informatics Centre (BMIC,2017) developed an online platform named “Count What you Eat”, which provides a pathway to healthy life. It is a user-friendly and gives information about nutrition for practical use at users disposal.

ICMR – National Institute of Nutrition, Hyderabad (2018) launched a mobile app named “Nutrify India Now”. It acts as a Nutri-guide or a companion that would help you assess the nutrients one gets from the foods one consumes. The app helps the users keep track of diet and activity, providing nutritional information on Indian foods, and Recommended Dietary Allowance (RDA and food consumption pattern).

Mobile application for Ergonomics Assessment of Drudgery Prone Activities in Agriculture (Sonkusale L., 2018, M.Sc. Thesis Dissertation): developed a mobile app named “Ergon” to evaluate the ergonomics of drudgery-inducing agricultural tasks. The app helps the farmers/users in assessing, quantification and reducing drudgery in various farm operations.

Kumari L. (2019) created a smartphone app for designing and evaluating farm implements and machinery based on the Anthropometrics dimension of agricultural workers. The application also quantifies the drudgery level with physiological ergonomics parameters.

Abhishek M P *et al.* (2019) developed a Decision Support System for Evaluating Postural Assessment of Agricultural Activities using the Rapid Upper Limb Assessment (RULA) Technique. In this DSS improved body postures are suggested during the different agricultural activities for preventing the development of the musculoskeletal disorder.

Mobile Based Decision Support System for evaluation of postural ergonomics using Rapid Entire Body Assessment [Sahana, 2020, M.Sc. Thesis Dissertation]: developed a mobile-based decision support system for the evaluation of postural ergonomics using the rapid entire body assessment (REBA) technique. By using this mobile application, physiological parameters and posture can be evaluated and farmers may be suggested/recommended the correct posture to avoid developing musculoskeletal disorders.

Kundu *et al.* (2021) developed a Decision Support System for Biomechanical Evaluation of Agricultural Activities using Human Physical Drudgery Index (HPDI). It is designed to evaluate how tedious the work is and offer suggestions to make it less tedious.

Mobile Based Decision Support System for Nutrition and Health Assessment Using Anthropometry Techniques [Sarkar, 2021, M.Sc. Thesis Dissertation]: To monitor the farmers' nutritional status using several anthropometric methodologies, the "Nutri Guide" mobile

application for Android has been created. This programme not only aids in determining one's level of physical fitness and nutritional condition, but it also gives users a brief summary of the benefits of various food types for various age groups. This Android-based mobile application was developed using the Android Studio development environment, the Java programming language, eXtensible Markup Language for application user-end development, and a MySQL server as a backend database to store the farmers' details data and the various index values of all the farmers.

CHAPTER III

MATERIALS AND METHODS

This chapter covers the architectural design, necessary methods and tools, and practical strategies for creating the mobile application. The three main focuses of this chapter are the application architecture, programming environment, and software development environments. The application was developed using the Java programming language, XML, Eclipse IDE, Spring boot framework to create REST API, JDBC API, Android Studio IDE, and My SQL server.

3.1 Application Architecture

These days, there are many different types of mobile phones available, each with a unique operating system. In actuality, several brands use a range of operating systems. For Nokia devices, Symbian is the operating system of choice, followed by iOS for Apple products, Bada for Android. The most widely used, favoured, and user-friendly of all of them now is Android. It is therefore frequently used all around the world. It is heavily dependent on open-source technologies, which offer a broad range of capabilities and utility needed for the creation of a variety of applications.

The operating system based on the Linux operating system is the Android operating system (OS). Java, a high-level programming language, was employed to construct a user-friendly interface that is primarily used with it. The main components are Linux kernel, libraries, Android libraries, Android Runtime, Application Framework, and Applications (Alatalo et al, 2002).

It's essential to have a solid understanding of Android's design in order to create mobile apps. The majority of apps are created in Java and then compiled into bytecode using the environment that comes with Android Studio. When an app is installed on a device, the Android Runtime (ART) converts the bytecode to the domestic format used by the CPU. According to Brahler, the main objectives of the Android architecture are effectiveness and efficiency, both in terms of the execution and implementation of applications.

A variety of software inventory apps, services, operating systems, runtime environments, middleware, libraries, and a number of stack layers make up Android. Each stack layer's pertinent components are closely coordinated with one another.

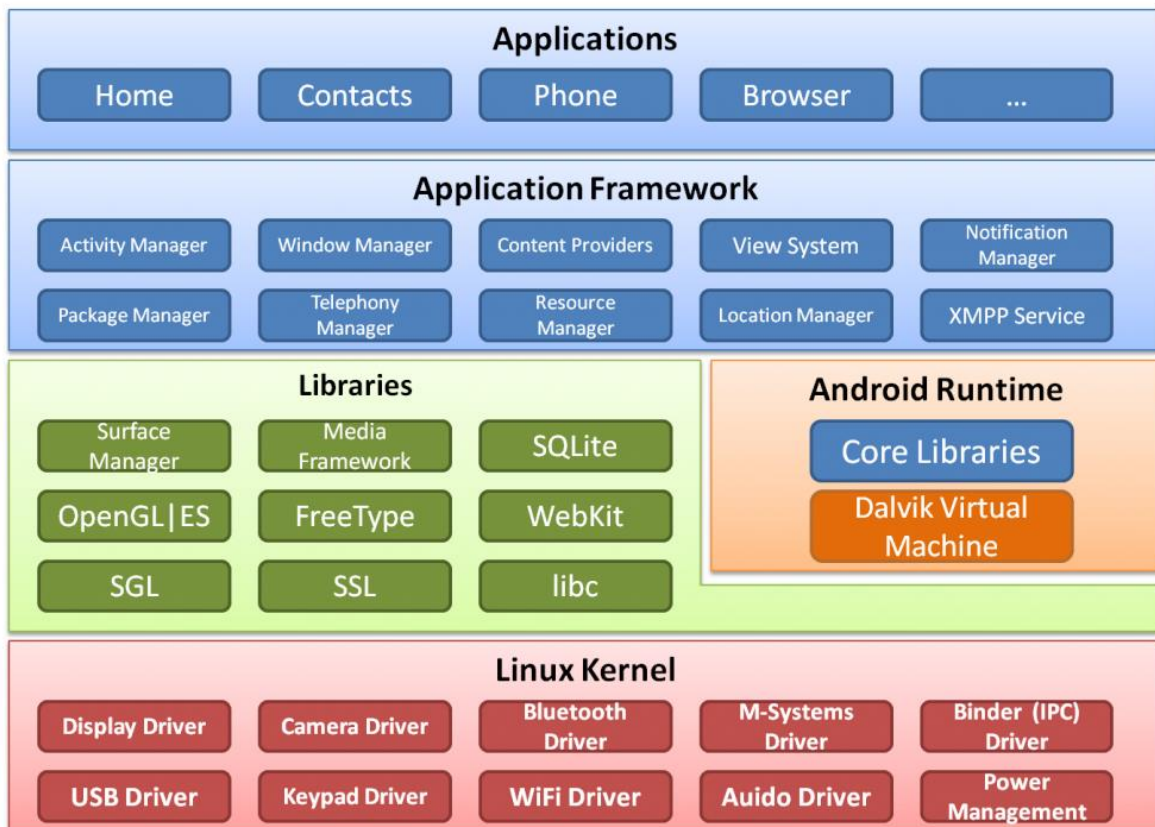


Fig 1: Android operating system architecture

3.1.1 Client-Side Interface Layer

The user interface of the application that interacts with users directly is called the client-side interface layer. The Java programming language, XML, and the Styles, Menu, Drawable, and Values components of the Android Software Development Kit were used to construct this layer (SDK). The client-side layer, which is the most important part of the application, views every screen of the programme as an activity for the user. These operations and the application's logic are implemented using the JAVA programming language.

An activity's lifespan is shown in Figure 3. The layouts of such activities are implemented using XML, which is also utilised to create the application. You may create better layouts by using the XML files Styles, Menu, drawable, and Values. A list of items in a Dropdown view, items in a Radio group view, buttons, an Image View, and a Video View are among the additional features of the CSIL. It also has forms for user input.

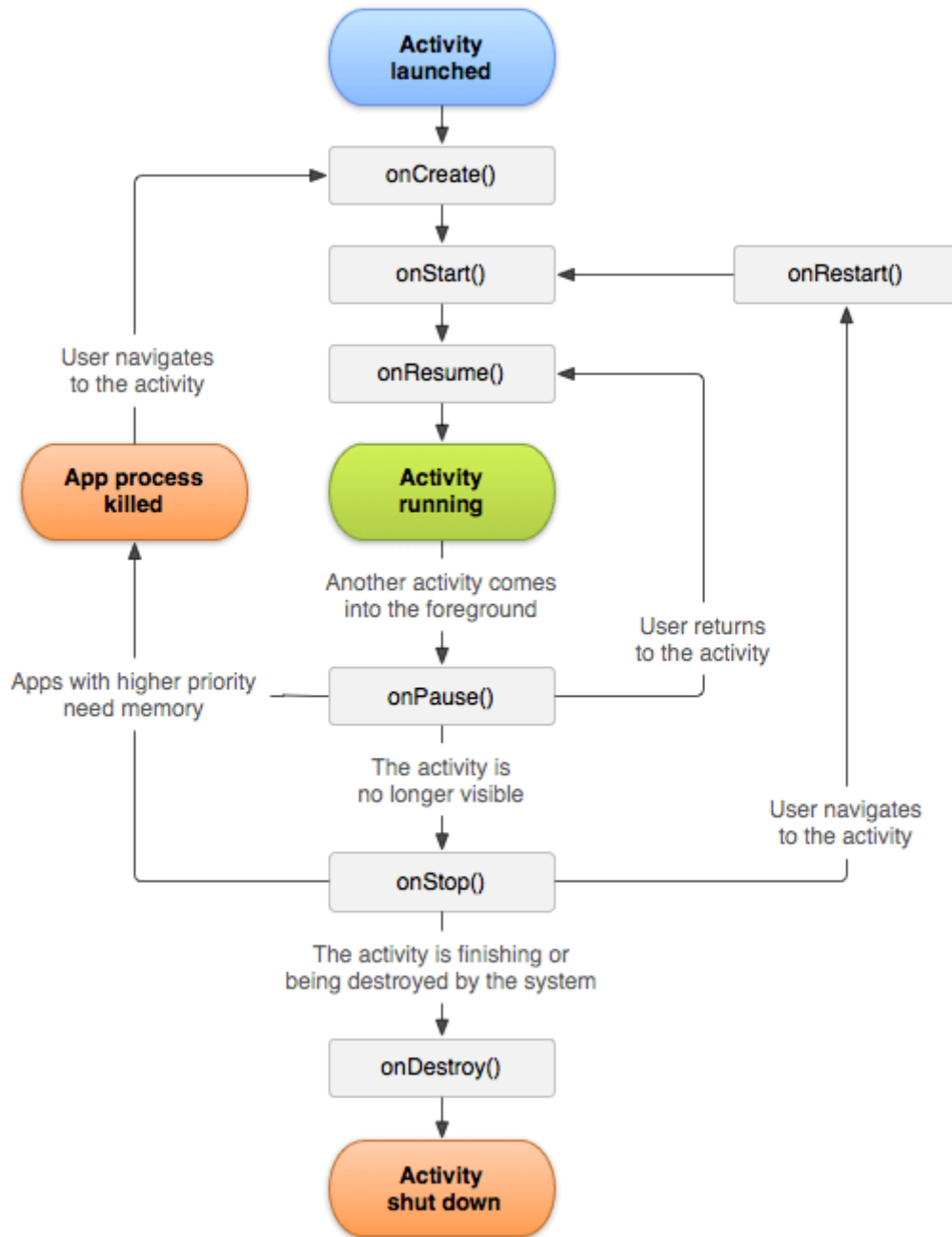


Fig 2: Android activity life cycle

3.1.2 Application Layer

Between the client layer, the mobile application, and the database layer, which is housed on the server, the application layer acts as an intermediary layer. In order to avoid the client layer having to interface directly with the database, it provides the service of getting data from the database and delivering it to the client layer in response to requests made by the client layer. Java programming is used to implement the application layer. It comprises of a number of web methods that expose

the data and make it accessible via HTTP protocols across the internet. It offers client layer and database layer interaction between programs. In addition to transferring data from the database to the client layer, it also gives mobile applications the tools they need to save data in the database utilising this layer.

3.1.3 Database Layer

MySQL server is used to implement the database layer. It manages and saves the application's database. The database includes data on users who have registered for the application, farmers they have registered, and various measurements of the farmer's parameters that have been calculated on the client side, including the physical fitness index (PFI), VO₂ max, body mass index (BMI), mid upper arm circumference (MUAC), calf circumference (CC), various measurements calculated using skinfold parameters, minimum dietary diversity, basal metabolic rate (BMR), resting energy expenditure (REE) and total energy expenditure (TEE).

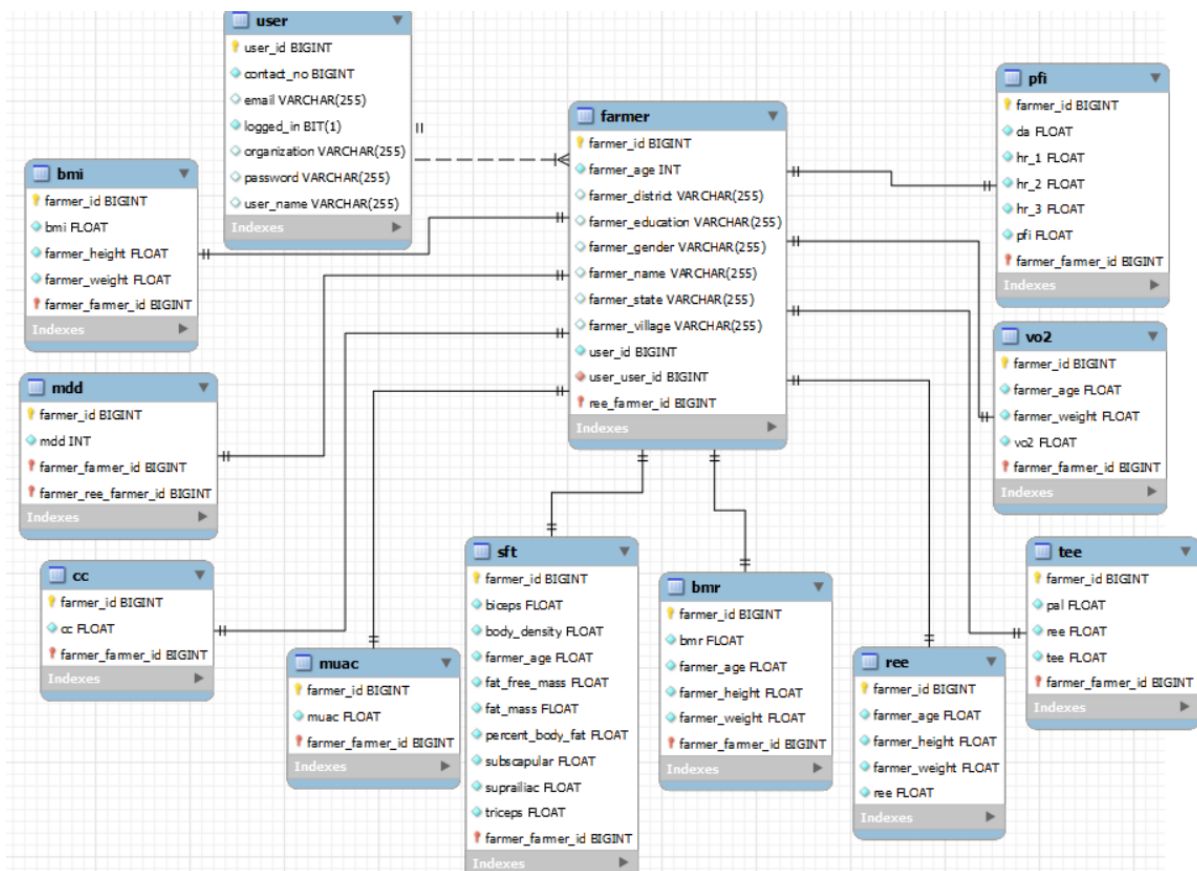


Fig 3: ER Diagram of the Database

3.1.4 Architecture of the application

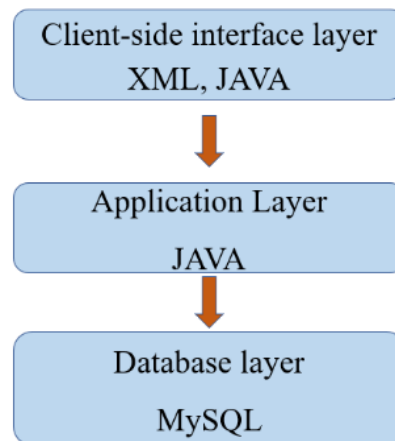


Fig 4: 3-tier Architecture of the Application

3.1.5 Work Flow of the DSS: NutriEnergy

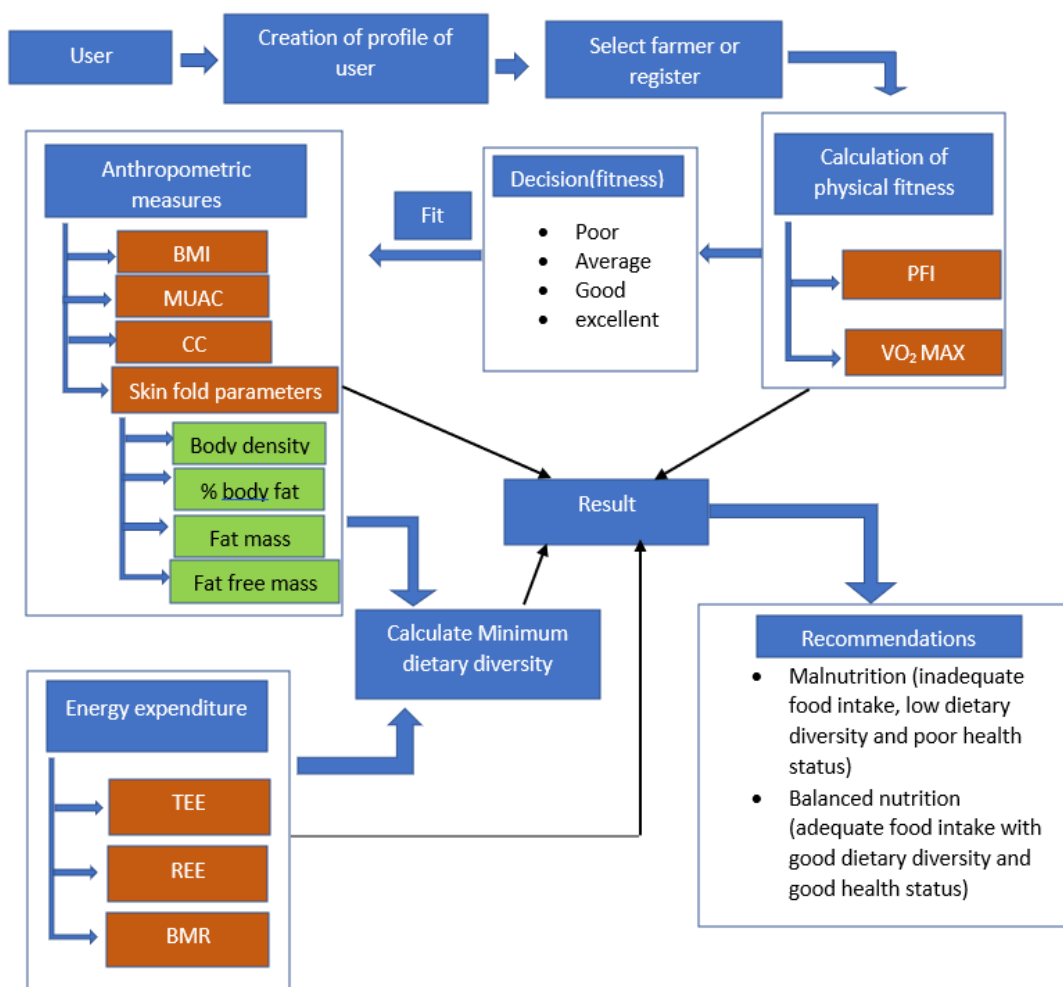


Fig 5: Work Flow of the NutriEnergy

3.2 Programming Environments

3.2.1 eXtensible Markup Language (XML)

The World Wide Web Consortium (W3C) created the markup language known as XML, or extensible markup language (W3C). In 1988, the World Wide Web Consortium (W3C) developed it. It is a Markup language designed specifically for data storage. An alternative name for it is a scripting language. An xml has the extension.xml and is case-sensitive. The ability of XML to communicate data between two different architectures and systems, as well as the capacity to create custom tags based on requirements, is by far its most crucial feature.

Data is transmitted between applications using XML. As a result, XML prioritizes the data or information that has to be shared or exchanged across programs. XML has no impact on software or hardware. In XML, the data is the most crucial factor. Closing tags are necessary in XML, which is frequently used to create online applications. Apart from that, faults are not permitted in XML. The Standard Generalized Markup Language (SGML) is where it originates (SGML). The most crucial technology for improving the web right now is XML. Almost all platforms and frameworks now in use support and use XML. The ability of XML to transfer all organized, semi-structured, and structured data between other systems is a significant additional benefit.

3.2.2 Java Programming language

Currently, one of the most well-liked and often used programming languages for creating applications is JAVA. A Sun Microsystems Company team lead by James Gosling created Java in 1991. In 1995, it had three different names: "Green Talk," "Oak," and ultimately "JAVA."

High-level programming languages like Java are frequently used to create desktop and web apps. Java is a platform-independent programming language with an easy-to-use, safe-to-use, and portable user interface. Java makes use of Object-Oriented Programming concepts like Encapsulation, Inheritance, and Polymorphism. In Java encapsulation, data and methods are combined into a single container. Data is shielded from unauthorised access from the outside world through encapsulation. By defining the data and methods inside the classes and packages and giving them various types of access specifiers, encapsulation is accomplished.

With the intention of employing the same method for several purposes, Java performs polymorphism through method overloading and overriding. The OOPs concept of inheritance

creates a parent-child relationship between classes and permits the transfer of properties from one class to another. Java can be used to create applications for the Android platform. There are numerous packages, interfaces, and abstract classes available in the Android programming language. Each activity in the Android application is a java class that is a component of the Activity Class.

3.3 Application Development Environments

3.3.1 Android Studio IDE

The official Integrated Development Environment (IDE) of Google, known as Android Studio, is widely used to create Android applications worldwide. Android Studio is a popular interface for DSS development among Android developers. It was first established on May 16, 2013. But the official debut took place at a Google conference. It can be downloaded and used in our system at no cost. It is without cost. There is now a licence for Apache 2.0. Android Studio IDEA is the foundation for IntelliJ IDEA. By utilising IntelliJ Android Studio's code editor and development tools, a variety of additional engaging interactive features are made available to users. The following list includes all of the Android Studio's features.

- A flexible Gradle-based framework that makes it simple and fun to construct our app.
- An efficient and interactive emulator for testing our project.
- A solitary setting where a development firm can create apps for various Android devices.
- Provides integrated support for Google Cloud Platform as well.

Versions 4.1 of Android Studio and 1.8 of JDK are used to create the application.

3.3.2 Eclipse IDE

The Eclipse was introduced by IBM in November 2001 and supported by a number of software firms. Eclipse is an integrated development environment for computer programming (IDE). It has a default workspace and a plug-in framework for environment customization. Although Eclipse is largely written in Java and is designed for the development of Java applications, it may also be used to create programmes in other programming languages, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, and more, through the use of plug-ins. Ruby, Perl, PHP, Prolog, Julia, Lasso, Lua, Natural, and JavaScript.

It can also be used to make Mathematica packages and LaTeX papers (with the aid of a TeXlipse plug-in). A few examples of development environments include Eclipse CDT for C/C++, Eclipse PDT for PHP, and Eclipse Java development tools (JDT) for Java and Scala. The Eclipse software development kit (SDK) is free and open-source software released under the terms of the Eclipse Public License, while not being compatible with the GNU General Public License. It works perfectly with IcedTea and was one of the first IDEs to support GNU Classpath. For the RESTful web service development in this project, the Eclipse IDE version 4.24 and JDK version 1.8 was used in this project.

3.3.3 Spring Boot Tool

The spring framework served as the foundation for the creation of Spring Boot, an open-source Java-based framework for building microservices. It was developed by Pivotal Team and is used to build standalone, production-ready spring apps. Java programmers can build standalone, immediately executable spring applications using the Spring Boot framework. Without having to set up a complete Spring configuration, you can get started with just a few configurations.

Due to its qualities and advantages, which are detailed below, Spring Boot is a wise choice.

- It gives you the option to customise how Java Beans, XML settings, and Database Transactions are configured.
- It provides reliable batch processing and maintains REST endpoints.
- Thanks to the annotation `@EnableAutoConfiguration`, everything in Spring Boot is automatically configured; no manual configuration is needed.
- It has an annotation-based spring application.
- It includes an embedded servlet container, which makes managing dependencies easier.

Additionally, Spring Boot offers starter to manage the dependencies, like Spring Boot Started Actuator dependency to manage and monitor your application, Spring Boot Starter Security dependence to construct Rest endpoints, Spring Boot Starter Thyme Leaf dependency to build web applications, etc. We also have the ability to use spring initializr to build and download the fundamental framework of a spring boot application.

3.4 Physical Fitness Index (PFI)

Physical fitness is the capacity to carry out daily tasks with vigour and alertness without suffering undue tiredness, as well as having the stamina to engage in leisure activities and react to emergencies and unforeseen circumstances. The Physical Fitness Index (PFI) computation is one of the most important aspects in figuring out someone's cardiopulmonary efficiency. Physically fit workers are often in good health. Workers were instructed to complete the exercise or 'step stool ergometer' for 5 minutes. After that, they were asked to report their heart rates at the end of the first, second, and third minutes of recovery. The following equation is used to calculate PFI:

$$\text{Physical Fitness Index (PFI)} = \frac{\text{Duration of activity/s}}{\text{Sum of 1st, 2nd and 3rd min of recovery of HR}} \times 100$$

Here, HR is heart rate of respondents.

Table 1: Range of PFI

PFI value	Level of physical fitness
Upto 80	Poor physical fitness
81-100	Low average fitness
101-115	High average fitness
116-135	Good fitness
136-150	Very good fitness
Beyond 150	Excellent fitness

3.5 VO₂ max (ml/kg/min)

VO₂ max is the term used to describe the maximal rate of oxygen (O₂) that your body can use when exercising. The respiratory process needs the usage of oxygen, which is necessary for breathing. Your lungs take in oxygen and transform it into adenosine triphosphate (ATP) energy when you breathe it in. Adenosine triphosphate (ATP) aids in the release of carbon dioxide (CO₂) during exhalation. The benefits are obvious: the more oxygen your body can receive and utilise to produce the greatest quantity of ATP energy is directly correlated to your VO₂ max. The VO₂ max can be calculated using the following formula based on the respondents' age and body weight to

identify their level of physical fitness. The respondents were categorised using classification given by Varghese et al (1996).

$$\text{VO}_2 \text{ Max (l/min)} = 0.023 \times \text{Body weight(Kgs)} - 0.034 \times \text{Age(yrs)} + 1.65$$

$$\text{VO}_2 \text{ Max (l/kg/min)} = \frac{\text{VO}_2 \text{ Max (l/min)}}{\text{Body Weight (kg)}} \times 1000$$

Table 2: Relative VO₂ Max range

Relative VO ₂ Max	Level of physical fitness
Up to 15	Poor
16-25	Low Average
26-30	High Average
31-40	Good
41-45	Very Good
Beyond 45	Excellent

3.6 Body Mass Index(BMI)

Based on height and weight, the body mass index (BMI) calculation determines the percentage of body fat. Instead of measuring body fat precisely, it uses an equation to estimate it. A person's BMI can be used to determine whether they are overweight or healthy. A high BMI implies that there is too much fat in the body, whereas a low BMI suggests that there is not enough fat in the body. Significant health issues like heart disease, high blood pressure, and diabetes are more likely to develop in people with higher BMIs. A low BMI can cause health problems such as anaemia, weakened immune system, and bone loss.

$$\text{BMI (kg/m}^2\text{)} = \frac{\text{Weight (kg)}}{\text{Height}^2\text{(m)}}$$

Table 3: BMI Classification (Garrow, 1987)

BMI Range	Presumptive diagnosis
<16.0	CED Grade III(Severe)
16.0-17.0	CED Grade II(Moderate)
17.0-18.5	CED Grade I(Mild)
18.5-20.0	Low weight normal
20.0-25.0	Normal
25.0-30.0	Obese Grade I
>30	Obese Grade II

3.7 Mid Upper Arm Circumference (MUAC)

It is common knowledge that the MUAC, or mid-upper arm circumference, is a sign of nutritional health. Protein-energy deficiency can be found using the mid-upper arm circumference (MUAC) measurement. Because the arm contains both subcutaneous fat and muscle, changes in MUAC may indicate a change in muscle mass, a change in subcutaneous fat, or both. In situations with limited resources, where people have a tendency to have less subcutaneous fat, changes in MUAC are more likely to reflect changes in muscle mass. Halfway between the scapula's acromion process and the ulna's olecranon process, the MUAC was measured.

Table 4: MUAC Classification (Tonder E V et al., 2019)

MUAC tertiles	For men	For women
I (Under nutrition)	<22.9 cm	<22.8 cm
II (Normal)	22.9 – 25.6 cm	22.8 – 25.4 cm
III (Obese)	\geq 25.7 cm	\geq 25.5 cm

3.8 Calf Circumference (CC)

Calf circumference (CC) is an anthropometric measurement that has been connected to the nutritional status of the senior population¹⁶ and is closely related to total body muscle mass. The maximum horizontal distance around the left leg was measured when the individual stood straight. CC were separated into quantitative tertiles based on sex.

Table 5: CC Classification (Portero McLellan *et al.*, 2010)

CC tertiles	For men	For women
I (Under nutrition)	<26.0 cm	<25.0 cm
II (Normal)	26.0 – 29.9 cm	25.0 – 28.3 cm
III (Obese)	\geq 30.0 cm	\geq 28.4 cm

3.9 Skin-fold Thicknesses

When assessing body fat content using skin-fold thicknesses, the underlying premise is that measurements of subcutaneous fat serve as a proxy for total body fat. This is not entirely accurate because the thickness of subcutaneous fat does not always correspond to a constant percentage of body fat. Furthermore, age, sex, and ethnicity all have a big impact on how subcutaneous fat is distributed. In any case, this approach provides useful information and is commonly employed. Skin-fold thickness measurements can be taken in one or more sites. Experts differ on which skin-fold represents total body fat more accurately. On the other hand, the triceps skin-fold is the one that is most frequently used for a single-site method to measure body fat. At the midpoint of the upper arm, the triceps skinfold is measured. When the elbow is 90 percent extended, the midpoint is halfway between the olecranon process of the ulna and the acromium process of the shoulder. The arm is then gently let hang while being extended to the side. With the thumb and forefinger, grip the skin-fold directly above the midarm point. The callipers are put at a straight angle to the skin at the midarm position, 1 cm beneath the surface. Maintain the fold when taking the measurement. It is possible to analyse several skin-fold locations, with limb and trunk locations being the most typical.

Skin-fold measures can be used to assess body density. The four skinfold sites of the triceps, biceps, subscapular, and supriliac are measured, and the log of their sum is substituted into one of the following equations, where D is the anticipated body density (g/ml) and L is the log of the total of the four skinfolds (mm). The density number can then be converted to Percent Body Fat (%BF). Additional body density calculations as well as other equations exist to calculate body fat using skinfold measurements. Skinfolds, sex, and age are utilised in regression models like Durnin and Womersley's to calculate body fat and fat-free mass.

Table 6: Body Density equations

age (years)	equations for males	equations for females
< 17	$D = 1.1533 - (0.0643 \times L)$	$D = 1.1369 - (0.0598 \times L)$
17-19	$D = 1.1620 - (0.0630 \times L)$	$D = 1.1549 - (0.0678 \times L)$
20-29	$D = 1.1631 - (0.0632 \times L)$	$D = 1.1599 - (0.0717 \times L)$
30-39	$D = 1.1422 - (0.0544 \times L)$	$D = 1.1423 - (0.0632 \times L)$
40 -49	$D = 1.1620 - (0.0700 \times L)$	$D = 1.1333 - (0.0612 \times L)$
> 50	$D = 1.1715 - (0.0779 \times L)$	$D = 1.1339 - (0.0645 \times L)$

3.10 Minimum Dietary Diversity

Data on dietary diversity will be calculated as the number of respondents divided by the number of food items consumed daily. The number of distinct food groups consumed by respondents the day before out of the list of 10 FGs suggested for use in calculating the MDD was used to construct the Dietary Diversity Score (DDS) (Leroy et al., 2015). All foods are given an equal weighting and the sum of the consumed food groups is calculated.

Table 7: Different Food Groups

Food stuffs
Grains, white roots and tubers, and plantains
Pulses (beans, peas and lentils)
Nuts and seeds
Dairy
Meat, poultry and fish
Eggs
Dark green leafy vegetables
Other vitamin A-rich fruits and vegetables
Other vegetables
Other fruits
Low dietary diversity <5 group
High dietary diversity ≥5 group

3.11 Total Energy Expenditure (TEE)

Total Energy Expenditure (TEE) is made up of three primary parts in humans:

- i. Basal Metabolic Rate (BMR): Energy used up while completely at rest and in a post-absorptive state; accounts for about 60% of total energy expenditure in sedentary people.
- ii. Thermic Effect of Food (TEF), which accounts for 10% of TEE, is an increase in energy expenditure brought on by the digestion, absorption, and storage of food and nutrients.
- iii. Energy Expenditure of Activity - This is further broken down into Exercise-related Activity Thermogenesis, which is associated with active sports or exercise, and Non-Exercise Activity Thermogenesis (NEAT), which is associated with daily activities like

fidgiting, spontaneous muscle contraction, etc. Exercise-related Activity Thermogenesis may account for up to 10% of TEE in active adults, compared to 0% of TEE in inactive people. The remaining TEE is made up of NEAT.

Predictive equations using anthropometric data:

Equations for predicting BMR is given by Harris-Benedict. Predictive equations for Resting energy expenditure (REE) are given by Mifflin et al, 1990.

TEE= REE x AF (Activity Factor by PAL)

Table 8: PAL values for different activities

Physical activity status	Activity Factor
Sedentary	1.2
Light Activity	1.375
Moderate Activity	1.55
Very Active	1.725
Extra Active	1.9

3.12 Resting Energy Expenditure (REE)

In an interprandial state, while a person is sleeping well and awake, they are consuming resting energy expenditure (REE), which in healthy individuals equates to about 1 kcal/kg of body weight each hour. **Mifflin et al. (1990)** used data from 498 healthy people, comprising females (n = 247) and men (n = 251), aged 19-78 y (45±14 y, I±SD), to create a prediction equation for resting energy expenditure (REE). Indirect calorimetry was used to measure REE in subjects who were normal weight (n = 264) and obese (n = 234). For both men and women, multiple-regression models were used to determine the associations between REE and weight, height, and age (R2= 0.71).

<p><u>Mifflin equations for REE:</u> REE (males) = 10 X weight (kg) + 6.25 X height (cm) - 5 x age(y) + 5 REE (females) = 10 X weight(kg) + 6.25 X height (cm) - 5 X age (y) - 161.</p>
--

3.13 Basal Metabolic Rate (BMR)

The energy expended when a subject is completely resting both mentally and physically—that is, when they are awake, sitting in a reclined position in a room with a comfortable temperature and humidity level—is known as basal metabolic rate. In essence, it is the bare minimum of energy needed to keep things like kidney and heart function going.

The Harris-Benedict principle, often known as the Harris-Benedict equation, is a technique for calculating a person's basal metabolic rate (BMR). The approximate daily kilocalorie intake to maintain current body weight can be calculated by multiplying the estimated BMR value by a factor that reflects the person's level of activity.

Harris-Benedict equations for BMR:

$$\text{BMR(Male)} = 66.47 + (13.7516 * \text{weight in kg}) + (5.0033 * \text{height in cm}) - (6.75 * \text{age in years})$$

$$\text{BMR(Female)} = 655.1 + (9.5634 * \text{weight in kg}) + (1.8496 * \text{height in cm}) - (4.67 * \text{age in years})$$

3.14 Integration of existing application NutriGuide

An android based mobile application named 'NutriGuide' has been developed with the purpose to keep track of nutritional status of the farmers using different anthropometric techniques. This application not only help in assessing the physical fitness and nutritional status but also provide them with a quick note on the value of different type of foods for different age groups.

Physical fitness includes physical fitness index (PFI) and VO₂ max. Different anthropometric measures that are provided in the application are body mass index (BMI), mid upper arm circumference (MUAC), calf circumference (CC) and skinfold thickness. The existing application's database was restructured and the energy balance techniques (like basal metabolic rate, resting energy expenditure and total energy expenditure) were added.

3.15 Database Designing

3.15.1 MySQL

The most popular database management system for relational databases in use today is MySQL. The Oracle Corporation offers support for this open-source database. It is a rapid, scalable, and user-friendly database management system when compared to Oracle Database and Microsoft SQL Server. In order to create robust and dynamic server-side or web-based enterprise systems, it is usually integrated with PHP scripts. It was developed, sold, and supported by the Swedish

company MySQL AB, and it was designed using the C and C++ programming languages.

Client-Server architecture is the foundation of MySQL. This paradigm is designed to let end users, or clients, or stakeholders, to access resources from a central computer, or server, via network services. A graphical user interface (GUI) is used by clients to submit requests, and the server responds with the intended outcome as soon as the requests and instructions match. The client-server architecture is used in the same way in the MySQL environment. The database for this mobile application was built using MySQL version 8.0.

3.15.2 Database Structure

Table 9: User details

Column Name	Data Type
user_id	Integer
contact_no	Integer
email	String
organization	String
password	String
user_name	String

Table 10: Farmer details

Column Name	Data Type
farmer_id	Integer
user_id	Integer
farmer_name	String
farmer_age	Integer
farmer_gender	String
farmer_education	String
farmer_state	String
farmer_district	String
farmer_village	String

Table 11: PFI details

Column Name	Data Type
farmer_id	Integer
da	Integer
Hr_1	Integer
Hr_2	Integer
Hr_3	Integer
pfi	Float

Table 12: VO₂ details

Column Name	Data Type
farmer_id	Integer
farmer_age	Integer
farmer_weight	Float
vo2	Float

Table 13: BMI details

Column Name	Data Type
farmer_id	Integer
farmer_height	Float
farmer_weight	Float
bmi	Float

Table 14: MUAC details

Column Name	Data Type
farmer_id	Integer
muac	Float

Table 15: CC details

Column Name	Data Type
farmer_id	Integer
cc	Float

Table 16: Skinfold thickness details

Column Name	Data Type
farmer_id	Integer
farmer_age	Integer
biceps	Float
triceps	Float
subscapular	Float
suprailiac	Float
body_density	Float
percent_body_fat	Float
fat_mass	Float
fat_free_mass	Float

Table 17: BMR details

Column Name	Data Type
farmer_id	Integer
farmer_height	Integer
farmer_weight	Integer
farmer_age	Integer
bmr	Float

Table 18: REE details

Column Name	Data Type
farmer_id	Integer
farmer_height	Integer
farmer_weight	Integer
farmer_age	Integer
ree	Float

Table 19: TEE details

Column Name	Data Type
farmer_id	Integer
ree	Float
pal	String
tee	Float

Table 20: MDD details

Column Name	Data Type
farmer_id	Integer
mdd	Integer

3.16 Apache Tomcat Server:

Apache Tomcat, also referred to as Tomcat, is an open-source web server created by the Apache Software Foundation (ASF). The Java EE web server Tomcat implements a number of Java EE specifications, such as Java Servlet, Java Server Pages (JSP), Java EL, and WebSocket, and it offers a "pure Java" HTTP web server environment where Java code can run. Tomcat can be downloaded and set up from the tomcat homepage. After setting up Tomcat, we can launch it by typing start in the /bin directory on the command line interface. Additionally, in order to deploy the Java application, navigate to the Web Apps folder and put the application's war file there. Once deployed, Tomcat will launch the Java application automatically. We use the stop command from the command line interface to terminate Tomcat. The REST API constructed using the Java programming language is hosted here using Tomcat 9 version.

CHAPTER IV

RESULTS

A Decision Support System (DSS) for Android called "*Nutri Energy*" has been created. *Nutri Energy* was developed by incorporating energy balance techniques with integration in already existing application '*Nutri Guide*'. This application's goals are to provide fundamental nutritional knowledge and assess nutritional status using different anthropometric and energy balance techniques. Utilizing various technologies and development techniques covered in earlier chapters, the application was created. The "Nutri Energy" application was created to extract values for many characteristics from specific inputs, allowing us to gain a general understanding of a person's body composition and energy expenditure. The mobile-based decision support system "Nutri Energy" will assist dietitians, nutritional experts, researchers and other stakeholders in maintaining individual dietary habits and in conducting wide or detailed surveys of a locality that will support the formulation of decisions and policies based on the findings.

4.1 Features of the Application:

The application '*Nutri Energy*' is developed, tested and validated with the sample data. The application has following features in it:

- Basic information regarding the nutritional value of various food types and appropriate foods for various age groups is provided in the nutritional information section.
- Then, to help the user use the app, it includes features for user sign-up and sign-in as well as app information.
- Another feature is the ability to register farmers.
- Physical fitness will be calculated using physical Fitness Index (PFI) and VO_2 max.
- Next, there are many anthropometric measurements, such as body mass index (BMI), mid upper arm circumference (MUAC), calf circumference (CC), body density, percentage body fat, fat mass, and fat free mass calculated in addition to dietary diversity with consumption pattern
- Finally, energy balance measures like Basal Metabolic Rate (BMR), Resting Energy Expenditure (REE) and Total Energy Expenditure (TEE) can be calculated.

4.2 User Interface of the DSS

4.2.1 Home Page

A splash-screen will appear when the app is first opened, and the home page will follow shortly after. There are two buttons on the home page, one for nutritional data and the other for assessing one's energy and nutritional state. The first button will direct user to the nutritional information display. The user can sign up, log in, and get information about the many metrics that can be calculated by clicking the second button, "Evaluate Nutritional and Energy Status".

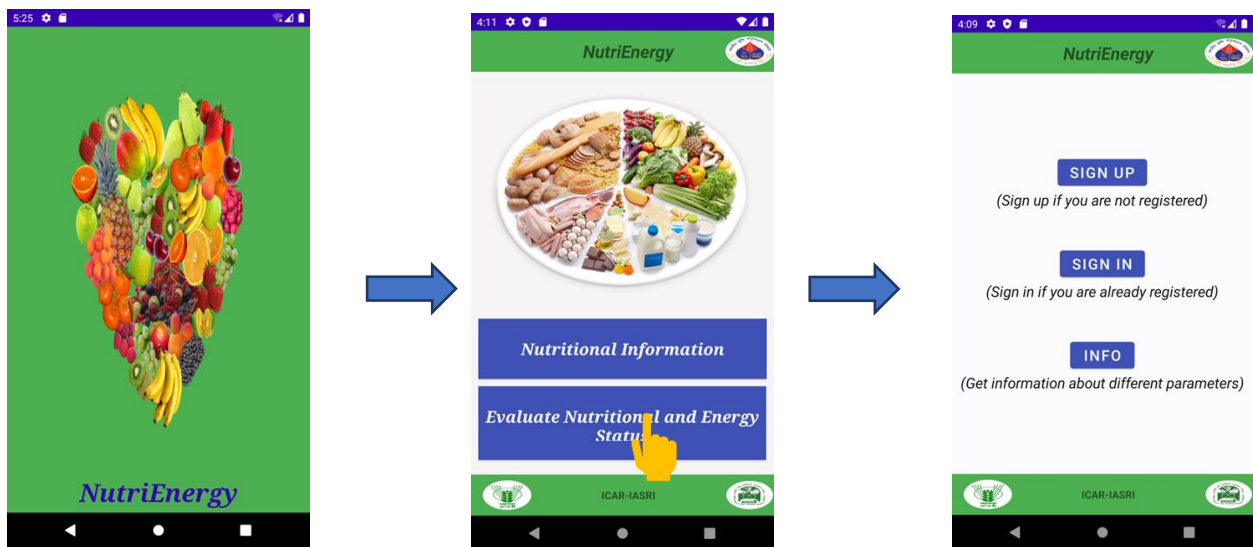


Fig 6: Splash screen and home page

4.2.2 User Registration

User will be able to evaluate different parameters only after registration. Users must enter information such as their name, email address, password, contact number and organisation. Each and every field on that page must be filled in. In the absence of any field a network fault or inaccurate data, the message "User is registered successfully" or "User Exists" is displayed. In the case of user already exists, it will directly take user to sign in page.

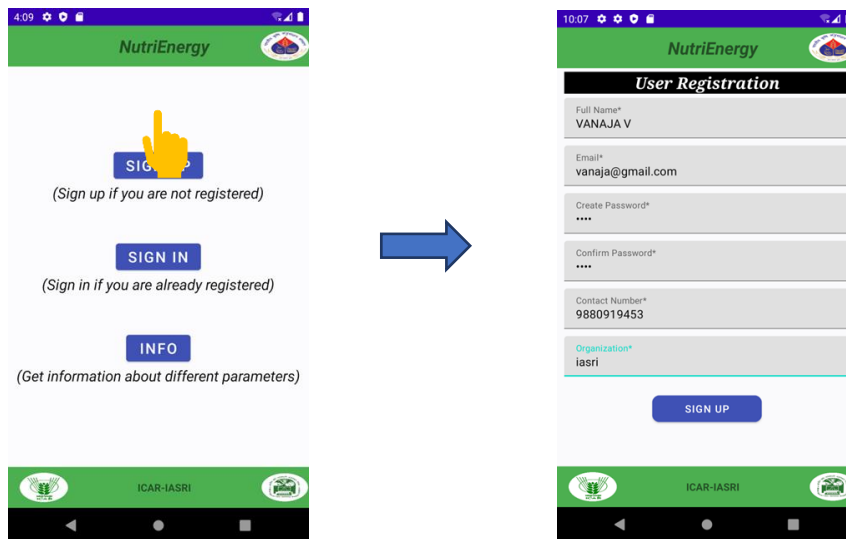


Fig 7: User registration

4.2.3 User sign in

After successfully logging in, the user can register a new farmer by selecting the "+" icon at the bottom right of the screen and evaluating the nutrition and energy of the farmer. The user must enter all required information, including name, age, qualification, state, district, and village. The fields are all required. When a user clicks the "register" button later, it displays the message "Farmer Registered Successfully" and returns them to the previous screen, which displays the list of registered farmers.

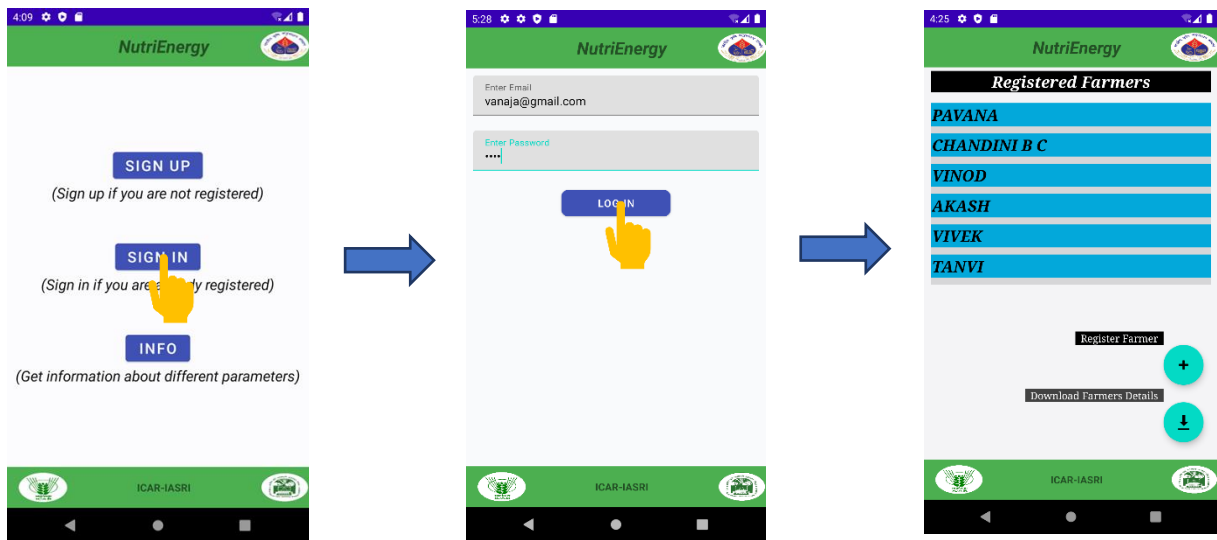


Fig 8: Login page

4.2.4 Farmer Registration and Information Download

4.2.4.1 Registration of New Farmer

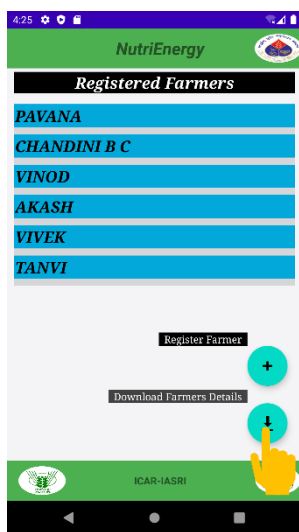
After the user successfully logged in, then user can register new farmer, whose evaluation of nutrition and energy can be done by clicking on the '+' sign on the right bottom of the screen. User have to fill all the fields like name, age, qualification, state, district and village. All the fields are made mandatory. Later, on clicking the 'register' button it will show 'Farmer Registered Successfully' message and automatically takes the user back to previous screen, where it will show the registered farmers list.



Fig 9: Farmer registration page

4.2.4.2 Download Farmers Details

When a user clicks the "Download" button on the right bottom of the screen, an excel file called "Farmers Details.xls" including all of the information provided at registration and the value of the measures can be downloaded.



	A	B	C	D	E	F	G	H
1								
2		Name	Age	Gender	Education	Village	District	State
3		PAVANA	24	Female	Graduation	godehosal	mandya	Karnataka
4		CHANDINI	56	Female	10th	balgatta	mandya	Karnataka
5		VINOD	24	Male	Post Grad	Pathamuth	Kolar	Karnataka
6		AKASH	24	Male	Post Grad	gulbarga	GULBARC	Karnataka
7		VIVEK	24	Male	Post Grad	Nandur	NASHIK	Maharash
8		TANVI	24	Female	Post Grad	Patna	Patna	Bihar
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								

Fig 10: Downloaded farmer's details excel sheet snapshot

4.2.5 Physical Fitness

User can click on the farmer's name, whose nutritional and energy parameters have to be calculated. On clicking the farmer's name it will take user to the page which contains four groups of measurements. In this under physical fitness category we have two measurements those are Physical Fitness Index (PFI) and VO₂ max. on top of this page name of the farmer will be shown. On clicking the particular button it will take user to respective measurements. The user can calculate PFI or VO₂ max or both.

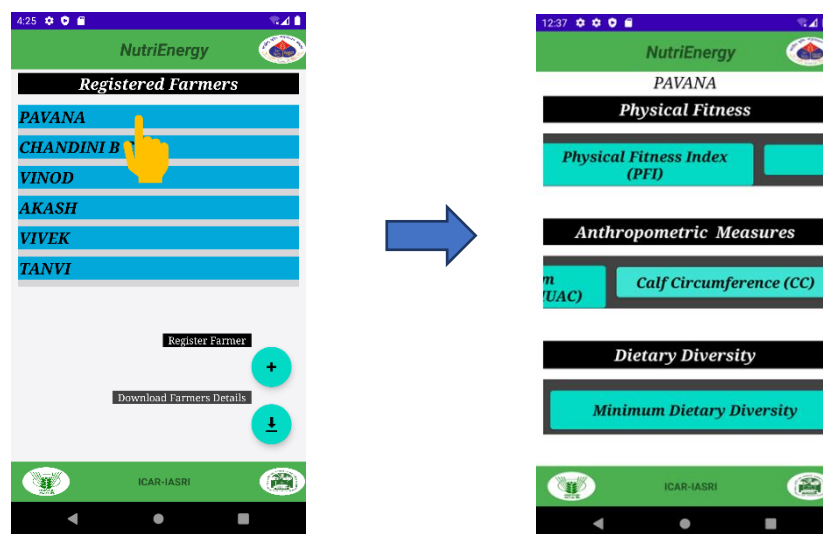


Fig 11: Moving to calculation screen

4.2.5.1 Physical Fitness Index (PFI)

On clicking the PFI button it will take user to a page where he/she have to fill the required fields. After five minutes of action with the farmer, the user must fill out three boxes: heart rate recovery after the first minute, heart rate recovery after the second minute, and heart rate recovery after the third minute. After filling all the fields user have click on 'submit' button, which gives the value of PFI, category and PFI table as shown below.



Fig 12: calculation of PFI

4.2.5.2 VO₂ max

On clicking the VO₂ max button, it takes the user to a page where it asks for farmer's body weight as input. Then user have to click on 'submit' button to get the value of VO₂ max, category and VO₂ table as shown below.

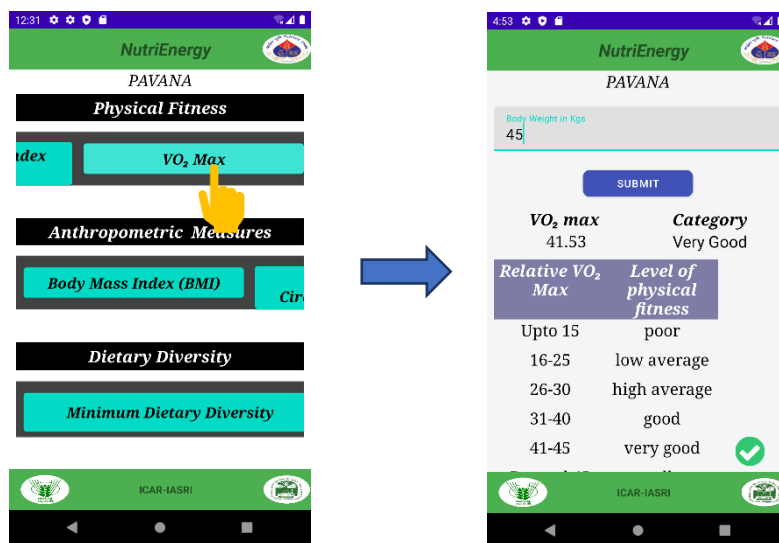


Fig 13: Calculation of VO₂ max

4.2.6 Anthropometric Measures

The user can choose from one of four anthropometric measurements in this category, including Skin Fold Parameters, Mid Upper Arm Circumference, and Body Mass Index (BMI).

4.2.6.1 Body Mass Index (BMI)

The BMI button will take you to a page where you must enter your height (in metres) and weight (in kilogrammes) for the farmer. After pressing the submit button, the BMI value, category, and BMI table will appear as displayed below.

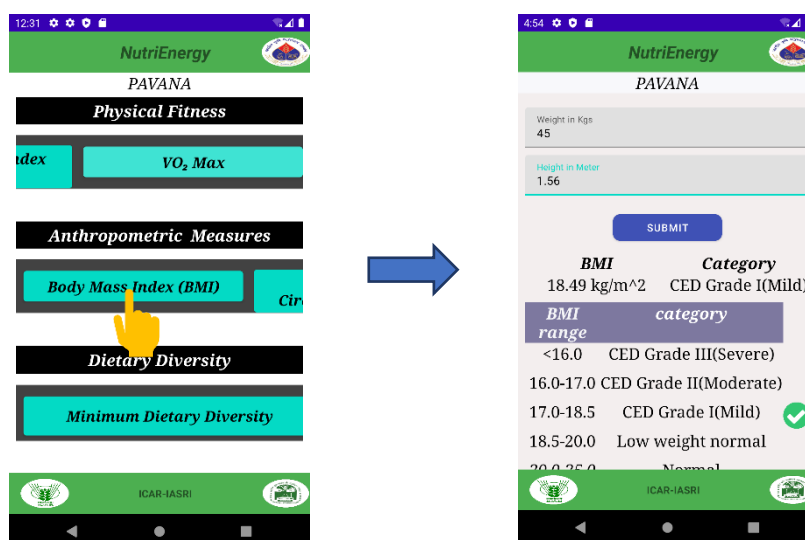


Fig 14: Calculation of BMI

4.2.6.2 Mid Upper Arm Circumference (MUAC)

The MUAC button will take user to a screen where you must enter the farmer's MUAC measurement. After pressing the submit button, the appropriate nutritional category will appear as shown below.

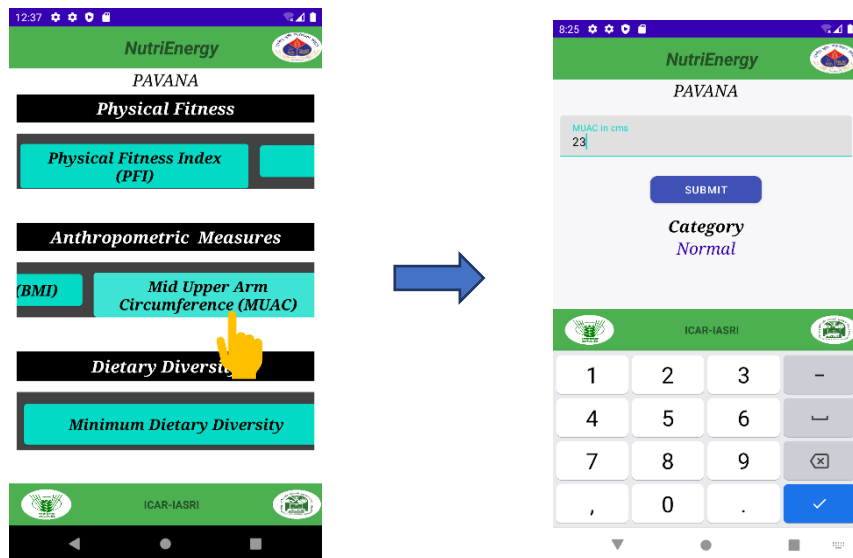


Fig 15: Categorizing MUAC

4.2.6.3 Calf Circumference (CC)

On clicking the CC button, it will lead to a page where it asks the user for CC measurement of the farmer. After clicking on submit button it will show the respective nutritional category as shown below.

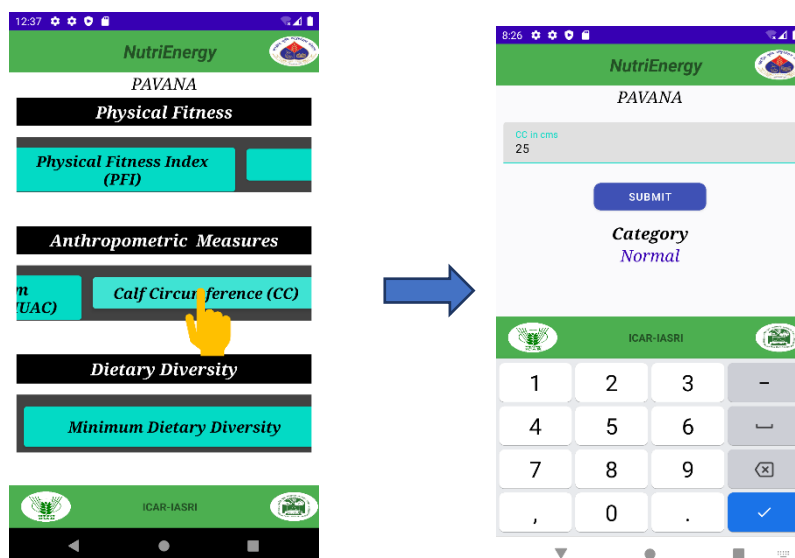


Fig 16: Categorizing CC

4.2.6.4 Skin Fold Parameters

On clicking the skin fold parameters button, it will take to a page where it asks the user for farmer's body weight (kg), skinfold at biceps, triceps, subscapular and supraillaic. After clicking on submit button it will show the values of body density, fat percentage, fat mass and fate free mass as shown below.

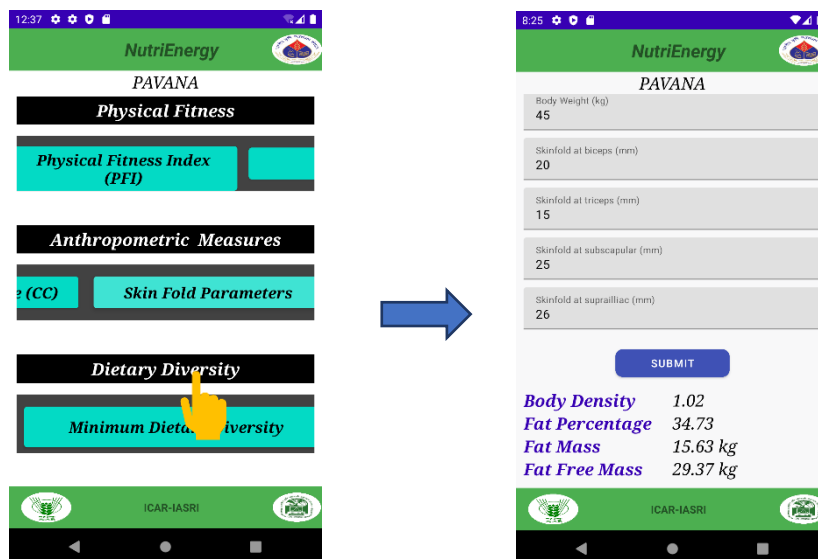


Fig 17: Calculation of skinfold thickness

4.2.7 Minimum Dietary Diversity

The user will be able to select the food groups typically consumed by the farmer. Clicking the "check" option will display the farmer's dietary diversity as low or high, as seen below, depending on the food groups they have chosen.

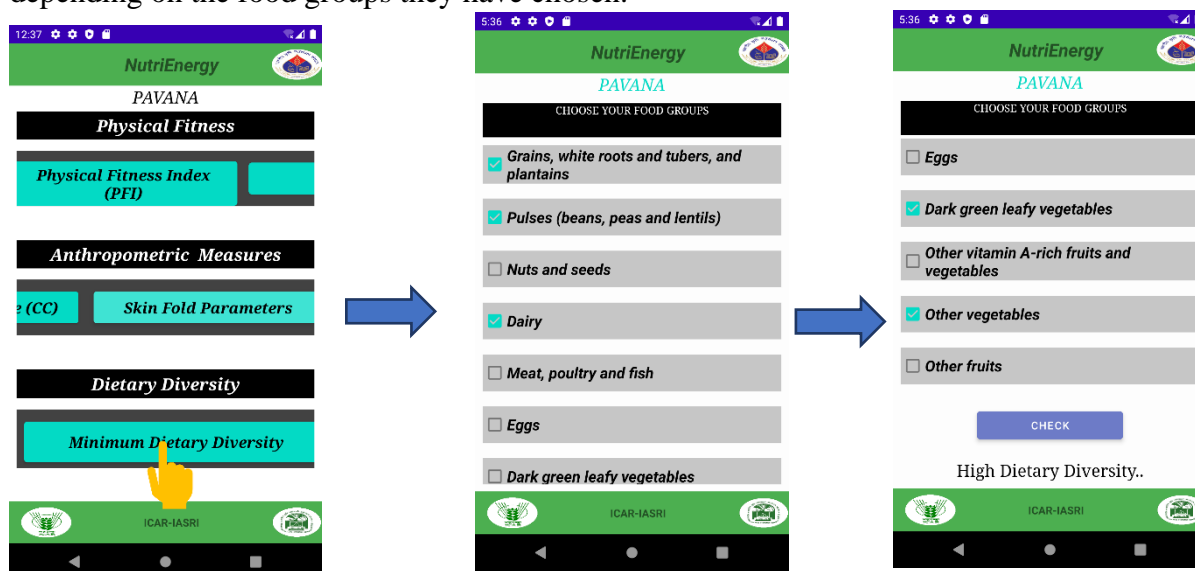


Fig 18: Calculation of Minimum Dietary Diversity

4.2.8 Energy Balance Techniques

In this category the user can select any of the three energy balance techniques like Basal Metabolic Rate (BMR), Resting Energy Expenditure (REE) and Total Energy Expenditure (TEE). Clicking on any of the button will lead the user to respective measurements.

4.2.8.1 Basal Metabolic Rate (BMR)

On clicking the BMR button, it will lead to a page where it asks the user for weight (Kg) and height (meter) of the farmer. After clicking on submit button it will show the value of BMR as shown below.

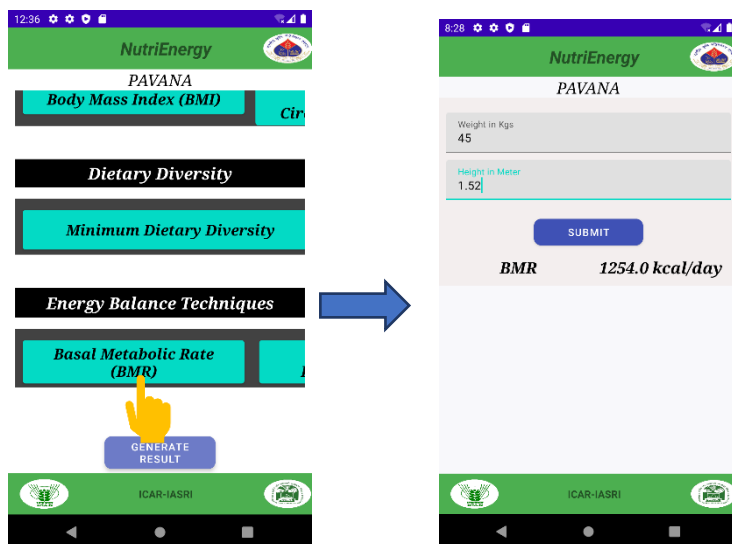


Fig 19: Calculation of BMR

4.2.8.2 Resting Energy Expenditure (REE)

On clicking the REE button, it will lead to a page where it asks the user for weight (Kg) and height (meter) of the farmer. After clicking on submit button it will show the value of REE as shown below.

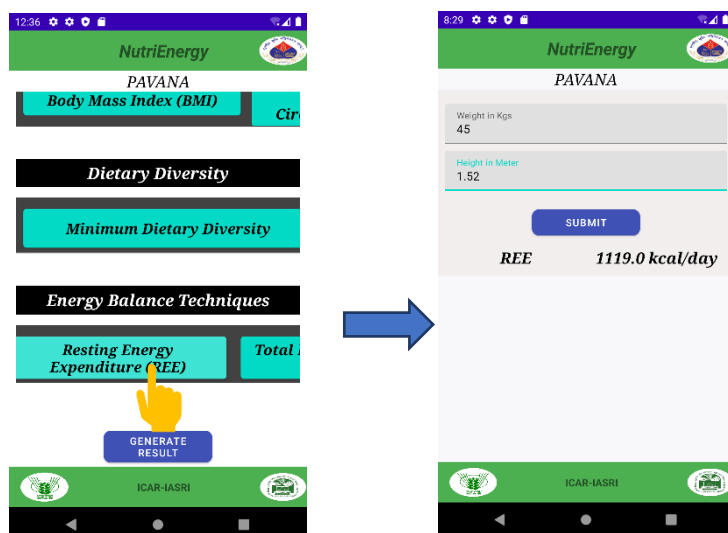


Fig 20: Calculation of REE

4.2.8.3 Total Energy Expenditure (TEE)

On clicking the TEE button, it will lead to a page where it asks the user for REE and Physical Activity Level (PAL) (selected from the table shown in that page based on the activity of the farmer) of the farmer. After clicking on submit button it will show the value of TEE as shown below.

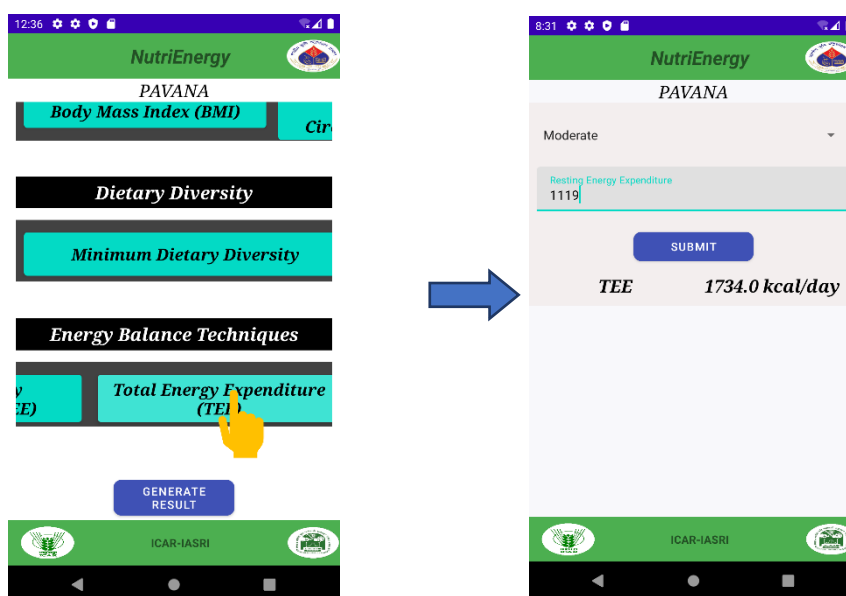


Fig 21: Calculation of TEE

4.2.9 Result

On clicking the generate result button, it will lead to a page where it shows all the details of farmer and calculation details.

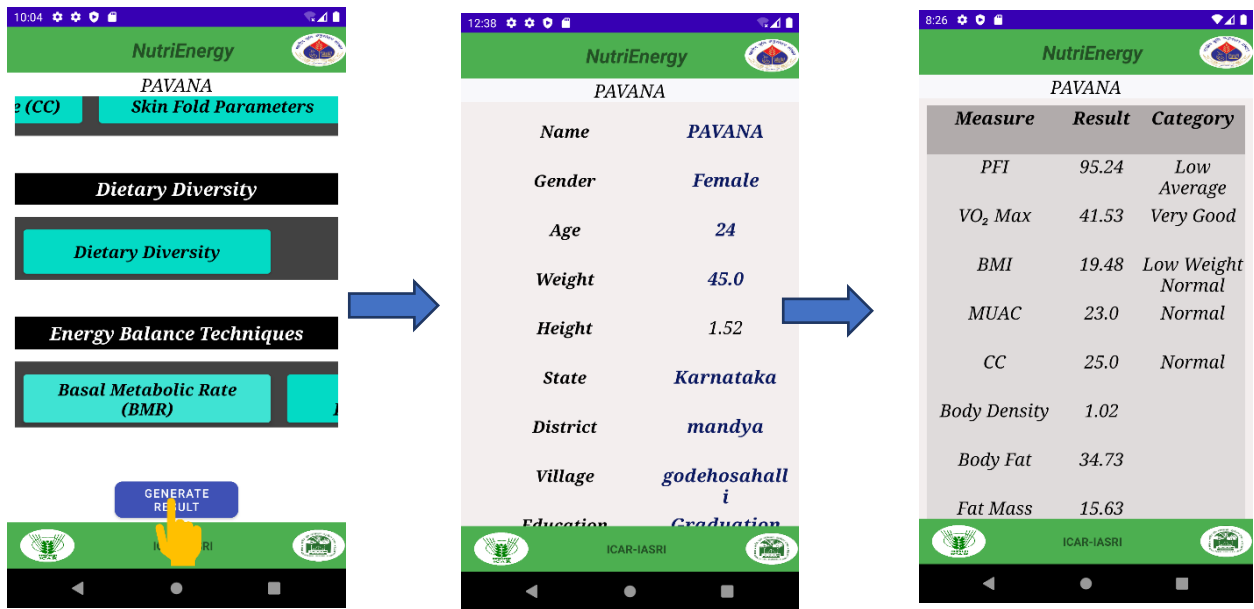


Fig 22: Snapshot of result page of particular farmer

CHAPTER V

DISCUSSION

The importance of mobile applications is growing as a result of the rapid growth of technology in information technology era. Everyone is familiar with a unique gadget known as a mobile phone, or more specifically, a smart phone, irrespective of their age. Using a smart phone, anyone can access all kinds of information from around the globe. An interactive programme that runs on a smartphone is known as a mobile application. All throughout the world, mobile applications of various kinds are being created for a variety of objectives. Mobile applications include, for instance, tools for online bill payment, online shopping, online money transfers, online hotel reservations, and online news. As a smart tool in all sectors, a smartphone application that would help farmers is now being developed by providing information and disseminating technologies. There are many different agricultural mobile applications, such as those that may help farmers with their cultivation techniques, weather forecasts, the current fertilizer rate in a given area, accurate disease and pest control information etc. Considering that a farmer cannot afford a desktop computer but may, to some extent, purchase a smart phone. All of these smartphone apps help farmers by giving them the knowledge they need to practice good farming.

In India, where agriculture still provides a significant portion of income for many small and marginal farmers and makes up a significant portion of our country's GDP, we need our farmers to be in good physical and nutritional condition. In order to protect our farmers from the dangers of malnutrition, frontline employees who are responsible for their welfare must regularly monitor the nutritional state of the farmers. The nutritional status, energy expenditure and body composition of farmers must be known in order for them to choose work limits that support their bodies and to know what types of food and quantities to consume in order to improve their nutritional status. The DSS named Nutri Energy was developed in this regard.

The DSS '*Nutri Energy*' provides the features that are present in already existing application '*Nutri Guide*' and also includes energy expenditure calculations. The application helps the user to calculate Physical Fitness Index (PFI) and VO_2 max to know about farmers fitness, which are prerequisites for further calculations. It helps with the calculation of anthropometric measures like Body Mass Index (BMI), Mid Upper Arm Circumference (MUAC), Calf Circumference (CC) and Skinfold thickness which are dependent on

These measurements will help the researchers health and nutritional experts, dieticians and

other stakeholders for assessment and evaluation of their nutritional profiling. The application offers the ability to download all of the respondents' information into an excel file, which will aid in the workers' ability to conduct further analysis. It also offers the capability of calculating the measures.

The Spring Boot tool with Eclipse IDE and Java programming language is used to create the REST api which is hosted on the server and works as a connection between the database, which is also resided on the server, and the mobile application. This mobile-based DSS "NutriGuide" is developed using the AndroidStudioIDE, Java programming language, and XML for front end development. The mobile application's backend database for this DSS runs on a MySQL server.

CHAPTER VI

SUMMARY AND CONCLUSION

Food is necessary for the nation's human resource development. It represents socioeconomic advancement, the eradication of poverty, and the safety of the nation's health. It also serves as the cornerstone of human development, particularly for women, adolescents, and children. Related biological, social, and cultural issues have an impact on health concerns for mothers and children. Agriculture continues to be a significant source of income for many small and marginal farmers in India, and it contributes significantly to the GDP of our countries. Therefore, in order to protect human resources from the risks of malnutrition, farmers and frontline staff who are in charge of their condition must regularly monitor the farmers' nutritional status. Malnutrition is a multifaceted issue that necessitates multisector responses. The results demonstrate that enhancing agricultural productivity can have a favourable effect on nutritional outcomes. In order to choose work limitations that are healthy for their bodies and to know what kind of food and how much of it they should eat in order to improve their nutritional condition, farmers must be aware of their nutritional status, body compositions and energy expenditure. In this regard, a DSS was created that not only determines various anthropometric measurements and energy balance techniques of the farmers but also provides the option to download the information in an Excel document and gives individual report.

The primary objective of the mobile-based DSS, "Nutri Energy," was to develop a mobile application for assessing the nutritional condition of farmers using several anthropometric approaches and energy balance techniques. Nutri Energy was developed by integrating energy balance techniques with already existing application NuriGuide. This application's goals are to present fundamental nutritional knowledge and assess nutritional status using several anthropometric and energy balance techniques. The DSS will work on three tier architecture. The three-tier architecture is a software architecture in which a presentation layer or interface runs on a client, the data layer or data structure gets stored on a server and Application layer will act as an intermediate layer. The DSS makes easy for researchers, dietitians, nutritionists and other stake holders to analyze farmers nutritional status using energy balance techniques.

ABSTRACT

Adequate nutrition is necessary for the country's human resource development. It is the foundation of human growth, particularly for women, adolescents, and children, and it also represents socioeconomic progress, national health security, and decreases in poverty. In India, one in three women had a body mass index (BMI) of less than 18.5 kg/m² (BMI). According to UNICEF, intergenerational malnutrition is perpetuated when an undernourished mother gives birth to an undernourished kid. Malnutrition problem can be managed by providing the desired information on time like person's intake of energy and/or nutrients based on the anthropometric measurements and energy balance techniques.

In this age of the digital revolution, numerous agricultural-related mobile applications has already been developed on various aspects. In light of this, the goal of this research is to develop an Android-based mobile application that helps assess human nutritional status using a variety of anthropometric measurements and energy balance techniques that are useful for giving them a basic picture of their health status and directing them. The already-existing application "NutriGuide" has also been integrated with a mobile-based decision support system called "NutriEnergy," which helps users assess their level of physical fitness, energy expenditure, and nutritional status as well as give them a quick summary of the benefits of various food types for various age groups.

This Android-based mobile application was developed using the Android Studio development environment, the Java programming language, eXtensible Markup Language for application user-end development, and a MySQL server as a backend database to store the personal information of humans and the various index values of all the humans. Additionally, it provides an option to save the data as an excel file for additional analysis and delivers personalized reports. In addition to all of this, the developed DSS offers a few brief details with recommendations for the recommended dietary allowance, recommended dietary intake, and food requirements for nutritional sufficiency.

सार

देश में मानव संसाधनों के विकास के लिए पर्याप्त पोषण की आवश्यकता होती है। यह मानव विकास की आधारशिला है, जोकि विशेष रूप से महिलाओं, किशोरों और बच्चों में गरीबी में कमी, सामाजिक-आर्थिक विकास और राष्ट्रीय स्वास्थ्य सुरक्षा को भी दर्शाता है। भारत में, तीन में से एक महिला बॉडी मास इंडेक्स (बीएमआई) के 18.5 किग्रा/एम² से कम के साथ कुपोषित है। यूनिसेफ के अनुसार, एक कुपोषित महिला एक कुपोषित बच्चे को जन्म देती है, जो अंतःपीढ़ी के कुपोषण का एक चक्र जारी रखता है। मानवशास्त्रीय माप और ऊर्जा संतुलन तकनीकों के आधार पर कुपोषण की समस्या को समय पर वांछित जानकारी प्रदान करके प्रबंधित किया जा सकता है जैसे कि मानव ऊर्जा और/या पोषक तत्वों का सेवन।

डिजिटल क्रांति के इस युग में, विभिन्न पहलुओं पर कृषि से संबंधित कई मोबाइल एप्लिकेशन पहले ही विकसित किए जा चुके हैं। इसके आलोक में, इस शोध का लक्ष्य एक एंड्रॉइड-आधारित मोबाइल एप्लिकेशन विकसित करना है जो विभिन्न प्रकार के मानवशास्त्रीय माप और ऊर्जा संतुलन तकनीकों का उपयोग करके किसानों की पोषण स्थिति का आकलन करने में मदद करता है जो उन्हें उनके स्वास्थ्य की स्थिति की एक बुनियादी तस्वीर देने के लिए उपयोगी हैं। इसके आलोक में, इस शोध का लक्ष्य एक एंड्रॉइड-आधारित मोबाइल एप्लिकेशन विकसित करना है जो विभिन्न प्रकार के मानवशास्त्रीय माप और ऊर्जा संतुलन तकनीकों का उपयोग करके मानव पोषण की स्थिति का आकलन करने में मदद करता है जो उन्हें उनके स्वास्थ्य की स्थिति और निर्देशन की एक बुनियादी तस्वीर देने के लिए उपयोगी हैं। पहले से मौजूद एप्लिकेशन "न्यूट्रीगाइड" को "न्यूट्रीएनर्जी" नामक एक मोबाइल-आधारित निर्णय समर्थन प्रणाली के साथ भी एकीकृत किया गया है, जो उपयोगकर्ताओं को उनके शारीरिक फिटनेस, ऊर्जा व्यय और पोषण की स्थिति के स्तर का आकलन करने में मदद करता है और साथ ही उन्हें विभिन्न आयु समूहों के लिए विभिन्न प्रकार के खाद्य पदार्थों के लाभों का त्वरित सारांश भी देता है।

एंड्रॉइड स्टूडियो डेवलपमेंट एनवायरनमेंट, जावा प्रोग्रामिंग लैंग्वेज, एप्लिकेशन यूजर-एंड डेवलपमेंट के लिए एक्स्टेंसिबल मार्कअप लैंग्वेज और मानव की व्यक्तिगत जानकारी को स्टोर करने के लिए बैकएंड डेटाबेस के रूप में एक MySQL सर्वर उपयोग किया गया है और इस Android-आधारित मोबाइल एप्लिकेशन को बनाने के लिए मानव के विभिन्न सूचकांक मूल्यों का उपयोग किया गया है। इसके अतिरिक्त, यह एक ऐसी सुविधा प्रदान करता है जो आपको आगे के विश्लेषण के लिए डेटा को एक्सेल फ़ाइल के रूप में डाउनलोड करने में सक्षम बनाता है और व्यक्तिगत रिपोर्ट प्रदान करता है। इन सभी के अलावा, विकसित डीएसएस संक्षिप्त में आहार भत्ता, आहार सेवन, और पोषण संबंधी पर्याप्तता के लिए खाद्य आवश्यकताओं के लिए सिफारिशों के साथ विवरण प्रदान करता है।

BIBLIOGRAPHY

- Abhishekh M.P., **Mukesh Kumar**, Pratibha Joshi, Shashi Dahiya, Alka Arora, Soumen Pal, (2019) Development And Validation Of Mobile Based Decision Support System For Postural Assessment Of Agricultural Activities Using Rapid Upper Limb Assessment (Rula) Technique, *Progressive Research – An International Journal, Volume 14 (Special) : 335-337, (2019) Print ISSN : 0973-6417, Online ISSN : 2454-6003*
- Alatalo, P., Järvenoja, J., Karvonen, J., Keronen, A., & Kuvaja, P. (2002). Mobile application architectures. In *International Conference on Product Focused Software Process Improvement*, 572-586, Springer, Berlin, Heidelberg.
- Bhattacharya, A., Pal, B., Mukherjee, S. and Roy, S K. (2019). Assessment of nutritional status using anthropometric variables by multivariate analysis. *BioMed Central PublicHealth* 19, 1045. <https://doi.org/10.1186/s12889-019-7372-2>.
- Durnin, J.V.G.A. and Womersley, J. (1974). *Body fat assessed from the total body density and its estimation from skinfold thickness: measurements on 481 men and women aged from 16 to 72 years. British Journal of Nutrition*, 32, 77-97.
- Elkan, AC., Engvall, IL., Cederholm, T. et al. *Rheumatoid cachexia, central obesity and malnutrition in patients with low-active rheumatoid arthritis: feasibility of anthropometry, Mini Nutritional Assessment and body composition techniques. Eur J Nutr* 48, 315–322 (2009). <https://doi.org/10.1007/s00394-009-0017-y>
- Galloway R., Dusch E., Elder L., et al. (2002). *Women's perceptions of iron deficiency and anemia prevention and control in eight developing countries. Soc Sci Med.*;55(4):529-544. doi:10.1016/s0277-9536(01)00185-x.
- Gopalan, C. (1987). *Heights of population - an index of their nutrition and socio-economic development. Bull. Nutr. Found. Ind.* 8: 1-5.
- Hand GA, Shook RP, Paluch AE, Baruth M, Crowley EP, Jagers JR, Prasad VK, Hurley TG, Hebert JR, O'Connor DP, Archer E, Burgess S, Blair SN. *The energy balance*

study: the design and baseline results for a longitudinal study of energy balance. Res Q Exerc Sport. 2013 Sep;84(3):275-86. doi: 10.1080/02701367.2013.816224. PMID: 24261006.

Harris, J. A., & Benedict, F. G. (1918). *A Biometric Study of Human Basal Metabolism. Proceedings of the National Academy of Sciences of the United States of America, 4(12), 370–373. <https://doi.org/10.1073/pnas.4.12.370>*

Mifflin MD, St Jeor ST, Hill LA, Scott BJ, Daugherty SA, Koh YO. *A new predictive equation for resting energy expenditure in healthy individuals. Am J Clin Nutr. 1990 Feb;51(2):241-7. doi: 10.1093/ajcn/51.2.241. PMID: 2305711.*

John W. Carbone, James P. McClung, Stefan M. Pasiakos, Skeletal Muscle Responses to Negative Energy Balance: Effects of Dietary Protein, *Advances in Nutrition*, Volume 3, Issue 2, March 2012, Pages 119–126, <https://doi.org/10.3945/an.111.001792>

Kumari, Lishi (2019). *Development of mobile app on Anthropometrics dimension of agricultural workers for designing and assessing farm tools and machineries (master's thesis)*. Indian Agricultural Statistics Research Institute, IARI, New Delhi.

Leroy J L, Ruel M, Frongillo E A, Harris J, Ballard T J. (2015) *Measuring the food access dimension of food security: a critical review and mapping of indicators. Food Nutrition Bulletin 36(2):167- 95.*

Pasricha,S. *Count what you eat - First Edn. 1989 - Hyderabad National Institute of Nutrition 2010 - 79pp. <http://218.248.6.43:8080/CountWhatYouEat/richfoods.do>*

Portero-McLellan, KC., Staudt, C., Silva, FR., Delbue Bernardi, JL, Baston Frenhani, P., Leandro Mehri, VA. (2010). *The use of calf circumference measurement as an anthropometric tool to monitor nutritional status in elderly inpatients. J Nutr Health Aging.;14(4):266-70. doi: 10.1007/s12603-010-0059-0. PMID: 20305992.*

Ross R, Soni S, Houle S. Negative Energy Balance Induced by Exercise or Diet: Effects on Visceral Adipose Tissue and Liver Fat. *Nutrients*. 2020; 12(4):891. <https://doi.org/10.3390/nu12040891>

- Sahana M R(2020). Mobile Based Decision support system for evaluation of postural ergonomics using Rapid Entire Body Assessment (master's thesis). Indian Agricultural Statistics Research Institute, IARI, New Delhi.
- Sharda, R., Barr, S. H., & McDonnell, J. C. (1988). *Decision support system effectiveness: a review and an empirical test. Management science, 34(2): 139-159.*
- Sonkusale, L. (2018). *Mobile App for Ergonomics assessment of Drudgery proneactivities in agriculture (master's thesis).* Indian Agricultural Statistics Research Institute, IARI, New Delhi.
- Subhasish Sarkar (2021). mobile based decision support system for nutrition and health assessment using anthropometry techniques (master's thesis). Indian Agricultural Statistics Research Institute, IARI, New Delhi.
- Tamal Kundu, Mukesh Kumar, Pratibha Joshi, Anshu Bharadwaj, Sudeep Marwaha, and Soumen Pal, (2022) "Development and Validation of Mobile Based Decision Support System for Biomechanical Evaluation of Agricultural Activities with Human Physical Drudgery Index (HPDI)", *Indian Journal of Agricultural Sciences 91 (8): 1165–7.*
- Tonder , E V., Mace, L., Steenkamp, L. , Tydeman-Edwards, R., Gerber, K. and Friskin, D. (2019) *Mid- upper arm circumference (MUAC) as a feasible tool in detecting adult malnutrition, South African Journal of Clinical Nutrition, 32:4,93-98, DOI: 10.1080/16070658.2018.1484622*
- Vergheese, MA., Saha, PN., Bhatnagar, A., Chauhan, M. (1994) *Development of data base for occupational Workload and physical fitness status of India women. Bombay: DSA Project Report, SNDT Women's University, 1994.*
- Varghese, M. A, et al., (1996). *Ergonomic Research on women in India. Post graduate Department of Family Resource Management, S N.D.T Women's University, Mumbai, pp 42-44.*
- WANG, J., THORNTON, J. C., KOLESNIK, S., and PIERSON,R.N.(2000).
Anthropometry in Body Composition *Body Composition Unit, St.*

Luke's/Roosevelt Hospital, Columbia University, New York, New York 10025, USA.

World Health Organization; (2017) *Nutritional anaemias: tools for effective prevention and control*. Geneva: (<http://apps.who.int/iris/bitstream/10665/259425/1/9789241513067eng.pdf>, retrieved 26 January 2022).

Android Studio: Available at: <https://developer.android.com/studio>

Eclipse IDE: Available at: <https://www.eclipse.org/ide/>

Dietary Guidelines for Indians: Available at: <https://play.google.com/store/apps/details?id=com.cdac.nindietaryguide&hl=en>

Nutritional Atlas: Available at: <http://www.nutrition-atlas.nin.res.in/nutritionatlas/home.php>

Nutrify India Now: Available at: <https://play.google.com/store/apps/details?id=com.ionicframework.myapp863035&gl=US>

Spring Boot Tool: Available at: <https://start.spring.io/>

MySQL server: Available at: <https://dev.mysql.com/downloads/mysql/>

APPENDIX

Sample Code

For PFI calculation:

```
package com.example.nutrienergy;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.apache.poi.ss.formula.functions.T;
import org.json.JSONException;
import org.json.JSONObject;
import org.w3c.dom.Text;

public class pfiActivity extends AppCompatActivity {
    int dur;
    EditText hr1;
    EditText hr2;
    EditText hr3;
    Float hr_1;
    Float hr_2;
    Float hr_3;
    Float pfi_cal;
    TextView pfi_category;
    TextView pfi_value;
    ImageView tickimage;
    TextView value_pfi;
    TextView value_pfi1;
    TextView value_pfi2;
    TextView value_pfi3;
    TextView value_pfi4;
    TextView value_pfi5;
    TextView value_pfi6;
    TextView category_pfi;
    TextView category_pfi1;
```

```

TextView category_pfi2;
TextView pfi_name;
TextView category_pfi_name;
TextView category_pfi3;
TextView category_pfi4;
TextView category_pfi5;
TextView category_pfi6;
ImageView tick_pfi1;
ImageView tick_pfi2;
ImageView tick_pfi3;
ImageView tick_pfi4;
ImageView tick_pfi5;
ImageView tick_pfi6;
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView((int) R.layout.activity_pfi);
    ((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);
    this.hr1 = (EditText) findViewById(R.id.hr1);
    this.hr2 = (EditText) findViewById(R.id.hr2);
    this.hr3 = (EditText) findViewById(R.id.hr3);
    this.tick_pfi1 = (ImageView) findViewById(R.id.tick_pfi1) ;
    this.tick_pfi2 = (ImageView) findViewById(R.id.tick_pfi2) ;
    this.tick_pfi3 = (ImageView) findViewById(R.id.tick_pfi3) ;
    this.tick_pfi4 = (ImageView) findViewById(R.id.tick_pfi4) ;
    this.tick_pfi5 = (ImageView) findViewById(R.id.tick_pfi5) ;
    this.tick_pfi6 = (ImageView) findViewById(R.id.tick_pfi6) ;
    this.pfi_name = (TextView) findViewById(R.id.pfi_name);
    this.category_pfi_name = (TextView) findViewById(R.id.category_pfi_name);
    this.pfi_value = (TextView) findViewById(R.id.pfi_value);
    this.pfi_category = (TextView) findViewById(R.id.pfi_category);
    this.value_pfi = (TextView) findViewById(R.id.value_pfi);
    this.value_pfi1 = (TextView) findViewById(R.id.value_pfi1);
    this.value_pfi2 = (TextView) findViewById(R.id.value_pfi2);
    this.value_pfi3 = (TextView) findViewById(R.id.value_pfi3);
    this.value_pfi4 = (TextView) findViewById(R.id.value_pfi4);
    this.value_pfi5 = (TextView) findViewById(R.id.value_pfi5);
    this.value_pfi6 = (TextView) findViewById(R.id.value_pfi6);
    this.category_pfi = (TextView) findViewById(R.id.category_pfi);
    this.category_pfi1 = (TextView) findViewById(R.id.category_pfi1);
    this.category_pfi2 = (TextView) findViewById(R.id.category_pfi2);
    this.category_pfi3 = (TextView) findViewById(R.id.category_pfi3);
    this.category_pfi4 = (TextView) findViewById(R.id.category_pfi4);
    this.category_pfi5 = (TextView) findViewById(R.id.category_pfi5);
    this.category_pfi6 = (TextView) findViewById(R.id.category_pfi6);
}

public void pfiResult(View view) {
    if (this.hr3.getText().toString().isEmpty() || this.hr2.getText().toString().isEmpty() ||
this.hr1.getText().toString().isEmpty()) {

```

```

        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }
    this.dur = 300;
    this.hr_1 = Float.valueOf(Float.parseFloat(this.hr1.getText().toString()));
    this.hr_2 = Float.valueOf(Float.parseFloat(this.hr2.getText().toString()));
    this.hr_3 = Float.valueOf(Float.parseFloat(this.hr3.getText().toString()));
    this.pfi_cal = Float.valueOf((((float) this.dur) / ((this.hr_1.floatValue() +
this.hr_2.floatValue() + this.hr_3.floatValue())) * 100.0f);
    this.pfi_cal = Float.valueOf(((float) Math.round(this.pfi_cal.floatValue() * 100.0f)) /
100.0f);
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();
    try {
        object.put("farmerId", Farmer.farmerId);
        object.put("hr_1", this.hr_1);
        object.put("hr_2", this.hr_2);
        object.put("hr_3", this.hr_3);
        object.put("da", this.dur);
        object.put("pfi", this.pfi_cal);
        System.out.println(object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
    requestQueue.add(new JsonObjectRequest(1, "http://10.0.2.2:8080/user/farmer/PFI",
object, new Response.Listener<JSONObject>() {
        @SuppressWarnings("ResourceAsColor")

        public void onResponse(JSONObject response) {
            pfiActivity.this.pfi_name.setText("PFI");
            pfiActivity.this.category_pfi_name.setText("Category");
            TextView textView = pfiActivity.this.pfi_value;
            textView.setText("" + pfiActivity.this.pfi_cal+"");
            pfiActivity.this.value_pfi.setText("PFI value");
            pfiActivity.this.value_pfi.setBackgroundColor(R.color.teal_200);
            pfiActivity.this.value_pfi1.setText("upto 80");
            pfiActivity.this.value_pfi2.setText("81-100");
            pfiActivity.this.value_pfi3.setText("101-115");
            pfiActivity.this.value_pfi4.setText("116-135");
            pfiActivity.this.value_pfi5.setText("136-150");
            pfiActivity.this.value_pfi6.setText("beyond 150");
            pfiActivity.this.category_pfi.setText("Level of Fitness");
            pfiActivity.this.category_pfi.setBackgroundColor(R.color.teal_200);
            pfiActivity.this.category_pfi1.setText("poor");
            pfiActivity.this.category_pfi2.setText("low average");
            pfiActivity.this.category_pfi3.setText("high average");
            pfiActivity.this.category_pfi4.setText("good");
            pfiActivity.this.category_pfi5.setText("very good");
        }
    });
}

```

```

        pfiActivity.this.category_pfi6.setText("excellent");

        if (((double) pfiActivity.this.pfi_cal.floatValue()) <= 80.0d) {
            pfiActivity.this.pfi_category.setText("Poor");
            pfiActivity.this.tick_pfi1.setImageResource(R.drawable.tickimage);
        }
        else if (((double) pfiActivity.this.pfi_cal.floatValue()) > 80.0d && ((double)
pfiActivity.this.pfi_cal.floatValue()) <= 100.0d) {
            pfiActivity.this.pfi_category.setText("Low Average");
            pfiActivity.this.tick_pfi2.setImageResource(R.drawable.tickimage);
        }
        else if (((double) pfiActivity.this.pfi_cal.floatValue()) > 100.0d && ((double)
pfiActivity.this.pfi_cal.floatValue()) <= 115.0d) {
            pfiActivity.this.pfi_category.setText("High Average");
            pfiActivity.this.tick_pfi3.setImageResource(R.drawable.tickimage);
        }
        else if (((double) pfiActivity.this.pfi_cal.floatValue()) > 115.0d && ((double)
pfiActivity.this.pfi_cal.floatValue()) <= 135.0d) {
            pfiActivity.this.pfi_category.setText("Good");
            pfiActivity.this.tick_pfi4.setImageResource(R.drawable.tickimage);
        }
        else if (((double) pfiActivity.this.pfi_cal.floatValue()) > 135.0d && ((double)
pfiActivity.this.pfi_cal.floatValue()) <= 150.0d) {
            pfiActivity.this.pfi_category.setText("Very Good");
            pfiActivity.this.tick_pfi5.setImageResource(R.drawable.tickimage);
        }
        else if (((double) pfiActivity.this.pfi_cal.floatValue()) > 150.0d) {
            pfiActivity.this.pfi_category.setText("Excellent");
            pfiActivity.this.tick_pfi6.setImageResource(R.drawable.tickimage);
        }
        else {
            pfiActivity.this.pfi_category.setText("Error");
        }
    }
},
    new Response.ErrorListener() {
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(pfiActivity.this.getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
        }
    });
}

public void sam(View view) {
    startActivity(new Intent(this, samActivity.class));
}
}

```

For VO2 calculation:

```

package com.example.nutrienergy;

```

```
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
```

```
public class vo2Activity extends AppCompatActivity {
    int age;
    EditText bodyWeight_vo2;
    float relativeVO2;
    Double vo2;
    TextView vo2_category;
    TextView vo2_value;
    Float weight;
    TextView vo2_name;
    TextView category_vo2_name;
    TextView value_vo2;
    TextView vo2_value1;
    TextView vo2_value2;
    TextView vo2_value3;
    TextView vo2_value4;
    TextView vo2_value5;
    TextView vo2_value6;
    TextView category_vo2;
    TextView vo2_category1;
    TextView vo2_category2;
    TextView vo2_category3;
    TextView vo2_category4;
    TextView vo2_category5;
    TextView vo2_category6;
    ImageView vo2_tick1;
    ImageView vo2_tick2;
    ImageView vo2_tick3;
    ImageView vo2_tick4;
    ImageView vo2_tick5;
    ImageView vo2_tick6;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

setContentView((int) R.layout.activity_vo2);
((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);
this.bodyWeight_vo2 = (EditText) findViewById(R.id.bodyWeight_vo2);
this.vo2_name = (TextView) findViewById(R.id.vo2_name);
this.category_vo2_name = (TextView) findViewById(R.id.category_vo2_name);
this.vo2_value = (TextView) findViewById(R.id.vo2_value);
this.vo2_category = (TextView) findViewById(R.id.vo2_category);
this.category_vo2 = (TextView) findViewById(R.id.category_vo2);
this.vo2_category1 = (TextView) findViewById(R.id.vo2_category1);
this.vo2_category2 = (TextView) findViewById(R.id.vo2_category2);
this.vo2_category3 = (TextView) findViewById(R.id.vo2_category3);
this.vo2_category4 = (TextView) findViewById(R.id.vo2_category4);
this.vo2_category5 = (TextView) findViewById(R.id.vo2_category5);
this.vo2_category6= (TextView) findViewById(R.id.vo2_category6);
this.value_vo2 = (TextView) findViewById(R.id.value_vo2);
this.vo2_value1 = (TextView) findViewById(R.id.vo2_value1);
this.vo2_value2 = (TextView) findViewById(R.id.vo2_value2);
this.vo2_value3= (TextView) findViewById(R.id.vo2_value3);
this.vo2_value4 = (TextView) findViewById(R.id.vo2_value4);
this.vo2_value5= (TextView) findViewById(R.id.vo2_value5);
this.vo2_value6 = (TextView) findViewById(R.id.vo2_value6);
this.vo2_tick1 = (ImageView) findViewById(R.id.vo2_tick1) ;
this.vo2_tick2 = (ImageView) findViewById(R.id.vo2_tick2) ;
this.vo2_tick3 = (ImageView) findViewById(R.id.vo2_tick3) ;
this.vo2_tick4 = (ImageView) findViewById(R.id.vo2_tick4) ;
this.vo2_tick5 = (ImageView) findViewById(R.id.vo2_tick5) ;
this.vo2_tick6 = (ImageView) findViewById(R.id.vo2_tick6) ;
}

public void vo2Result(View view) {
    if (this.bodyWeight_vo2.getText().toString().isEmpty()) {
        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }
    this.weight =
Float.valueOf(Float.parseFloat(this.bodyWeight_vo2.getText().toString()));
    Farmer.farmerWeight = (double) this.weight.floatValue();
    this.age = Farmer.farmerAge;
    double floatValue = (double) this.weight.floatValue();
    Double.isNaN(floatValue);
    double d = (double) this.age;
    Double.isNaN(d);
    this.vo2 = Double.valueOf(((floatValue * 0.023d) - (d * 0.034d)) + 1.65d);
    double doubleValue = this.vo2.doubleValue();
    double floatValue2 = (double) this.weight.floatValue();
    this.relativeVO2 = ((float) Math.round(100.0d * ((doubleValue / floatValue2) *
1000.0d))) / 100.0f;
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();

```

```

try {
    object.put("farmerId", Farmer.farmerId);
    object.put("farmerAge", this.age);
    object.put("farmerWeight", this.weight);
    object.put("vo2", (double) this.relativeVO2);
    System.out.println(object);
} catch (JSONException e) {
    e.printStackTrace();
}
System.out.println(object);
requestQueue.add(new JsonObjectRequest(1, "http://10.0.2.2:8080/user/farmer/VO2",
object, new Response.Listener<JsonObject>() {
    @SuppressWarnings("ResourceAsColor")
    public void onResponse(JsonObject response) {
        vo2Activity.this.vo2_name.setText("VO\u2082 max ");
        vo2Activity.this.category_vo2_name.setText("Category");
        TextView textView = vo2Activity.this.vo2_value;
        textView.setText("" + vo2Activity.this.relativeVO2);
        vo2Activity.this.value_vo2.setText("Relative VO\u2082 Max");
        vo2Activity.this.value_vo2.setBackgroundColor(R.color.teal_200);
        vo2Activity.this.vo2_value1.setText("Upto 15");
        vo2Activity.this.vo2_value2.setText("16-25");
        vo2Activity.this.vo2_value3.setText("26-30");
        vo2Activity.this.vo2_value4.setText("31-40");
        vo2Activity.this.vo2_value5.setText("41-45");
        vo2Activity.this.vo2_value6.setText("Beyond 45");

        vo2Activity.this.category_vo2.setText("Level of physical fitness");
        vo2Activity.this.category_vo2.setBackgroundColor(R.color.teal_200);
        vo2Activity.this.vo2_category1.setText("poor");
        vo2Activity.this.vo2_category2.setText("low average");
        vo2Activity.this.vo2_category3.setText("high average");
        vo2Activity.this.vo2_category4.setText("good");
        vo2Activity.this.vo2_category5.setText("very good");
        vo2Activity.this.vo2_category6.setText("excellent");

        if (((double) vo2Activity.this.relativeVO2) <= 15.0d) {
            vo2Activity.this.vo2_category.setText("Poor");
            vo2Activity.this.vo2_tick1.setImageResource(R.drawable.tickimage);
        }
        else if (((double) vo2Activity.this.relativeVO2) > 15.0d && ((double)
vo2Activity.this.relativeVO2) <= 25.0d) {
            vo2Activity.this.vo2_category.setText("Low Average");
            vo2Activity.this.vo2_tick2.setImageResource(R.drawable.tickimage);
        }
        else if (((double) vo2Activity.this.relativeVO2) > 25.0d && ((double)
vo2Activity.this.relativeVO2) <= 30.0d) {
            vo2Activity.this.vo2_category.setText("High Average");
            vo2Activity.this.vo2_tick3.setImageResource(R.drawable.tickimage);
        }
    }
}

```

```

        else if (((double) vo2Activity.this.relativeVO2) > 30.0d && ((double)
vo2Activity.this.relativeVO2) <= 40.0d) {
            vo2Activity.this.vo2_category.setText("Good");
            vo2Activity.this.vo2_tick4.setImageResource(R.drawable.tickimage);
        }
        else if (((double) vo2Activity.this.relativeVO2) > 40.0d && ((double)
vo2Activity.this.relativeVO2) <= 45.0d) {
            vo2Activity.this.vo2_category.setText("Very Good");
            vo2Activity.this.vo2_tick5.setImageResource(R.drawable.tickimage);
        }
        else if (((double) vo2Activity.this.relativeVO2) > 45.0d) {
            vo2Activity.this.vo2_category.setText("Excellent");
            vo2Activity.this.vo2_tick6.setImageResource(R.drawable.tickimage);
        }
        else {
            vo2Activity.this.vo2_category.setText("Error");
        }
    }
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(vo2Activity.this.getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
}));
}

public void sam(View view) {
    startActivity(new Intent(this, samActivity.class));
}
}

```

For BMI calculation:

```

package com.example.nutrienergy;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.apache.poi.ss.formula.functions.T;
import org.json.JSONException;

```

```

import org.json.JSONObject;

public class bmiActivity extends AppCompatActivity {
    Float bmi;
    TextView bmi_category;
    EditText bmi_height;
    TextView bmi_value;
    EditText bmi_weight;
    Float height_;
    Float weight_;
    TextView value;
    TextView value1;
    TextView value2;
    TextView value3;
    TextView value4;
    TextView value5;
    TextView value6;
    TextView value7;
    TextView category;
    TextView category1;
    TextView category2;
    TextView category3;
    TextView category4;
    TextView category5;
    TextView category6;
    TextView category7;
    TextView bmi_name;
    TextView category_name;

    ImageView tick1;
    ImageView tick2;
    ImageView tick3;
    ImageView tick4;
    ImageView tick5;
    ImageView tick6;
    ImageView tick7;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_bmi);
        ((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);
        this.bmi_weight = (EditText) findViewById(R.id.bmi_weight);
        this.bmi_height = (EditText) findViewById(R.id.bmi_height);
        this.bmi_value = (TextView) findViewById(R.id.bmr_value);
        this.bmi_category = (TextView) findViewById(R.id.bmi_category);
        this.bmi_name = (TextView) findViewById(R.id.bmi_name);
        this.category_name = (TextView) findViewById(R.id.category_name);
        this.value = (TextView) findViewById(R.id.value);
        this.value1 = (TextView) findViewById(R.id.value1);
        this.value2 = (TextView) findViewById(R.id.value2);
    }
}

```

```

this.value3 = (TextView) findViewById(R.id.value3);
this.value4 = (TextView) findViewById(R.id.value4);
this.value5 = (TextView) findViewById(R.id.value5);
this.value6 = (TextView) findViewById(R.id.value6);
this.value7 = (TextView) findViewById(R.id.value7);
this.category = (TextView) findViewById(R.id.category);
this.category1 = (TextView) findViewById(R.id.category1);
this.category2 = (TextView) findViewById(R.id.category2);
this.category3 = (TextView) findViewById(R.id.category3);
this.category4 = (TextView) findViewById(R.id.category4);
this.category5 = (TextView) findViewById(R.id.category5);
this.category6 = (TextView) findViewById(R.id.category6);
this.category7 = (TextView) findViewById(R.id.category7);
this.tick1 = (ImageView) findViewById(R.id.tick1) ;
this.tick2 = (ImageView) findViewById(R.id.tick2) ;
this.tick3 = (ImageView) findViewById(R.id.tick3) ;
this.tick4 = (ImageView) findViewById(R.id.tick4) ;
this.tick5 = (ImageView) findViewById(R.id.tick5) ;
this.tick6 = (ImageView) findViewById(R.id.tick6) ;
this.tick7 = (ImageView) findViewById(R.id.tick7) ;
}

public void bmiResult(View view) {
    if (this.bmi_weight.getText().toString().isEmpty() ||
this.bmi_height.getText().toString().isEmpty()) {
        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }
    this.weight_ = Float.valueOf(Float.parseFloat(this.bmi_weight.getText().toString()));
    this.height_ = Float.valueOf(Float.parseFloat(this.bmi_height.getText().toString()));

    Farmer.farmerWeight = (double) this.weight_.floatValue();
    this.bmi = Float.valueOf(this.weight_.floatValue() / (this.height_.floatValue() *
this.height_.floatValue()));
    this.bmi = Float.valueOf(((float) Math.round(this.bmi.floatValue() * 100.0f)) / 100.0f);

    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();
    try {
        object.put("farmerId", Farmer.farmerId);
        object.put("farmerHeight", this.height_);
        object.put("farmerWeight", this.weight_);
        object.put("bmi", this.bmi);
        System.out.println(object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
    requestQueue.add(new JsonRequest(1, "http://10.0.2.2:8080/user/farmer/BMI",

```

```

object, new Response.Listener<JSONObject>() {
    @SuppressWarnings("ResourceAsColor")
    public void onResponse(JSONObject response) {
        bmiActivity.this.bmi_name.setText("BMI");
        bmiActivity.this.category_name.setText("Category");
        TextView textView = bmiActivity.this.bmi_value;
        textView.setText("" + bmiActivity.this.bmi + " kg/m^2");
        bmiActivity.this.value.setText("BMI range");
        bmiActivity.this.value.setBackground-color(R.color.teal_200);
        bmiActivity.this.value1.setText("<16.0");
        bmiActivity.this.value2.setText("16.0-17.0");
        bmiActivity.this.value3.setText("17.0-18.5");
        bmiActivity.this.value4.setText("18.5-20.0");
        bmiActivity.this.value5.setText("20.0-25.0");
        bmiActivity.this.value6.setText("25.0-30.0");
        bmiActivity.this.value7.setText(">30");
        bmiActivity.this.category.setText("category");
        bmiActivity.this.category.setBackground-color(R.color.teal_200);
        bmiActivity.this.category1.setText("CED Grade III(Severe)");
        bmiActivity.this.category2.setText("CED Grade II(Moderate)");
        bmiActivity.this.category3.setText("CED Grade I(Mild)");
        bmiActivity.this.category4.setText("Low weight normal");
        bmiActivity.this.category5.setText("Normal");
        bmiActivity.this.category6.setText("Obese Grade I");
        bmiActivity.this.category7.setText("Obese Grade II");

        if (((double) bmiActivity.this.bmi.floatValue()) <= 16.0d) {
            bmiActivity.this.bmi_category.setText("CED Grade III(Severe)");
            bmiActivity.this.tick1.setImageResource(R.drawable.tickimage);
        }
        else if (((double) bmiActivity.this.bmi.floatValue()) > 16.0d && ((double)
bmiActivity.this.bmi.floatValue()) <= 17.0d) {
            bmiActivity.this.bmi_category.setText("CED Grade II(Moderate)");
            bmiActivity.this.tick2.setImageResource(R.drawable.tickimage);
        }
        else if (((double) bmiActivity.this.bmi.floatValue()) > 17.0d && ((double)
bmiActivity.this.bmi.floatValue()) <= 18.5d) {
            bmiActivity.this.bmi_category.setText("CED Grade I(Mild)");
            bmiActivity.this.tick3.setImageResource(R.drawable.tickimage);
        }
        else if (((double) bmiActivity.this.bmi.floatValue()) > 18.5d && ((double)
bmiActivity.this.bmi.floatValue()) <= 20.0d) {
            bmiActivity.this.bmi_category.setText("Low Weight Normal");
            bmiActivity.this.tick4.setImageResource(R.drawable.tickimage);
        }
        else if (((double) bmiActivity.this.bmi.floatValue()) > 20.0d && ((double)
bmiActivity.this.bmi.floatValue()) <= 25.0d) {
            bmiActivity.this.bmi_category.setText("Normal");
            bmiActivity.this.tick5.setImageResource(R.drawable.tickimage);
        }
    }
}

```

```

        else if (((double) bmiActivity.this.bmi.floatValue()) > 25.0d && ((double)
bmiActivity.this.bmi.floatValue()) <= 30.0d) {
            bmiActivity.this.bmi_category.setText("Obese Grade I");
            bmiActivity.this.tick6.setImageResource(R.drawable.tickimage);
        }
        else if (((double) bmiActivity.this.bmi.floatValue()) > 30.0d) {
            bmiActivity.this.bmi_category.setText("Obese Grade II");
            bmiActivity.this.tick7.setImageResource(R.drawable.tickimage);
        }
        else {
            bmiActivity.this.bmi_category.setText("Error");
        }
    }
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(bmiActivity.this(getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
}));
}

public void sft(View view) {
    startActivity(new Intent(this, sftActivity.class));
}
}

```

For MUAC:

```

package com.example.nutrienergy;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;

public class muacActivity extends AppCompatActivity {
    EditText muac;
    TextView muac_category;
    float muac_value;
    TextView category_muac;
    public void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView((int) R.layout.activity_muac);
((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);
this.muac = (EditText) findViewById(R.id.muac);
this.muac_category = (TextView) findViewById(R.id.muac_category);
this.category_muac = (TextView) findViewById(R.id.category_muac);
}

public void get_muac(View view) {
    if (this.muac.getText().toString().isEmpty()) {
        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }
    float parseFloat = Float.parseFloat(this.muac.getText().toString());
    this.muac_value = parseFloat;
    this.muac_value = ((float) Math.round(parseFloat * 100.0f)) / 100.0f;
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();
    try {
        object.put("farmerId", Farmer.farmerId);
        object.put("muac", (double) this.muac_value);
        System.out.println(object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
    requestQueue.add(new JsonRequest(1, "http://10.0.2.2:8080/user/farmer/MUAC",
object, new Response.Listener<JSONObject>() {
        public void onResponse(JSONObject response) {
            muacActivity.this.category_muac.setText("Category");
            if (Farmer.farmerGender.equals("Male")) {
                if (((double) muacActivity.this.muac_value) < 22.9d) {
                    muacActivity.this.muac_category.setText("Under Nutrition");
                }
                else if (((double) muacActivity.this.muac_value) >= 22.9d && ((double)
muacActivity.this.muac_value) < 25.7d) {
                    muacActivity.this.muac_category.setText("Normal");
                }
                else if (((double) muacActivity.this.muac_value) >= 25.7d) {
                    muacActivity.this.muac_category.setText("Obese");
                }
            }
            else if (!Farmer.farmerGender.equals("Female")) {
            }
            else {
                if (((double) muacActivity.this.muac_value) < 22.8d) {
                    muacActivity.this.muac_category.setText("Under Nutrition");
                }
                else if (((double) muacActivity.this.muac_value) >= 22.8d && ((double)

```

```

muacActivity.this.muac_value) < 25.5d) {
    muacActivity.this.muac_category.setText("Normal");
}
else if (((double) muacActivity.this.muac_value) >= 25.5d) {
    muacActivity.this.muac_category.setText("Obese");
}
}
}
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(muacActivity.this(getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
}));
}

```

```

public void sft_from_muac(View view) {
    startActivity(new Intent(this, sftActivity.class));
}
}

```

For CC calculation:

```

package com.example.nutrienergy;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;

public class ccActivity extends AppCompatActivity {
    EditText cc;
    TextView cc_category;
    float cc_value;
    TextView cc_category_value;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_cc);
        ((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);
        this.cc = (EditText) findViewById(R.id.cc);
        this.cc_category = (TextView) findViewById(R.id.cc_category);
    }
}

```

```

        this.cc_category_value= (TextView) findViewById(R.id.cc_category_value);
    }

    public void get_cc(View view) {
        if (this.cc.getText().toString().isEmpty()) {
            Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
            return;
        }
        float parseFloat = Float.parseFloat(this.cc.getText().toString());
        this.cc_value = parseFloat;
        this.cc_value = ((float) Math.round(parseFloat * 100.0f)) / 100.0f;
        RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
        JSONObject object = new JSONObject();
        try {
            object.put("farmerId", Farmer.farmerId);
            object.put("cc", (double) this.cc_value);
            System.out.println(object);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        System.out.println(object);
        requestQueue.add(new JsonRequest(1, "http://10.0.2.2:8080/user/farmer/CC",
object, new Response.Listener<JSONObject>() {
            public void onResponse(JSONObject response) {
                ccActivity.this.cc_category.setText("Category");
                if (Farmer.farmerGender.equals("Male")) {
                    if (((double) ccActivity.this.cc_value) < 22.9d) {
                        ccActivity.this.cc_category_value.setText("Under Nutrition");
                    }
                    else if (((double) ccActivity.this.cc_value) >= 22.9d && ((double)
ccActivity.this.cc_value) < 25.7d) {
                        ccActivity.this.cc_category_value.setText("Normal");
                    }
                    else if (((double) ccActivity.this.cc_value) >= 25.7d) {
                        ccActivity.this.cc_category_value.setText("Obese");
                    }
                }
                else if (!Farmer.farmerGender.equals("Female")) {
                }
                else {
                    if (((double) ccActivity.this.cc_value) < 22.8d) {
                        ccActivity.this.cc_category_value.setText("Under Nutrition");
                    }
                    else if (((double) ccActivity.this.cc_value) >= 22.8d && ((double)
ccActivity.this.cc_value) < 25.5d) {
                        ccActivity.this.cc_category_value.setText("Normal");
                    }
                    else if (((double) ccActivity.this.cc_value) >= 25.5d) {
                        ccActivity.this.cc_category_value.setText("Obese");
                    }
                }
            }
        });
    }

```

```

        }
    }
}
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(ccActivity.this.getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
}));
}

public void sft_from_cc(View view) {
    startActivity(new Intent(this, sftActivity.class));
}
}

```

For skinfold thickness parameters calculation:

```

package com.example.nutrienergy;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;

public class sftActivity extends AppCompatActivity {
    int age;
    EditText biceps;
    Float biceps_;
    double density;
    Float density_;
    double fatFreeMass;
    Float fatFreeMass_;
    double fatMass;
    Float fatMass_;
    String gender;
    double l;
    double percentBodyFat;
    Float percentBodyFat_;
    TextView sft_body_density;
    TextView sft_fatFreeMass;
    TextView sft_fatMass;
}

```

```

TextView sft_percentage_body_fat;
EditText sft_weight;
EditText subscapular;
Float subscapular_;
EditText suprailliac;
Float suprailliac_;
EditText triceps;
Float triceps_;
double weight;
TextView bd_name;
TextView fm_name;
TextView ffm_name;
TextView fp_name;

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView((int) R.layout.activity_sft);
    ((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);
    this.sft_weight = (EditText) findViewById(R.id.sft_weight);
    this.biceps = (EditText) findViewById(R.id.biceps);
    this.triceps = (EditText) findViewById(R.id.triceps);
    this.subscapular = (EditText) findViewById(R.id.subscapular);
    this.suprailliac = (EditText) findViewById(R.id.suprailliac);
    this.sft_body_density = (TextView) findViewById(R.id.sft_body_density);
    this.sft_percentage_body_fat = (TextView)
findViewById(R.id.sft_percentage_body_fat);
    this.sft_fatMass = (TextView) findViewById(R.id.sft_fatMass);
    this.sft_fatFreeMass = (TextView) findViewById(R.id.sft_fatFreeMass);
    this.bd_name = (TextView) findViewById(R.id.bd_name);
    this.fm_name = (TextView) findViewById(R.id.fm_name);
    this.ffm_name = (TextView) findViewById(R.id.ffm_name);
    this.fp_name = (TextView) findViewById(R.id.fp_name);
    if (Farmer.farmerWeight != 0.0d) {
        EditText editText = this.sft_weight;
        editText.setText(Farmer.farmerWeight + "");
    }
}

public void sftResult(View view) {
    if (this.sft_weight.getText().toString().isEmpty() ||
this.biceps.getText().toString().isEmpty() || this.triceps.getText().toString().isEmpty() ||
this.subscapular.getText().toString().isEmpty() ||
this.suprailliac.getText().toString().isEmpty()) {
        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }
    this.age = Farmer.farmerAge;
    this.weight = (double) Float.parseFloat(this.sft_weight.getText().toString());
    this.gender = Farmer.farmerGender;
}

```

```

this.biceps_ = Float.valueOf(Float.parseFloat(this.biceps.getText().toString()));
this.triceps_ = Float.valueOf(Float.parseFloat(this.triceps.getText().toString()));
this.subscapular_ =
Float.valueOf(Float.parseFloat(this.subscapular.getText().toString()));
this.suprailliac_ = Float.valueOf(Float.parseFloat(this.suprailliac.getText().toString()));
this.l = Math.log10((double) (this.biceps_.floatValue() + this.triceps_.floatValue() +
this.subscapular_.floatValue() + this.suprailliac_.floatValue()));
if (this.gender.equals("Male")) {
    System.out.println(this.gender);
    int i = this.age;
    if (i < 17) {
        this.density = 1.1533d - (this.l * 0.0643d);
    }
    else if (i > 17 && i <= 19) {
        this.density = 1.162d - (this.l * 0.063d);
    }
    else if (i > 19 && i <= 29) {
        this.density = 1.1631d - (this.l * 0.0632d);
    }
    else if (i > 29 && i <= 39) {
        this.density = 1.1422d - (this.l * 0.0544d);
    }
    else if (i > 39 && i <= 49) {
        this.density = 1.162d - (this.l * 0.07d);
    }
    else if (i > 49) {
        this.density = 1.1715d - (this.l * 0.0779d);
    }
}
else if (this.gender.equals("Female")) {
    int i2 = this.age;
    if (i2 < 17) {
        this.density = 1.1369d - (this.l * 0.0598d);
    }
    else if (i2 > 17 && i2 <= 19) {
        this.density = 1.1549d - (this.l * 0.0678d);
    }
    else if (i2 > 19 && i2 <= 29) {
        this.density = 1.1599d - (this.l * 0.0717d);
    }
    else if (i2 > 29 && i2 <= 39) {
        this.density = 1.1423d - (this.l * 0.0632d);
    }
    else if (i2 > 39 && i2 <= 49) {
        this.density = 1.1333d - (this.l * 0.0612d);
    }
    else if (i2 > 49) {
        this.density = 1.1339d - (this.l * 0.0645d);
    }
}
}

```

```

this.density_ = Float.valueOf(((float) Math.round(this.density * 100.0d)) / 100.0f);
double d = (495.0d / this.density) - 450.0d;
this.percentBodyFat = d;
this.percentBodyFat_ = Float.valueOf(((float) Math.round(d * 100.0d)) / 100.0f);
double d2 = this.weight * ((4.95d / this.density) - 4.5d);
this.fatMass = d2;
this.fatMass_ = Float.valueOf(((float) Math.round(d2 * 100.0d)) / 100.0f);
double d3 = this.weight - this.fatMass;
this.fatFreeMass = d3;
this.fatFreeMass_ = Float.valueOf(((float) Math.round(d3 * 100.0d)) / 100.0f);
RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
JSONObject object = new JSONObject();
try {
    object.put("farmerId", Farmer.farmerId);
    object.put("farmerAge", Farmer.farmerAge);
    object.put("biceps", this.biceps_);
    object.put("triceps", this.triceps_);
    object.put("subscapular", this.subscapular_);
    object.put("suprailiac", this.suprailiac_);
    object.put("bodyDensity", this.density_);
    object.put("percentBodyFat", this.percentBodyFat_);
    object.put("fatMass", this.fatMass_);
    object.put("fatFreeMass", this.fatFreeMass_);
    System.out.println(object);
} catch (JSONException e) {
    e.printStackTrace();
}
System.out.println(object);
requestQueue.add(new JSONObjectRequest(1, "http://10.0.2.2:8080/user/farmer/SFT",
object, new Response.Listener<JSONObject>() {
    public void onResponse(JSONObject response) {
        sftActivity.this.bd_name.setText("Body Density");
        TextView textView = sftActivity.this.sft_body_density;
        textView.setText("" + sftActivity.this.density_);
        sftActivity.this.fp_name.setText("Fat Percentage");
        TextView textView2 = sftActivity.this.sft_percentage_body_fat;
        textView2.setText("" + sftActivity.this.percentBodyFat_);
        sftActivity.this.fm_name.setText("Fat Mass");
        TextView textView3 = sftActivity.this.sft_fatMass;
        textView3.setText("" + sftActivity.this.fatMass_ + " kg");
        sftActivity.this.ffm_name.setText("Fat Free Mass");
        TextView textView4 = sftActivity.this.sft_fatFreeMass;
        textView4.setText("" + sftActivity.this.fatFreeMass_ + " kg");
    }
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(sftActivity.this(getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
}));

```

```

    }

    public void goto_mdd(View view) {
        startActivity(new Intent(this, mddActivity.class));
    }
}

```

For MDD calculation:

```

package com.example.nutrienergy;

import android.os.Bundle;
import android.view.View;
import android.widget.CheckBox;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;

public class mddActivity extends AppCompatActivity {
    CheckBox checkBox1;
    CheckBox checkBox10;
    CheckBox checkBox2;
    CheckBox checkBox3;
    CheckBox checkBox4;
    CheckBox checkBox5;
    CheckBox checkBox6;
    CheckBox checkBox7;
    CheckBox checkBox8;
    CheckBox checkBox9;
    int foodGroups;
    TextView mdd_category;

    /* access modifiers changed from: protected */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_mdd);
        ((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);
        this.checkBox1 = (CheckBox) findViewById(R.id.checkBox1);
        this.checkBox2 = (CheckBox) findViewById(R.id.checkBox2);
        this.checkBox3 = (CheckBox) findViewById(R.id.checkBox3);
        this.checkBox4 = (CheckBox) findViewById(R.id.checkBox4);
        this.checkBox5 = (CheckBox) findViewById(R.id.checkBox5);
        this.checkBox6 = (CheckBox) findViewById(R.id.checkBox6);
        this.checkBox7 = (CheckBox) findViewById(R.id.checkBox7);
    }
}

```

```

this.checkBox8 = (CheckBox) findViewById(R.id.checkBox8);
this.checkBox9 = (CheckBox) findViewById(R.id.checkBox9);
this.checkBox10 = (CheckBox) findViewById(R.id.checkBox10);
this.mdd_category = (TextView) findViewById(R.id.mdd_category);
}

public void mdd(View view) {
    this.foodGroups = 0;
    if (this.checkBox1.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox2.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox3.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox4.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox5.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox6.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox7.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox8.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox9.isChecked()) {
        this.foodGroups++;
    }
    if (this.checkBox10.isChecked()) {
        this.foodGroups++;
    }
    System.out.println(this.foodGroups);
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();
    try {
        object.put("farmerId", Farmer.farmerId);
        object.put("mdd", this.foodGroups);
        System.out.println(object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
    requestQueue.add(new JsonRequest(1, "http://10.0.2.2:8080/user/farmer/MDD",

```

```

object, new Response.Listener<JSONObject>() {
    public void onResponse(JSONObject response) {
        if (mddActivity.this.foodGroups >= 5) {
            mddActivity.this.mdd_category.setText("High Dietary Diversity..");
        }
        else {
            mddActivity.this.mdd_category.setText("Low Dietary Diversity...");
        }
    }
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(mddActivity.this.getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
}));
}
}

```

For BMR calculation:

```

package com.example.nutrienergy;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;

public class bmrActivity extends AppCompatActivity {
    EditText bmr_weight;
    EditText bmr_height;
    Float bmrWeight;
    Float bmrHeight;
    Float bmr_cal;
    Button get_bmr;
    TextView bmr_value;
    TextView name_bmr;
    double age;
    String gender;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_bmr);
    }
}

```

```

((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);

this.bmr_weight = (EditText) findViewById(R.id.bmr_weight);
this.bmr_height= (EditText) findViewById(R.id.bmr_height);
this.get_bmr= (Button) findViewById(R.id.get_bmr);
this.bmr_value= (TextView) findViewById(R.id.bmr_value);
this.name_bmr = (TextView) findViewById(R.id.name_bmr);
if (Farmer.farmerWeight != 0.0d) {
    EditText editText = this.bmr_weight;
    editText.setText(Farmer.farmerWeight + "");
}
if (Farmer.farmerHeight != 0.0d) {
    EditText editText = this.bmr_height;
    editText.setText(Farmer.farmerHeight + "");
}
}

public void get_bmr(View view) {
    if (this.bmr_weight.getText().toString().isEmpty() ||
this.bmr_height.getText().toString().isEmpty() ) {
        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }
    this.age = (double) Farmer.farmerAge;
    this.gender = Farmer.farmerGender;
    this.bmrWeight = Float.valueOf(Float.parseFloat(this.bmr_weight.getText().toString()));
    this.bmrHeight = Float.valueOf(Float.parseFloat(this.bmr_height.getText().toString()));
    if (this.gender.equals("Male")) {
        this.bmr_cal = Float.valueOf((float) Math.round(66.47 + (13.75 * this.bmrWeight) +
(5.00 * this.bmrHeight * 100.0f) - (6.75 * this.age)));
    }
    else {
        this.bmr_cal = Float.valueOf((float) Math.round(655.1 + (9.56 * this.bmrWeight) +
(1.85 * this.bmrHeight * 100.0f) - (4.67 * this.age)));
    }
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();
    try {
        object.put("farmerId", Farmer.farmerId);
        object.put("farmerAge", Farmer.farmerAge);
        object.put("farmerHeight", this.bmrHeight);
        object.put("farmerWeight", this.bmrWeight);
        object.put("bmr",this.bmr_cal);
        System.out.println(object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
    requestQueue.add(new JsonRequest(1, "http://10.0.2.2:8080/user/farmer/BMR",

```

```

object, new Response.Listener<JSONObject>() {
    public void onResponse(JSONObject response) {
        TextView textView = bmrActivity.this.bmr_value;
        textView.setText("" + bmrActivity.this.bmr_cal+" kcal/day");
        name_bmr.setText("BMR");
    }
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(bmrActivity.this.getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
}));
}

public void sam(View view) {
    startActivity(new Intent(this, samActivity.class));
}
}

```

For REE calculation:

```

package com.example.nutrienergy;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;

public class reeActivity extends AppCompatActivity {
    EditText reeWeight;
    EditText reeHeight;
    Float ree_weight;
    Float ree_height;
    Float ree_cal;
    Button get_ree;
    TextView name_ree;
    TextView ree_value;
    double age;
    String gender;
    /* access modifiers changed from: protected */
    public void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentViews((int) R.layout.activity_ree);
((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);

this.reeWeight = (EditText) findViewById(R.id.reeWeight);
this.reeHeight = (EditText) findViewById(R.id.reeHeight);
this.get_ree = (Button) findViewById(R.id.get_ree);
this.ree_value = (TextView) findViewById(R.id.ree_value);
}

public void get_ree(View view) {
    if (this.reeWeight.getText().toString().isEmpty() ||
this.reeHeight.getText().toString().isEmpty() ) {
        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }
    this.age = (float) Farmer.farmerAge;
    this.gender = Farmer.farmerGender;
    this.ree_weight = Float.valueOf(Float.parseFloat(this.reeWeight.getText().toString()));
    this.ree_height = Float.valueOf(Float.parseFloat(this.reeHeight.getText().toString()));
    if (this.gender.equals("Male")) {
        this.ree_cal = Float.valueOf((float) Math.round((10.0f * this.ree_weight) + (6.25 *
this.ree_height * 100.0f) - (5.0f * this.age) + 5.0f));
    }
    else {
        this.ree_cal = Float.valueOf((float) Math.round((10.0f * this.ree_weight) + (6.25 *
this.ree_height * 100.0f) - (5.0f * this.age) - 161.0f));
    }
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();
    try {
        object.put("farmerId", Farmer.farmerId);
        object.put("farmerAge", Farmer.farmerAge);
        object.put("farmerWeight", this.ree_weight);
        object.put("farmerHeight", this.ree_height);
        object.put("ree",this.ree_cal);
        System.out.println(object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
    requestQueue.add(new JsonRequest(1, "http://10.0.2.2:8080/user/farmer/REE",
object, new Response.Listener<JSONObject>() {
        public void onResponse(JSONObject response) {
            name_ree = (TextView) findViewById(R.id.name_ree);
            name_ree.setText("REE");
            TextView textView = reeActivity.this.ree_value;
            textView.setText("" + reeActivity.this.ree_cal+" kcal/day");
        }
    }
});
}

```

```

    }
    }, new Response.ErrorListener() {
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(reeActivity.this.getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
        }
    }));
}

public void sam(View view) {
    startActivity(new Intent(this, samActivity.class));
}
}

```

For TEE calculation:

```

package com.example.nutrienergy;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

public class teeActivity extends AppCompatActivity {
    EditText pal_value;
    Float palValue;
    TextView name_tee;
    Float Tee_value;
    TextView ree_value;
    Float valueRee;
    TextView Teevalue;
    Spinner pal_spinner;
    String pal;
    Button get_tee;
    public void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView((int) R.layout.activity_tee);
((TextView) findViewById(R.id.farmer)).setText(Farmer.farmerName);

this.name_tee = (TextView) findViewById(R.id.name_tee);
this.Teevalue = (TextView) findViewById(R.id.Teevalue);
this.ree_value= (TextView) findViewById(R.id.ree_value);
this.pal_spinner = (Spinner) findViewById(R.id.pal_spinner);
this.get_tee = (Button) findViewById(R.id.get_tee);
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.pal_values, android.R.layout.simple_spinner_item);

this.pal_spinner.setAdapter(adapter);
this.pal_spinner.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view, int position, long id)
    {
        teeActivity.this.pal = (String) parent.getItemAtPosition(position);
        System.out.println(teeActivity.this.pal);
    }

    public void onNothingSelected(AdapterView<?> adapterView) {
    }
});
}

public void get_tee(View view) {
    if (this.pal_spinner.equals("Physical Activity Status") ||
this.ree_value.getText().toString().isEmpty() ) {
        Toast.makeText(getApplicationContext(), "Please fill all the field...",
Toast.LENGTH_SHORT).show();
        return;
    }

    if (teeActivity.this.pal.equals("Sedentary") )
    {
        this.palValue = 1.2f;
    }
    else if (teeActivity.this.pal.equals( "Light Activity"))
    {
        this.palValue = 1.375f;
    }
    else if (teeActivity.this.pal.equals("Moderate"))
    {
        this.palValue = 1.55f;
    }
    else if (teeActivity.this.pal.equals("Very Active"))
    {
        this.palValue = 1.725f;
    }
}

```

```

    }
    else if (teeActivity.this.pal.equals("Extra Active"))
    {
        this.palValue = 1.9f;
    }
    System.out.println(this.palValue);

    this.valueRee = Float.valueOf(Float.parseFloat(this.ree_value.getText().toString()));
    this.Tee_value = Float.valueOf((float) Math.round(this.palValue * this.valueRee));

    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    JSONObject object = new JSONObject();
    try {
        object.put("farmerId", Farmer.farmerId);
        object.put("pal", this.palValue);
        object.put("ree", this.valueRee);
        object.put("tee",this.Tee_value);
        System.out.println(object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    System.out.println(object);
    requestQueue.add(new JsonRequest(1, "http://10.0.2.2:8080/user/farmer/TEE",
object, new Response.Listener<JSONObject>() {
    public void onResponse(JSONObject response) {
        TextView textView = teeActivity.this.Teevalue;
        textView.setText("" + teeActivity.this.Tee_value+" kcal/day");
        name_tee.setText("TEE");
    }
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(teeActivity.this(getApplicationContext(), "Network Error...",
Toast.LENGTH_SHORT).show();
    }
})));
}

public void sam(View view) {
    startActivity(new Intent(this, samActivity.class));
}
}

```

Controller Code in REST API

```

package com.NutriEnergy.NutriEnergy.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.GetMapping;

```

```
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
```

```
import com. NutriEnergy.NutriEnergy.dao.BMIDao;
import com. NutriEnergy.NutriEnergy.dao.CCDao;
import com. NutriEnergy.NutriEnergy.dao.FarmerDao;
import com. NutriEnergy.NutriEnergy.dao.MDDDao;
import com. NutriEnergy.NutriEnergy.dao.MUACDao;
import com. NutriEnergy.NutriEnergy.dao.PFIDao;
import com. NutriEnergy.NutriEnergy.dao.SFTDao;
import com. NutriEnergy.NutriEnergy.dao.VO2Dao;
import com. NutriEnergy.NutriEnergy.entities.BMI;
import com. NutriEnergy.NutriEnergy.entities.CC;
import com. NutriEnergy.NutriEnergy.entities.Farmer;
import com. NutriEnergy.NutriEnergy.entities.MDD;
import com. NutriEnergy.NutriEnergy.entities.MUAC;
import com. NutriEnergy.NutriEnergy.entities.PFI;
import com. NutriEnergy.NutriEnergy.entities.SFT;
import com. NutriEnergy.NutriEnergy.entities.Status;
import com. NutriEnergy.NutriEnergy.entities.User;
import com. NutriEnergy.NutriEnergy.entities.VO2;
import com. NutriEnergy.NutriEnergy.services.BMIService;
import com. NutriEnergy.NutriEnergy.services.CCService;
import com. NutriEnergy.NutriEnergy.services.FarmerService;
import com. NutriEnergy.NutriEnergy.services.MDDService;
import com. NutriEnergy.NutriEnergy.services.MUACService;
import com. NutriEnergy.NutriEnergy.services.PFIService;
import com. NutriEnergy.NutriEnergy.services.SFTService;
import com. NutriEnergy.NutriEnergy.services.UserService;
import com. NutriEnergy.NutriEnergy.services.VO2Service;
import com. NutriEnergy.NutriEnergy.dao.REEDao;
import com. NutriEnergy.NutriEnergy.dao.TEEDao;
import com. NutriEnergy.NutriEnergy.dao.BMRDao;
import com. NutriEnergy.NutriEnergy.entities.REE;
import com. NutriEnergy.NutriEnergy.entities.TEE;
import com. NutriEnergy.NutriEnergy.entities.BMR;
import com. NutriEnergy.NutriEnergy.services.REEService;
import com. NutriEnergy.NutriEnergy.services.TEEService;
import com. NutriEnergy.NutriEnergy.services.BMRService;
```

```
@RestController
```

```
public class UserController {
    @GetMapping("/home")
    public String home() {
        return "goddamn";
    }

    // controller for user table
    @Autowired
    private UserService userService;
    @GetMapping("/user")
    public List<User> getUser() {
        return userService.getUser();
    }
}
```

```

    }

    @PostMapping("/users/register")
    public User registerUser(@Validated @RequestBody User newUser) {
        return this.userService.registerUser(newUser);
    }
    @PostMapping("/users/login")
    public User loginUser(@Validated @RequestBody User user) {
        return this.userService.loginUser(user);
    }
}

// controller for farmer table
@Autowired
private FarmerService farmerService;
@GetMapping("/user/farmers")
public List<Farmer> getFarmer(){
    return this.farmerService.getFarmer();
}
@PostMapping("/user/farmer")
public Farmer addFarmer(@RequestBody Farmer farmer) {
    return this.farmerService.addFarmer(farmer);
}
@GetMapping("/user/farmers/{userId}")
public List<Farmer> getFarmerByuserId(@PathVariable long userId){
    return this.farmerService.getFarmerByuserId(userId);
}
@Autowired
private FarmerDao farmerDao;
@GetMapping("/user/farmer/{farmerId}")
public Farmer getFarmerByfarmerId(@PathVariable long farmerId){
    return this.farmerDao.getFarmerByfarmerId(farmerId);
}
//Controller for PFI table
@Autowired
private PFIService pfiService;
@PostMapping("/user/farmer/PFI")
public PFI addPFI(@RequestBody PFI pfi){
    return this.pfiService.addPFI(pfi);
}
@Autowired
private PFIDao pfiDao;

@GetMapping("/user/farmer/PFI/{farmerId}")
public List<PFI> getPFIByfarmerId(@PathVariable List<Long> farmerId){
    return this.pfiDao.getPFIByfarmerId(farmerId);
}
//Controller for VO2 table
@Autowired
private VO2Service vo2Service;

@PostMapping("/user/farmer/VO2")
public VO2 addVO2(@RequestBody VO2 vo2) {

```

```

        return this.vo2Service.addVO2(vo2);
    }
    @Autowired
    private VO2Dao vo2Dao;

    @GetMapping("/user/farmer/VO2/{farmerId}")
    public List<VO2> getVO2ByfarmerId(@PathVariable List<Long> farmerId){
        return this.vo2Dao.getVO2ByfarmerId(farmerId);
    }

    //Controller for BMI table
    @Autowired
    private BMIService bmiService;
    @PostMapping("/user/farmer/BMI")
    public BMI addBMI(@RequestBody BMI bmi) {
        return this.bmiService.addBMI(bmi);
    }

    @Autowired
    private BMIDao bmiDao;
    @GetMapping("/user/farmer/BMI/{farmerId}")
    public List<BMI> getBMIByfarmerId(@PathVariable List<Long> farmerId){
        return this.bmiDao.getBMIByfarmerId(farmerId);
    }

    //Controller for MUAC table
    @Autowired
    private MUACService muacService;
    @PostMapping("/user/farmer/MUAC")
    public MUAC addMUAC(@RequestBody MUAC muac) {
        return this.muacService.addMUAC(muac);
    }

    @Autowired
    private MUACDao muacDao;
    @GetMapping("/user/farmer/MUAC/{farmerId}")
    public List<MUAC> getMUACByfarmerId(@PathVariable List<Long> farmerId){
        return this.muacDao.getMUACByfarmerId(farmerId);
    }

    //Controller for CC table

    @Autowired
    private CCService ccService;

    @PostMapping("/user/farmer/CC")
    public CC addCC(@RequestBody CC cc) {
        return this.ccService.addCC(cc);
    }

    @Autowired
    private CCDao ccDao;

    @GetMapping("/user/farmer/CC/{farmerId}")
    public List<CC> getCCByfarmerId(@PathVariable List<Long> farmerId){
        return this.ccDao.getCCByfarmerId(farmerId);
    }
}

```

```

//Controller for SFT table

@Autowired
private SFTService sftService;

@PostMapping("/user/farmer/SFT")
public SFT addSFT(@RequestBody SFT sft) {
    return this.sftService.addSFT(sft);
}

@Autowired
private SFTDao sftDao;

@GetMapping("/user/farmer/SFT/{farmerId}")
public List<SFT> getSFTByfarmerId(@PathVariable List<Long> farmerId){
    return this.sftDao.getSFTByfarmerId(farmerId);
}

//Controller for MDD table

@Autowired
private MDDService mddService;

@PostMapping("/user/farmer/MDD")
public MDD addMDD(@RequestBody MDD mdd) {
    return this.mddService.addMDD(mdd);
}

@Autowired
private MDDDao mddDao;

@GetMapping("/user/farmer/MDD/{farmerId}")
public List<MDD> getMDDByfarmerId(@PathVariable List<Long> farmerId){
    return this.mddDao.getMDDByfarmerId(farmerId);
}

//controller for REE table

@Autowired
private REEService reeService;

@PostMapping("/user/farmer/REE")
public REE addREE(@RequestBody REE ree) {
    return this.reeService.addREE(ree);
}

@Autowired
private REEDao reeDao;

@GetMapping("/user/farmer/REE/{farmerId}")
public List<REE> getREEByfarmerId(@PathVariable List<Long> farmerId){
    return this.reeDao.getREEByfarmerId(farmerId);
}
//CONTROLLER FOR TEE TABLE

```

```

    @Autowired
    private TEEService teeService;

    @PostMapping("/user/farmer/TEE")
    public TEE addTEE(@RequestBody TEE tee) {
        return this.teeService.addTEE(tee);
    }

    @Autowired
    private TEEDao teeDao;

    @GetMapping("/user/farmer/TEE/{farmerId}")
    public List<TEE> getTEEByfarmerId(@PathVariable List<Long> farmerId){
        return this.teeDao.getTEEByfarmerId(farmerId);
    }
    //controller for bmr table
    @Autowired
    private BMRService bmrService;
    @PostMapping("/user/farmer/BMR")
    public BMR addBMR(@RequestBody BMR bmr) {
        return this.bmrService.addBMR(bmr);
    }
    @Autowired
    private BMRDao bmrDao;
    @GetMapping("/user/farmer/BMR/{farmerId}")
    public List<BMR> getBMRByfarmerId(@PathVariable List<Long> farmerId){
        return this.bmrDao.getBMRByfarmerId(farmerId);
    }
}
}

```