PREGNANCY ASSISTANT

A PROJECT REPORT

Submitted by

AHAMMED BASIM C A (ADR18CS002)

JOSEPH PIOUS (ADR18CS021)

NAVNEETH CHANDRAN (ADR18CS030)

NIVYA SARA REJI (ADR18CS031)

in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

at

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING ADOOR

2021-22

PREGNANCY ASSISTANT

A PROJECT REPORT

Submitted by

AHAMMED BASIM C A (ADR18CS002)

JOSEPH PIOUS (ADR18CS021)

NAVNEETH CHANDRAN (ADR18CS030)

NIVYA SARA REJI (ADR18CS031)

in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

at

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING ADOOR

2021-22

COLLEGE OF ENGINEERING ADOOR MANAKKALA P O,

ADOOR, PATHANAMTHITTA, KERALA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

Certified that this is a bonafide record of the project report entitled

PREGNANCY ASSISTANT

done by

AHAMMED BASIM C A (ADR18CS002)
JOSEPH PIOUS (ADR18CS021)
NAVNEETH CHANDRAN(ADR18CS030)
NIVYA SARA REJI (ADR18CS031)

during the year 2021-2022 in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of APJ Abdul Kalam Technological University, Kerala.

PROJECT GUIDE

Prof. Ajeesh S
Assistant Professor
Dept Of CSE, CEA

PROJECT CO-ORDINATOR

Prof. Honey Mol M
Assistant Professor
Dept Of CSE, CEA

HEAD OF THE DEPARTMENT

Prof. SHIBU J
Associate Professor
Dept Of CSE, CEA

EXTERNAL EXAMINER

DECLARATION

We undersigned hereby declare that the project report “PREGNANCY ASSISTANT”, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Mr. Ajeesh S. This submission represents our ideas in our own words and where ideas or words of others have been included wehave adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Adoor

Date :

AHAMMED BASIM C A
JOSEPH PIOUS
NAVNEETH CHANDRAN
NIVYA SARA REJI

ACKNOWLEDGEMENT

First and foremost, we thankfully acknowledge our Principal **Prof. Dr. Sunilkumar** K for giving me an opportunity to present this seminar.

The constant encouragement and timely support rendered by our Head of Department, **Prof. Shibu** J is deeply remembered.

We express our heartfelt gratitude to our seminar guide, **Prof.** Ajeesh S, Assistant Professor, Department of Computer Science and Engineering, for his valuable guidance, support and encouragement during the course of the seminar and in the preparation of the report. We have greatly benefited from his experience and knowledge.

We are extremely grateful to our project coordinator, **Prof. Honey Mol M,** Assistant Professor, Department of Computer Science and Engineering, for all the help and co-ordination extended in bringing out this seminar successfully in time.

The help extended by all other staff members of the department are remembered with gratitude.

We also remember with thanks to all our friends and well-wishers for their encouragementand support.

Above all, we would like to express our profound gratitude to God Almighty for the immense blessings upon me that led to the successful completion of this project.

AHAMMED BASIM C A
JOSEPH PIOUS
NAVNEETH CHANDRAN
NIVYA SARA REJI

ABSTRACT

Pregnancy, also known as gestation, is the time during which one or more offspring develops inside a woman's womb. It is the time when she is to be taken care of. An obstetrician use their specialist training and skills in caring for mothers and babies during pregnancy , but apart than this self care must be taken to have a healthy pregnancy. The health of the child especially the weight of the fetus is also of due importance. Recently, there are many mobile application that act as a personal health assistant. But an application that can help a woman throughout her pregnancy are a few. So with the aid of mobile technology and machine learning, the project proposes a machine learning model implemented as an android application which helps the mother and her fetus.

Keywords*: Low Birth Weight,Pregnancy, Machine Learning, Random Forest, Android*

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

LBW	Low Birth Weight
ML	Machine Learning
SVM	Support Vector Classifier
KNN	K-Nearest Neighbour
SQL	Structured Query Language
IDE	Integrated Development Environment
API	Application Programmable Interface
PHP	Hypertext Pre-Processor

LIST OF TABLES

1. INTRODUCTION

1.1 GENERAL BACKGROUND

Healthcare is one of the most pressing concerns facing our country and the rest of the world in the 21st century. As healthcare costs soar ever higher, industry professionals have been looking for ways to keep these costs at bay. One such method has been the use of technology. Technology has been a core player in helping improve healthcare, by reducing costs, improving patient safety and satisfaction, saving time and effort for both patients and clinics alike, and reducing potential errors. This shift in paradigm is, in part, due to the fact that technology can help automate a lot of manual processes and procedures involved in healthcare practices. In recent years, mobile technology, in particular, has greatly evolved and become a mature platform, and according to CBS News , there are around 4.6 billion mobile phones worldwide. Because of this, many health organisations such as the World Health Organisation and experts from across the healthcare industry are looking to tap into the potential of mobile technology to revolutionise healthcare management and delivery.

As mobile technology, specifically smartphones, becomes more widespread, we believe that many of our daily activities will be performed through interactions with our smartphones. This is also true of self-care. Many healthcare providers will jump onto the mobile bandwagon to realize the potential savings and efficiencies by having users take matters into their own hands, through the use of their smartphones. As we move forward into the next decade, mobile technology will play an essential role in revolutionising the healthcare industry.

It is in this regard that we look to employ mobile technology to help pregnant women more effectively manage and keep track of their health during their pregnancy. A large percentage of women of child-bearing age are internet-connected, either through their computers and/or handheld devices, and welcome the idea of utilising tools such as the Internet. People carry their smartphones wherever they go, which allows them to be connected to the Internet at all times.

Low Birth Weight (LBW) is the major problem for the newborn . LBW is a term used to describe babies who are born weighing less than 5 pounds, 8 ounces (2,500 grams). LBW babies are more likely than babies with normal weight to have health problems as a newborn. Almost 40 percent of the newborn suffer from underweight. Predicting birth weight before the birth of the baby is the best way to help the baby get special care as early as possible. It helps us to arrange for doctors and special facilities before the baby is born. The project implements ML based LBW model. There are three types of ML techniques : supervised,unsupervised and reinforcement. Random Forest which comes under supervised ML model is implemented in this project.

1.2 OBJECTIVES

Objectives of this project are:

- ➔ To design a machine learning model implemented as an android application which consists of :
  - ◆ A supervised machine learning model which predicts the low birth weight of the foetus.
  - ◆ Personalised food and exercise information to be followed by the pregnant woman during every week of her pregnancy along with reminders.

2. LITERATURE REVIEW

Based on the objectives of the project work, a comprehensive review of literature is carried out in the field of low birth weight prediction and study of various algorithms have been done.The literature reviewed is categorised into the following areas:

2.1 PREGNANCY

In humans, pregnancy lasts 40 weeks and, between conception and childbirth there are three phases, or trimesters, of pregnancy. As the baby develops, see Figure 2.1, and the expecting mother moves from one trimester to the next there are numerous physical changes and behavioural recommendations.

The first trimester covers the time from conception through the twelfth week and has a number of special considerations and changes. This is the period of time during much of the initial development of the fetus occurs. During this time the baby will start to develop many parts of its anatomy including the skin, circulatory system, bladder, head, toes, neck, fingernails, and genitals. This is also the part of the pregnancy when the mother goes through the most hormonal changes, which can lead to a number of unpleasant symptoms including morning sickness or nausea, swollen breasts, food aversions or cravings, fatigue, headaches, constipation, mood swings, and dizziness.

The second trimester covers week 13 through week 28. During this time the baby starts to take a more definite form. For this trimester, the baby develops many more parts of its core anatomy, including bones and fat. As a result of this growth, the baby develops more definite human characteristics such as hair, fingernails, and fingerprints. In addition to these developments, the gender of the baby can be determined, usually around the 14th week of pregnancy, and by the 18th week the baby is capable of forming facial expressions and hearing.

The third trimester lasts from week 28 until the child is born. This last stage of pregnancy is when many intense changes occur. The fetus continues to develop physically, and during this trimester the baby will develop eyelids and eyelashes, and finish developing bones and sexual organs. In addition to these developments, the baby will prepare for life after birth by practising breathing, kicking, grasping, and even constricting the pupils in its eyes to react to changes in light. After these final developments occur, the baby will begin to rapidly gain weight and change its orientation for delivery.

Fig 2.1(a): Fetal Growth from 8 to 40 weeks

2.1.1 RESOURCES

The pregnancy resources are the general health information from websites. Websites such as WebMB.com, MedicineNet.com, BabyCenter, and Everyday Health have pregnancy portals that allow users to access a variety of articles and tools for free. The most common areas covered by these websites are information about getting pregnant, basic pregnancy information, and labor/birth information. Most of this information is broken down into first, second, and third trimesters. These portals also include photo slideshows, newsletters, fitness/nutrition information, and other health-related information. These resources also offer various tools women can use. These include due date calculators, ovulation calculators, and pregnancy calendars. Of these types of websites, WebMD appears to be the most up-to-date and complete- even offering expert blogs and a social community larger than most available. The

important aspect of these sites is that they are part of a larger, commercial health company. As a result of this, the sites advertise very heavily. While this may not affect the quality of the information provided, it does leave some questions as to the slant the site is taking. These sites do offer very credible information with content by real doctors.

2.2 LOW BIRTH WEIGHT PREDICTION

In 2015, 20.5 million newborns, an estimated 14.6 per cent of all babies born globally that year, suffered from low birth weight. The estimates from the UNICEF-WHO study was shown in Fig. 2.2(a). These babies were more likely to die during their first month of life and those who survived to face lifelong consequences including a higher risk of stunted growth, lower IQ, and adult-onset chronic conditions such as obesity and diabetes. To grow a healthy baby, mothers need good nutrition and rest, adequate antenatal care, and a clean environment. Together, these ingredients for a healthy pregnancy can help to prevent, identify and treat the conditions that cause low birth weight and thus foster the achievement of the World Health Assembly (WHA) nutrition target to reduce low birth weight by 30 per cent between 2012 and 2025.

A survey says almost 40 percent of newborns suffer from being underweight. An underweight baby is more likely susceptible to many health problems than a baby with a normal weight. Special care and facilities are to be given to them when they are born so that their health condition will be improved. Predicting the weight before the birth of the baby and arranging doctors and facilities if the weight of the baby is less than 5 pounds and 8 ounces helps the baby get better treatment. The solution is to create a machine learning model that predicts the baby's weight from major factors like gestation weeks, mother’s age, gender and plurality of baby.

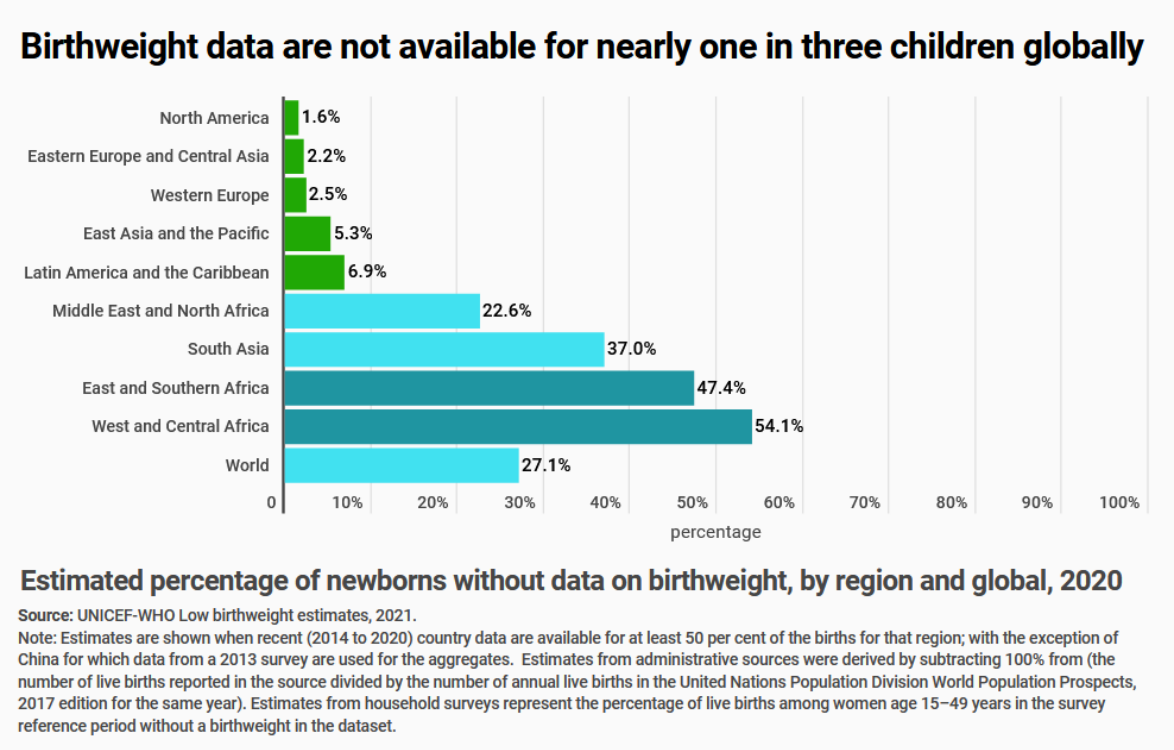

Fig 2.2(a) Low Birth Weight Estimates by UNICEF-WHO

2.3 MACHINE LEARNING ALGORITHM

The Dataset used is from the website and it has 666 rows with 23 columns. We have refined the dataset by dataset preprocessing. The dataset was divided into training and test data randomly. The training set comprises 75% of the data. Whereas, the test dataset comprises the remaining 25%. After the dataset creation, we considered some machine learning algorithms for the development of the model to perform low birth weight prediction.

Logistic Regression, Random Forest Classifier, K Nearest Neighbour, Support Vector Classifier, Decision Tree Classifier and Gradient Boosting Classifier algorithms are considered for the primary evaluation. A detailed evaluation report was given on the fig. 2.2. Hence, after the evaluation Random Forest approach has a very good performance for both prediction and classification of the Low Birth Weight data set.

	Model	Accuracy	Precision	Sensitivity	Specificity
0	Random Forest	0.862275	0.868056	0.968992	0.500000
1	logistic	0.772455	0.775758	0.992248	0.026316
2	KNN	0.754491	0.793333	0.922481	0.184211
3	SVC	0.772455	0.772455	1.000000	0.000000
4	CART	0.848679	0.921260	0.906977	0.736842
5	GBM	0.856287	0.862069	0.968992	0.473684

Table 2.3(a): Comparison of different machine learning algorithms

2.4 SUMMARY

With more people than ever connected to the internet wherever they go with the many smartphones and other mobile devices on the market, it makes sense to develop solutions for tasks that were previously cumbersome. While the healthcare industry has widely accepted mobile technology, this project falls into a niche. Different applications and websites have provided a multitude of tools and a wealth of information, but there has yet to be a specific assistant for the special category, which consists of pregnant women. Our goal is to design and implement a Pregnancy Assistant to maintain the better health of mothers and newborn children.

3. RANDOM FOREST

3.1 BACKGROUND OF RANDOM FOREST

The Random forest is an ensemble method which was developed by Leo Breiman in 2001. Breiman states 'The generalisation error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them'. The early development of Breiman's notion of random forests was influenced by the work of Amit and Geman who introduced the idea of searching over a random subset of the available decisions when splitting a node, in the context of growing a single tree.

3.2 WHAT EXACTLY IS A RANDOM FOREST?

In a random forest, many decision tree models are built and the final prediction is made based on the average of those independent decision tree predictions. To improve the performance of the model output, the random forest algorithm will use different subsets of data for each decision tree and also different subsets of the features. This ensures that there is no bias in the model as there is no particular segment of data or a set of features that dominates the prediction results. It also ensures that each model is different from one another otherwise there isn't much benefit to having so many models. If the individual models were built using the same dataset and the same features then no matter how many different models we build the predictions would not be much different from each other. Random forest uses an effective method called bootstrap aggregation, as well as the random subspace method to grow individual trees to achieve an extremely powerful aggregated predictor, capable of classification and regression, with better generalisation error than an individual decision tree.

Bootstrap aggregating or Bagging is a method that is used to generate multiple versions of a predictor and use the predictors to achieve an aggregated predictor. Randomness is introduced into each predictor by making bootstrap replicates — randomly sampling instances from the training

set with replacement — and using this as the new bootstrap data as the training set. The instances are being replaced back into the original training set when they are randomly sampled into the bootstrap set, therefore it is likely that some instances would occur more than once in the bootstrap training set that the predictor is trained on. The aggregated predictor, in this instance is our Random forest, outputs the relative majority of the predictions of each model.

Random subspace method is used as a means of increasing the generalisation accuracy of decision tree classifiers without trading away accuracy on training data by reducing the correlation between the predictive models in an ensemble. This method is similar to bagging in the sense that each predictive model within the ensemble is trained on a random sample of features with replacement, instead of training the model on all the features. This technique allows for the predictors to not focus all of its attention on the features that appear more predictive than others.

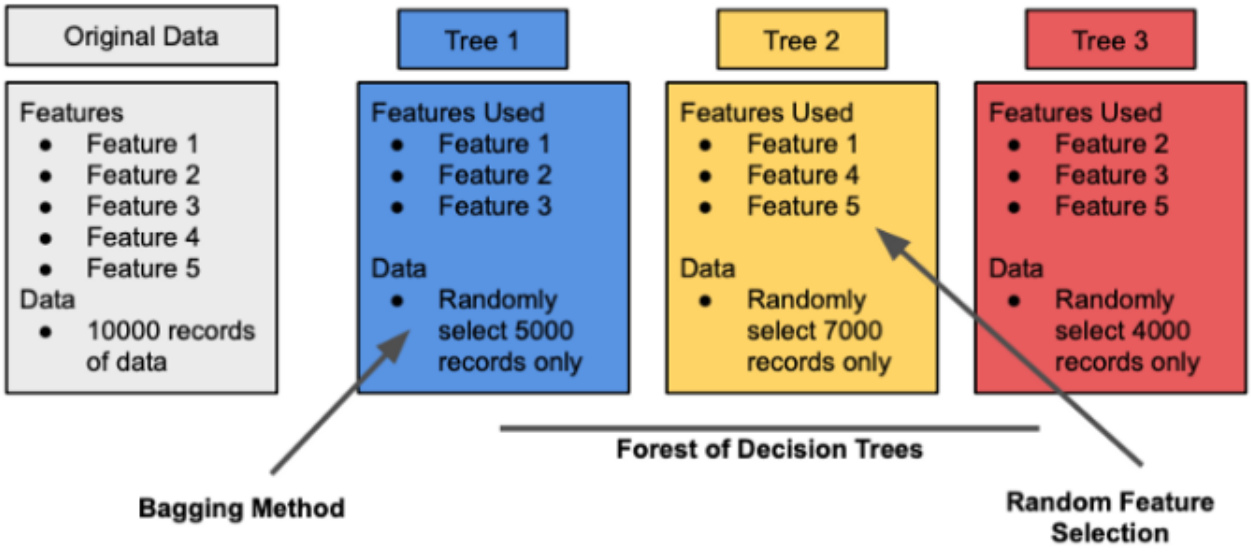

Fig 3.2(a): Random Forest

3.3 WHY IS IT CALLED RANDOM FOREST?

It is called a Random Forest because we use Random subsets of data and features and end up building a Forest of decision trees (many trees). Random Forest is also a classic example of a bagging approach as we use different subsets of data in each model to make predictions.

3.4 CHARACTERISTICS OF RANDOM FOREST

- ➔ The final model prediction in the random forest is based on aggregated predictions from all the individual decision tree models and hence overfitting is avoided.
- ➔ Random forest algorithm is made up of many decision trees, the mechanism behind decision trees also helps in identifying features that are important in making predictions hence naturally random forest is good at the variable selection.
- ➔ Unlike many other algorithms (like linear regression, SVM, etc.) random forest doesn't make any assumption about the data or its distribution. Hence it generally requires minimal data transformations.
- ➔ Random forest algorithm makes use of random subsets of features and hence it can perform quite well with a high dimensional dataset (a dataset with a large number of features).
- ➔ Generated forests can be saved for future use on other data.
- ➔ It can handle thousands of input variables without variable deletion.
- ➔ It gives estimates of what variables are important in the classification.

3.5 APPLICATIONS OF RANDOM FOREST

Healthcare and Medicine

- ➔ Cardiovascular Disease Prediction
- ➔ Diabetes Prediction

Stock Market

- ➔ Stock Market Prediction
- ➔ Stock Market Sentimental Analysis

E-Commerce

- ➔ Product Recommendation
- ➔ Price Optimisation

4. ANDROID APPLICATION

4.1 BACKGROUND OF ANDROID APPLICATION

Android began in 2003 as a project of the American technology company Android Inc. which was founded in Palo Alto, California. Rich Miner, Nick Sears, Chris White, and Andy Rubin were its four founders, at that time they were developing an operating system for digital cameras. In 2004 the project changed to become an operating system for smartphones. Android IncIn 2005, the next significant chapter in Android history began when the original company was acquired by Google. Rubin and other founding members continued developing the OS under their new owners. They then decided to use Linux as the basis for the Android OS. That made it possible to offer the operating system to third-party mobile manufacturers for free. Google and the Android team felt the company could profit from providing other services, including apps.

4.2 WHAT EXACTLY IS ANDROID APPLICATION?

Android App is software designed to run on an Android device or emulator. The term also refers to an APK file which stands for Android package. This file is a Zip archive containing app code, resources, and meta information.

Applications ("apps"), which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, Kotlin programming language, which replaced Java as Google's preferred language for Android app development in May 2019 and was originally announced in May 2017. Java is still supported (originally the only option for user-space programs, and is often mixed with Kotlin), as is C++. Java or other JVM languages, such as Kotlin, may be combined with C/C++ together with a choice of non-default runtimes that allow better C++ support. The Go programming language is also supported, although with a limited set of application programming interfaces (API).

Android has a growing selection of third-party applications, which can be acquired by users by downloading and installing the application's APK (Android application package) file, or by

downloading them using an application store program that allows users to install, update, and remove applications from their devices. Google Play Store is the primary application store installed on Android devices that comply with Google's compatibility requirements and licence the Google Mobile Services software. Google Play Store allows users to browse, download and update applications published by Google and third-party developers; as of January 2021, there are more than three million applications available for Android in Play Store. As of July 2013, 50 billion application installations had been performed. Some carriers offer direct carrier billing for Google Play application purchases, where the cost of the application is added to the user's monthly bill. As of May 2017, there are over one billion active users a month for Gmail, Android, Chrome, Google Play and Maps.

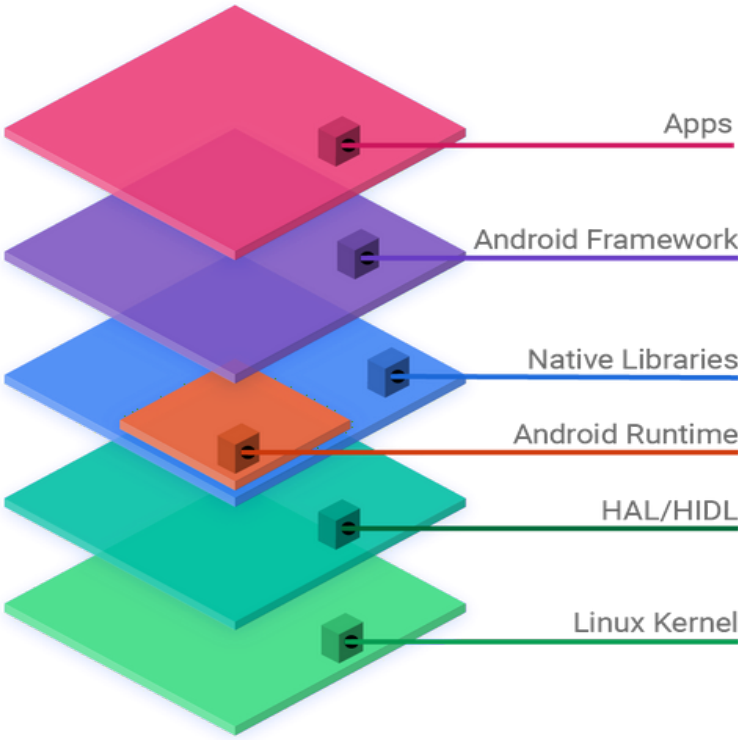

Fig 4.2(a): Android Software Development Stages

4.3 CHARACTERISTICS OF ANDROID APPLICATION

- ➔ Simple and User-Friendly Interface, which means that anyone even without having prior experience in handling smartphones can operate it with ease
- ➔ One of the critical advantages of Android app development is the easy availability of the Android SDK. The development teams can use the material design from these SDKs to build interactive apps. However, the developers/development teams are required to pay a one-time registration fee for application distribution. After that, they can leverage any computer device to build and test the product for their smartphones, ensuring low investment and increased user engagement. In turn, the end-users benefit from an interactive app, and the enterprise gains a higher return on investment.
- ➔ The android application offers to streamline the business process by improving the communication between the employees and different teams. Secondly, it also focuses on providing robust security, which means that your data is safe.
- ➔ The android application is improving by the user feedback, it is supposed to be one of the most crucial elements of the android application. The more feedback received from the potential audience, the more scope you would get to improve the app functionalities. It is a necessary part of the job to evaluate the performance of the Android app at regular intervals. This is also a nice and positive gesture from the app owner and developer showing his concerns about the users.
- ➔ Support for multiple languages, the android app provides support for the majorly spoken international languages such as English, French, Chinese, Spanish, Arabic, etc.
- ➔ Android application is highly compact which means that the app is less in size with high functionality and features.

4.4 APPLICATIONS OF ANDROID APPLICATION

Healthcare

- ➔ Exercise, diet recommendation

Education

- ➔ Learning apps
- ➔ Integrated institution work app

E-Commerce

- ➔ Online shopping
- ➔ Online payment
- ➔ Banking

5. SYSTEM ANALYSIS

5.1 EXISTING SYSTEM

Nowadays most people are facing lots of issues with their health. Lifestyle choices are contributing factors to poor health in many cases. These include smoking cigarettes, and can also include a poor diet, whether it is overeating or an overly constrictive diet. Inactivity can also contribute to health issues and also a lack of sleep, excessive alcohol consumption, and neglect of oral hygiene. Health care is a basic need of every human being. So, there should be better health care facilities in our society. The existing health care system has better exposure to the usage of technologies. It has the application of organised knowledge and skills in the form of devices, medicines, vaccines, procedures, and systems developed to solve health problems and improve the quality of life.

“Healthify Me” applications and other personal healthcare assistant facilities are providing better healthcare services to people using technology. But, these all facilities are covered in the category of general people in our society. But, the existing system has some drawbacks to improving the facility of the healthcare system.

Now in health care facilities, most of the applications are concerned with the general category. Hence it is not focused on a particular group with special interest. So, the pregnant women group also faces these drawbacks in the healthcare system.

5.2 PROPOSED SYSTEM

Prediction of foetus health is important, especially for pregnancies with a high risk of complications where special treatment is needed. At a population level, the proportion of infants with low birth weight is an indicator of a multifaceted public health problem that includes long-term maternal malnutrition, ill health and poor health care during the pregnancy period. Low birth weight remains a significant public health problem in many developing countries, and poor nutrition both before and during pregnancy is recognized as an important cause.

We propose a system that only focuses on pregnant women and their health. In this system, we integrated low birth weight prediction, food recommendation, exercise information and reminder functionality in an Android Application (Care'S). This app helps pregnant women to go through a healthy pregnancy period and it helps to predict whether the foetus is low birth or not, by this, it is much easier for the mother to take care of the foetus and herself.

6. SYSTEM OVERVIEW

6.1 SYSTEM DESCRIPTION

Our system is a Pregnancy Assistant and it is centred around the four pillars. Those pillars are Android Application, ML Model, Database and the User. Android application is used to act as an interface between the ML model and the User. ML Model is used for predicting the Low Birth Weight of a newborn child. The database contains details of every user.

6.2 SYSTEM STRUCTURE

Initially users access the Android application from a smartphone. Through the application, the system collects the input details from the user. These details were stored in the database. When the user needs the predicted output, the application calls the ML model. ML Model was completely trained and tested with the required datasets. Then, the model will fetch the corresponding datasets of the user from the database. Then the result from the ML model will be stored in the database and presented to the user through the android application. The system operates with this structure and is in fig. 6.2.1. The connection between the Android Application & Database, Database & ML Model and ML Model and Android Application were established by the respective APIs.

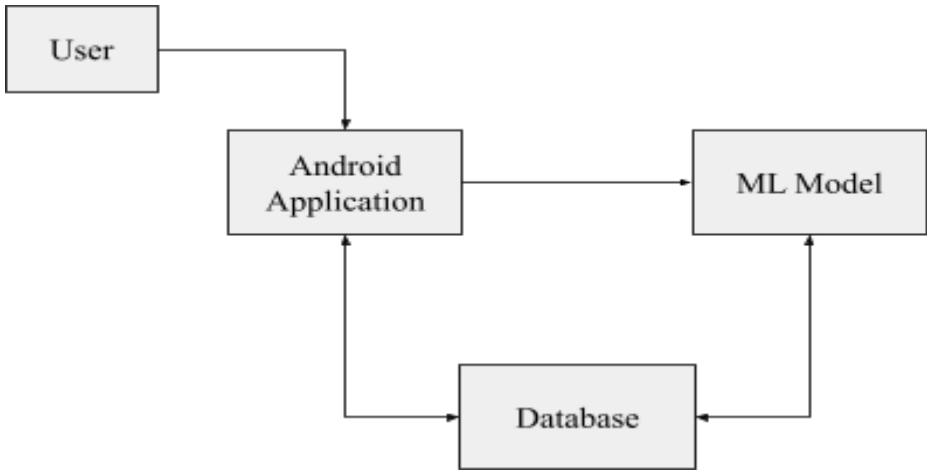

Fig. 6.2(a): System Structure

7. SYSTEM DESIGN

7.1 MODULE 1

7.1.1 SKELETON DESIGN

Skeletons are a way to communicate to the user that the content is loading while at the same time providing a close preview of what the content will more or less look like. Different types of navigation bars are available at the android studio like lateral navigation, reverse navigation etc. Among these, this project follows the bottom navigation bar. Bottom navigation is chosen because:

- ➔ It allows the user to switch among different fragments/activities easily.
- ➔ It makes the user aware of different screens available in the app.
- ➔ The user can check which screen they are on at the moment.

Bottom navigation activities are implemented using Fragments. Fragments have their lifecycle and layouts or UI components. Fragments help enrich the UI design, pass data between different screens, and adapt to different device configurations. The different fragments used in this project are Home, Dashboard, Notification and User Fragments. The home fragment is shown in Fig7.1.1(a).

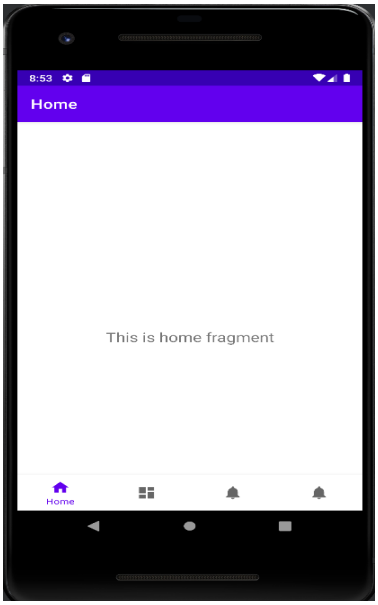

Fig 7.1.1 (a) Home Fragment

7.1.2 CONNECTING DATABASE

Database connecting was a primary task completed by our project. Initially, a table was created in the SQL database and assigned feature titles such as id, username, email and password where id acts as the primary key. This table contains the tags of the primary features collected from the user. This connection was established for linking the API through the PHP scripting language.

7.2 MODULE 2

7.2.1 DATA COLLECTION OF PREGNANCY PERIOD

Data collection is the process of gathering and measuring information on targeted variables, which then enables one to answer relevant questions and evaluate outcomes. Ensuring adequate nutrition and exercise during pregnancy is one of the key actions to health for both mother and child. In order to understand the food intake and exercise followed data is been collected from various sources such as articles, health magazines, and internet sources and these are analysed and studied. Some of the collected data are shown in Fig 7.2.1 (a)

Meats*
Lean cuts of:
beef
ham
lamb
pork
veal

Game meats:
bison
rabbit
venison

Lean ground meats:
beef
pork
lamb

Lean luncheon meats
Organ meats:
liver
giblets

Poultry*
chicken
duck
goose
turkey
ground chicken and turkey

Eggs*
chicken eggs
duck eggs

Dry beans and peas:
black beans
black-eyed peas
chickpeas (garbanzo beans)
falafel
kidney beans
lentils
lima beans (mature)
navy beans
pinto beans
soy beans
split peas
tofu (bean curd made from soy beans)
white beans

bean burgers:
garden burgers
veggie burgers

tempeh
texturized vegetable protein (TVP)

Nuts & seeds*
almonds
cashews
hazelnuts (filberts)
mixed nuts
peanuts
peanut butter
pecans
pistachios

Fish*
Finfish such as:
catfish
cod
flounder
haddock
halibut
herring
mackerel
pollock
porgy
salmon
sea bass
snapper
swordfish
trout
tuna

Shellfish such as:
clams
crab
crayfish
lobster
mussels
octopus
oysters
scallops
squid (calamari)
shrimp

Canned fish such as:
anchovies
clams

Whole grains:
brown rice
buckwheat
bulgur (cracked wheat)
oatmeal
popcorn

Ready-to-eat breakfast cereals:
whole wheat cereal flakes
muesli

whole grain barley
whole grain cornmeal
whole rye
whole wheat bread
whole wheat crackers
whole wheat pasta
whole wheat sandwich buns and rolls
whole wheat tortillas
wild rice

Less common whole grains:
amaranth
millet
quinoa
sorghum
triticale

Refined grains:
cornbread*
corn tortillas*
couscous*
crackers*
flour tortillas*
grits
noodles*

*Pasta**
spaghetti
macaroni

pitas*
pretzels

Ready-to-eat breakfast cereals
corn flakes

white bread
white sandwich buns and rolls
white rice.

Fig 7.2.1(a) Data Collection

7.3 MODULE 3

7.3.1 ADDING FUNCTIONALITIES

Other than the prediction of Low Birth Weight of the fetus the android application provides:

Food Information: The various foods to be consumed by the pregnant woman are provided on a week-to-week basis.

Exercise Information: The various exercises to be followed by the pregnant woman are provided on a week-to-week basis.

Notification System: Gives reminders for checkups one day prior to the appointment scheduled day.

7.4 MODULE 4

7.4.1 DATASET CREATION

The Dataset used for the LBW Model initially had 666 rows with 75 columns where columns represent the features. Through dataset preprocessing, these 75 columns have been refined and renamed to 22. These 22 features are used for the prediction of low birth weight. They are:

parity = order of gravida
mother_age = mother's age
mother_height = mother's height
mid_arm_cir = mid-arm circumference
workload = taking rest or having workload at different stages of pregnancy
anaemia = is anaemic or not
asthma = is an asthma patient or not
bad_obs_history = had an abortion or not
injection = has taken injection or not
falif = is diabetic or not
iron = whether taken the iron tonic or not
2nd_workload = taking rest or having heavy workload at different stages of pregnancy
1st_convulsion = had convulsion or not
bleed = having bleeding or not
asthma2 = whether an asthma patient or not
inject2 = whether taken injection or not
2nd_convulsion = had convulsion or not

bleed1 = having bleeding or not
fever = having fever or not
BMI = Body Mass Index

7.4.2 MODEL DEVELOPING

The LBW Model has been developed using the Random Forest algorithm which comes under supervised machine learning technique. We performed analysis using logistic regression to obtain insight into the association of each potential determinant with LBW and to select potential predictors for multivariable analysis The model has been developed with the help of python language. As the pregnant women came from different clusters, individual data were likely to be clustered within the different kebeles, which could affect the association of the predictors with low birth weight. Main.py specifies the detailed algorithm code of LBW model development.

7.5 MODULE 5

7.5.1 TRAINING AND TESTING OF MODEL

From the created dataset, 75 % of the dataset is used for training the ML model and 25 % of the dataset is used for testing the ML model. Since the random forest is used, bootstrapping - random sample with replacement is followed. The numerical features are mother_age and gestation_weeks. The categorical features are is_male and gestation_weeks. One hot encoding is applied to categorical columns. The wide model works well on categorical features. Hence numeric features are Bucketized and passed to a wide model. On the other hand deep model works well on numeric features. Wide features are crossed, embedded and passed as additional input to the deep model. An input_fn is created to return a batch of examples for training for each invocation. It identifies files that match the filename pattern and shuffles them before retrieving examples.

7.6 MODULE 6

7.6.1 CREATE API OF LBW

Here the Flask framework is used for creating the API of the LBW model of our system. Flask is a widely used micro web framework for creating APIs in Python. It is a simple yet powerful web framework which is designed to get started quickly and easily, with the ability to scale up to complex applications. REST stands for REpresentational State Transfer and is an architectural style used in modern web development. It defines a set of rules/constraints for a web application to send and receive data. It is now growing as the most common method for connecting components in a microservice architecture. APIs will enable you to get or send data to the model and perform some action to get your task done from the application. Flask Restful is an extension for Flask that adds support for building REST APIs in Python using Flask as the back-end. It encourages best practices and is very easy to set up. Flask restful is very easy to pick up if you're already familiar with flask.

In flask_restful, the main building block is a resource. Each resource can have several methods associated with it such as GET, POST, PUT, DELETE, etc. for example, there could be a resource that calculates the square of a number whenever a get request is sent to it. Each resource is a class that inherits from the Resource class of flask_restful. Once the resource is created and defined, we can add our custom resource to the API and specify a URL path for that corresponding resource. Here app.py is used for creating the API of the LBW model.

7.7 MODULE 7

7.7.1 GENERATE OUTPUT

The output was presented through the android application's home page. Initially, the user gives the input details to the database through the android application. Then the ML model predicted the output results regarding the LBW details of the fetus. The ML model fetched the respective user's input details from the database and used them for prediction.

8. SYSTEM WORKFLOW

8.1 ML MODEL WORKFLOW

The workflow is a series of systematic steps of a particular process. The ML workflow can be in various forms, but it generally consists of four steps, i.e. data exploration, data cleaning, model building, and presenting the results. Our System workflow was mainly divided into two parts and one of those parts was the ML Model Workflow. ML Model was initially triggered by the user through the button named “Predict” in the android application. When the button is triggered by the user, the ML model collects the input dataset from the database. The respective user’s input data was already collected through the android application and stored in the database. ML Model fetches those data as input. After the input collection, the model predicts output with these datasets and it is stored in the database. Fig. 8.1(a) contains the ML Model Workflow diagram.

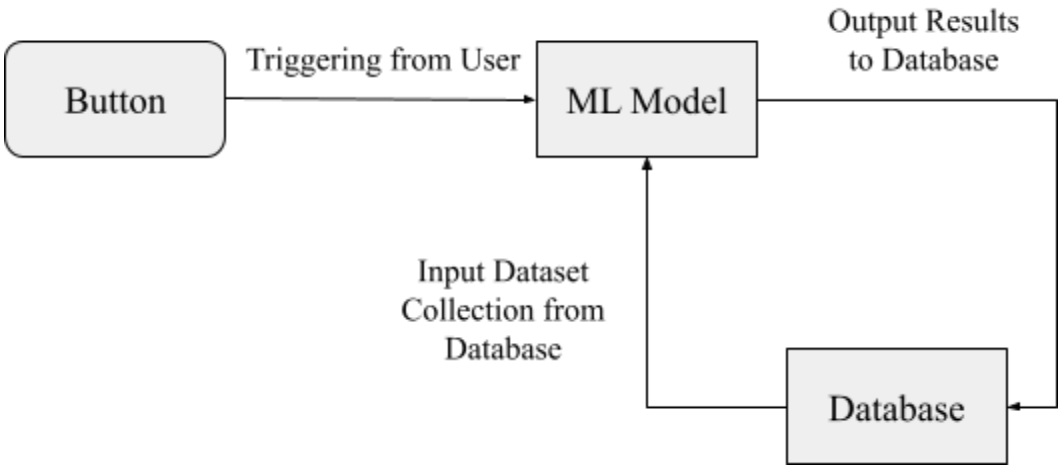

Fig. 8.1(a): ML Model Workflow

8.2 ANDROID WORKFLOW

It will start from the user's accessing of the android application through the smartphone. Initially, the system checks the “Is User logged in?” statement. If the user was already logged in, then the workflow will point toward the Home page. Otherwise, the system checks another conditional statement that “ Is User registered?”. If the user was already registered, then the workflow will point to the Login page and after the successful login, the user will reach the Home page. If the user was not registered, then the user will reach the registration page and after the successful registration with the username, email id and password the user accesses the weak activity page. Week activity can store the input through the calculation of date or direct week input. After this page, the user will enter the Home page.

On the Home page, the user has an option to predict the model by accessing the “Predict” button. But for the results, the user initially needs to enter the input features and details of that respective user through the Profile page. Profile page, Activity page and More Actions page were displayed in the bottom navigation bar of the android application and this page contains a specific set of functionalities. The Profile page is used for inputting the user’s essential information for predicting the low birth weight of the foetus. Already data is given users can update their data with the button “Update” on the Profile page. The Activity page is basically for setting the notification for the user. The notification was set for the remainder of the doctor’s appointment prior to one day. The input information of the reminder date can be entered through the activity page. After that, the final page More Actions are used for the user to log out from the current profile. Then the workflow will point towards the Login page. This is our system’s Android application workflow and it is presented in fig. 8.2(a).

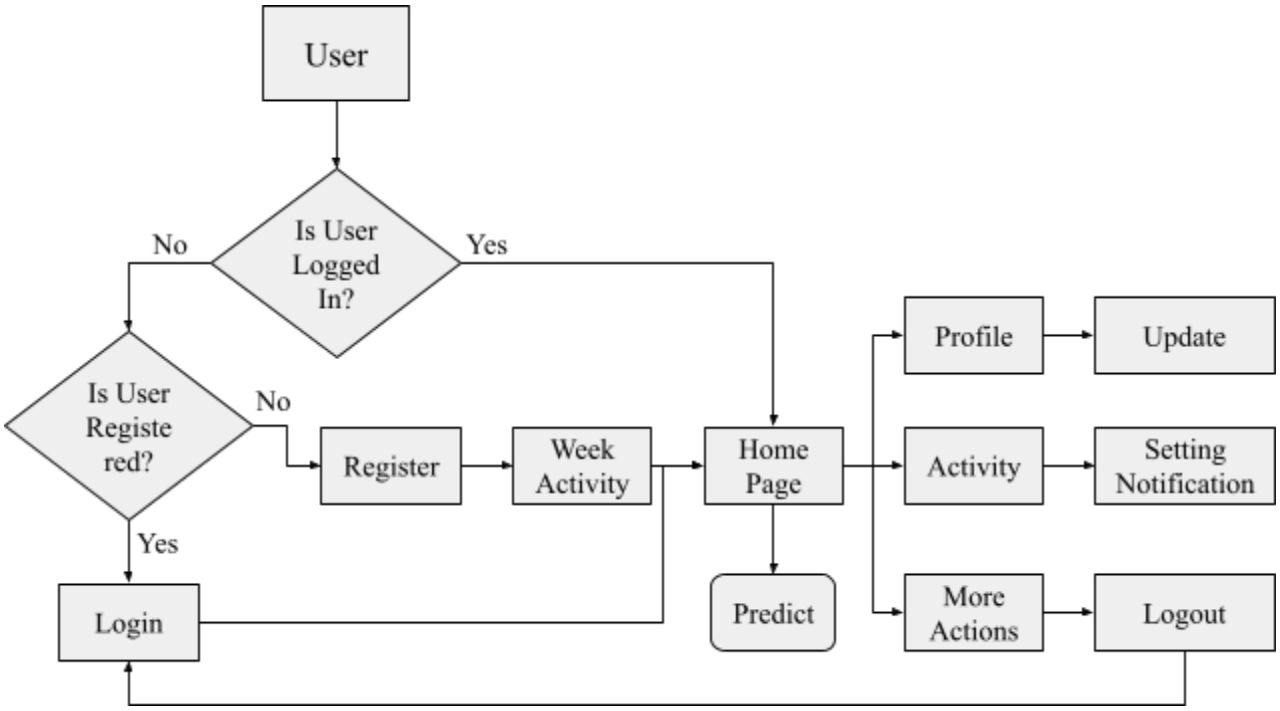

fig. 8.2(a): Android Workflow Diagram

9. SOFTWARE DEVELOPMENT

9.1 SOFTWARE ANALYSIS

9.1.1 FEASIBILITY STUDY

When a new project is proposed, it normally goes through a feasibility assessment. The feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Three areas of the feasibility study,

(i) Technological feasibility: Technological feasibility is related to the software and equipment specified in the design for implementing a new system. It is the study of function, performance and constraints that may affect the ability to achieve an acceptable system. Technological feasibility includes whether the technology is available on the market for development and its availability. During technical analysis, the analyst evaluates the technical merits of the system, at the same time collecting additional information about performance, reliability, maintainability and productivity. The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs, and procedures. The system “Pregnancy Assistant” is developed using Android, and Python, which is easily accessible. After conducting the technical analysis, we found that this project fulfils all the technical prerequisites.

(ii) Economic feasibility: This feasibility study presents tangible and intangible benefits from the project by comparing the development cost and operational cost. This evaluation looks at the financial aspects of the project. The cost to conduct a full system investigation, the cost of hardware and software, and the benefits in the form of reduced expenditure are all discussed during the economic feasibility. The technique of cost-benefit analysis is often used as a basis for assessing economic feasibility. It determines whether the investments needed to implement the system will be recovered. The system “Pregnancy Assistant” is economically feasible, as the

hardware and software resources are required for the functioning of the system and the actual no money value is needed for the purchase or implementation of the project.

(iii) Operational Feasibility: Operational feasibility is a measure of how people think about the system. This analysis involves how it will work when it is installed and the assessment of the political and managerial environment in which it is implemented. People are inherently resistant to change, and computers have been known to facilitate change. The system operation is the longest phase in the development cycle of a system. So, operational feasibility should be given much importance. The users of the system don't need thorough training on the system. All they are expected to know to operate the system is basic net surfing knowledge. It has a user-friendly interface. Thus, the organisation is convinced that the system is feasible. The system "Pregnancy Assistant" is operationally feasible.

9.1.2 REQUIREMENTS

Software Requirements:

- ➔ Technology used: Machine Learning, Android
- ➔ IDE: PyCharm, Android Studio
- ➔ Framework: FLASK
- ➔ Database: MySQL
- ➔ API creation: Postman

9.2 SOFTWARE DESIGN

9.2.1 PLATFORMS AND TOOLS

Android:

Android is a software stack for mobile devices that includes an operating system based on the Linux Kernel, middleware and key applications. On a larger scale, Android is an Ecosystem built by Google and its affiliates, which enables Android developers to build, deploy and monetize

applications for a range of touchscreen Android devices. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars etc.

Flask:

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. A framework 29 is a code library that makes a developer's life easier when building reliable, scalable, and maintainable web applications by providing reusable code or extensions for common operations. Each framework has a different way to put together its routes, models, views, database interaction, and overall application configuration. The Flask package is installable from the Python Package Index (PPI).

MySQL:

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases.

Android Studio:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built based on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

PyCharm:

PyCharm is a dedicated python IDE providing a wide range of essential tools for python developers, tightly integrated to create a convenient environment for productive Python, web and data science development.

Postman:

Postman is an API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. The result - is more efficient and less tedious work.

9.3 UI DESIGN

Appearance of Care'S on a mobile phone:

The initial appearance of the android application with its logo is shown in Fig 9.3(a)

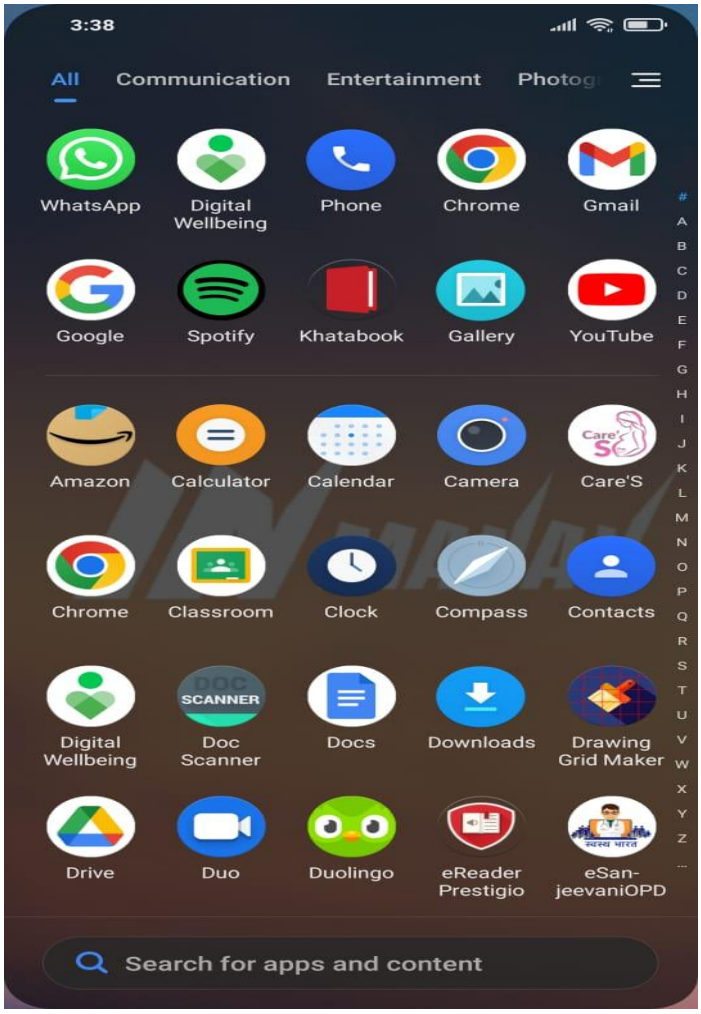

Fig 9.3(a) Outlook of Care'S

Login Page

If the user is registered, then one can directly login else register and log in (Fig 9.3(b) and 9.3(c))

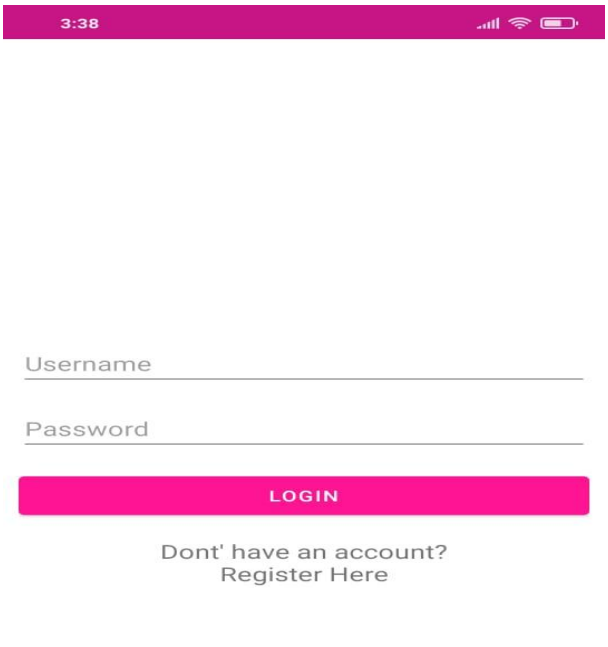

Fig 9.3 (b) Login Page

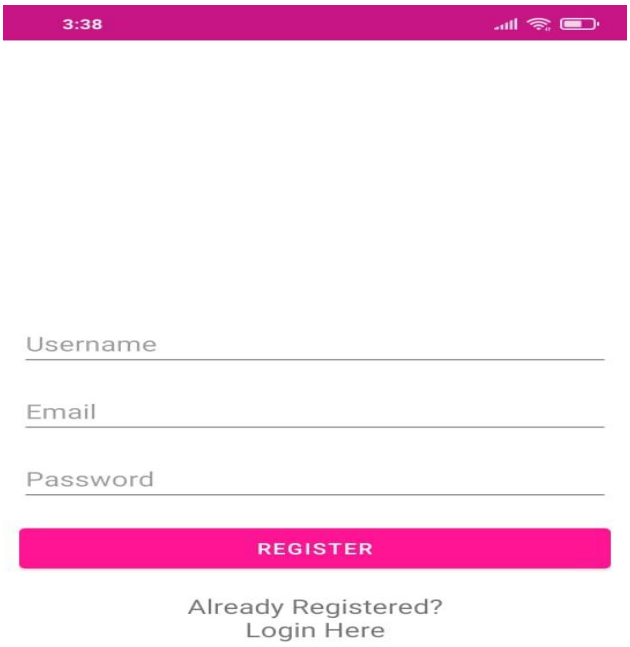

Fig 9.3(c) Register Page

Week Input

After the login, the user is asked to provide either the week of her pregnancy or the first day of her last menstruation in order to provide the food and exercise recommendation.

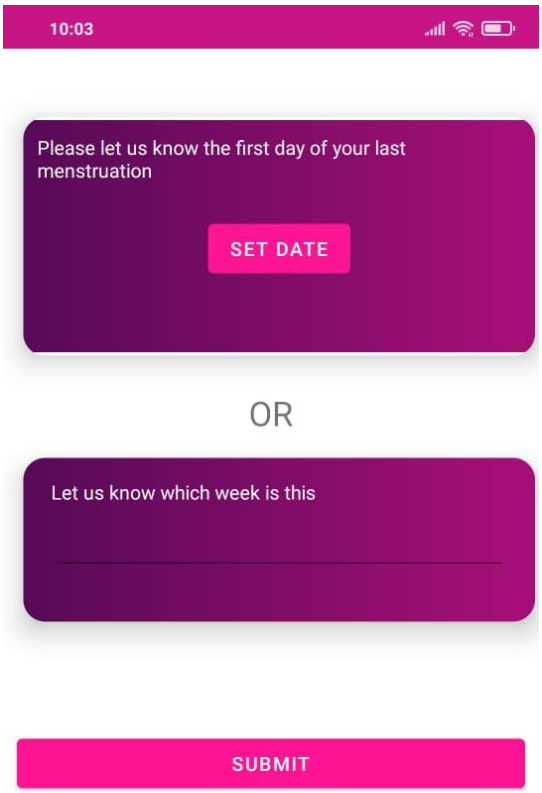

Fig 9.3(d) Week Input

Home Page

The Home Page consists of the LBW prediction along with the recommended food items (Fig 9.3(e)).

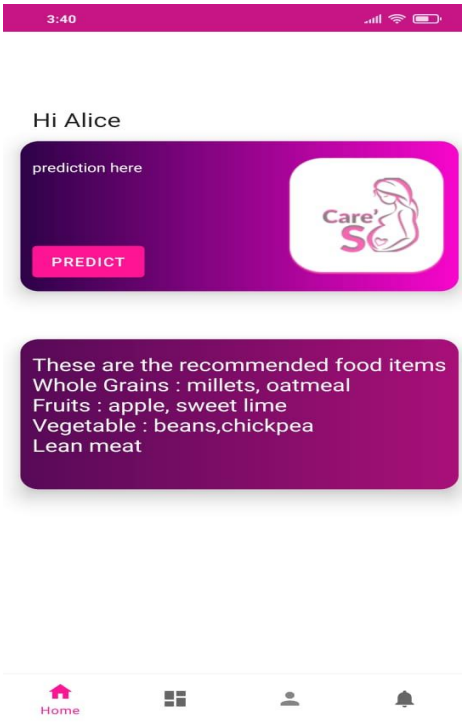

Fig 9.3(e) Home Page

Activity Page

The Activity Page consists of the exercise to be followed along with the appointment date in order to provide the notification.

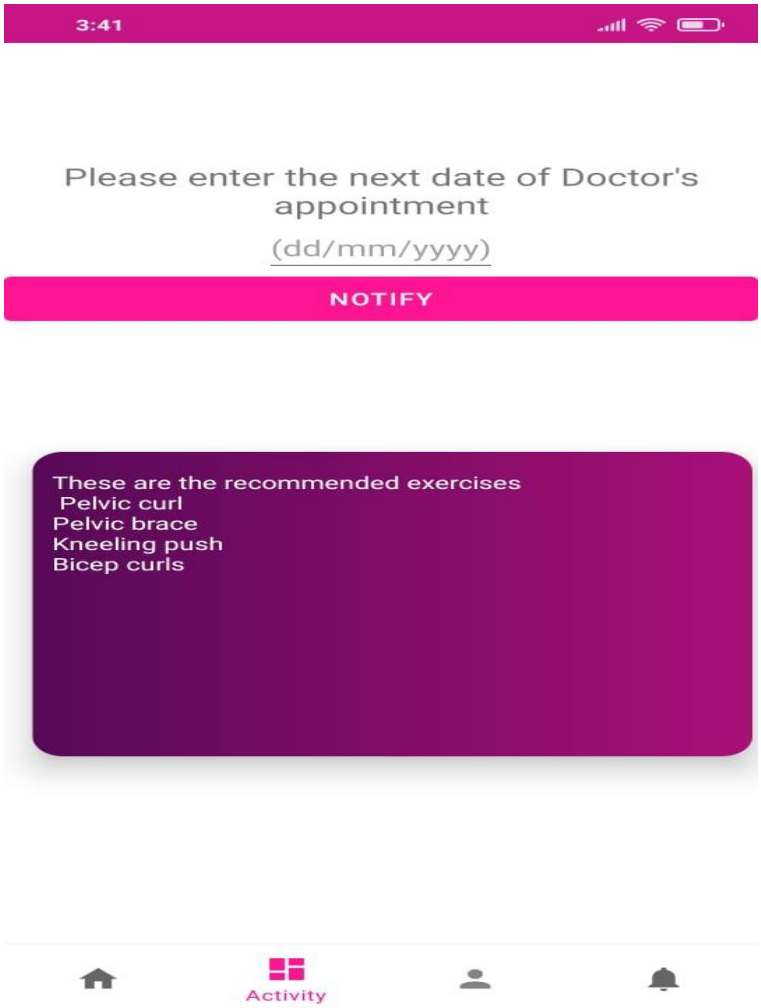

Fig 9.3(f) Activity Page

Profile and More Actions Page

The Profile page helps the user to update their mail id whereas the more actions page consists of the logout icon.(Fig 9.3(g))

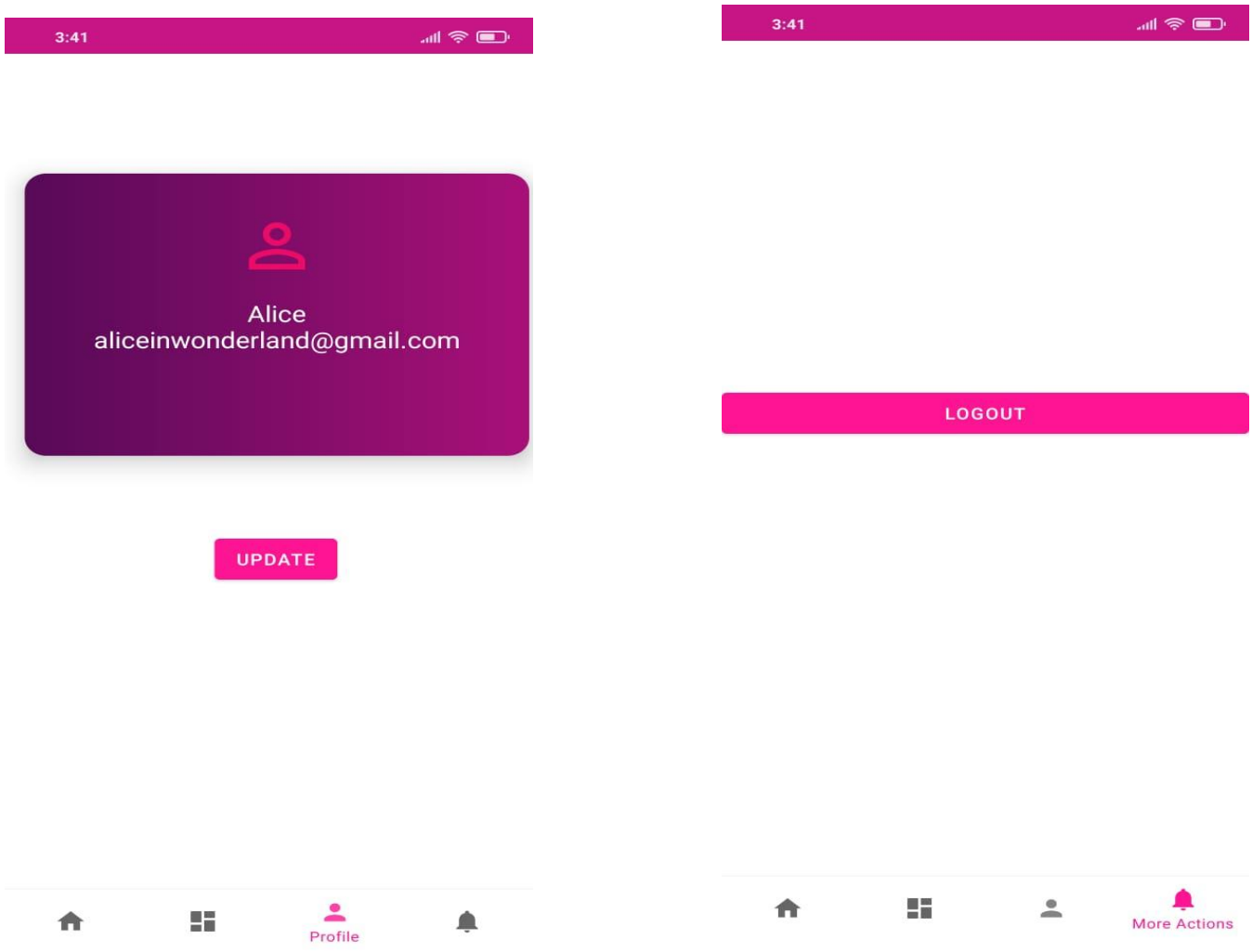

Fig 9.3(g) User and More Actions Page

LBW Model Inputs

The input features to predict the LBW of the foetus are provided by the user as in Fig 9.3(h).

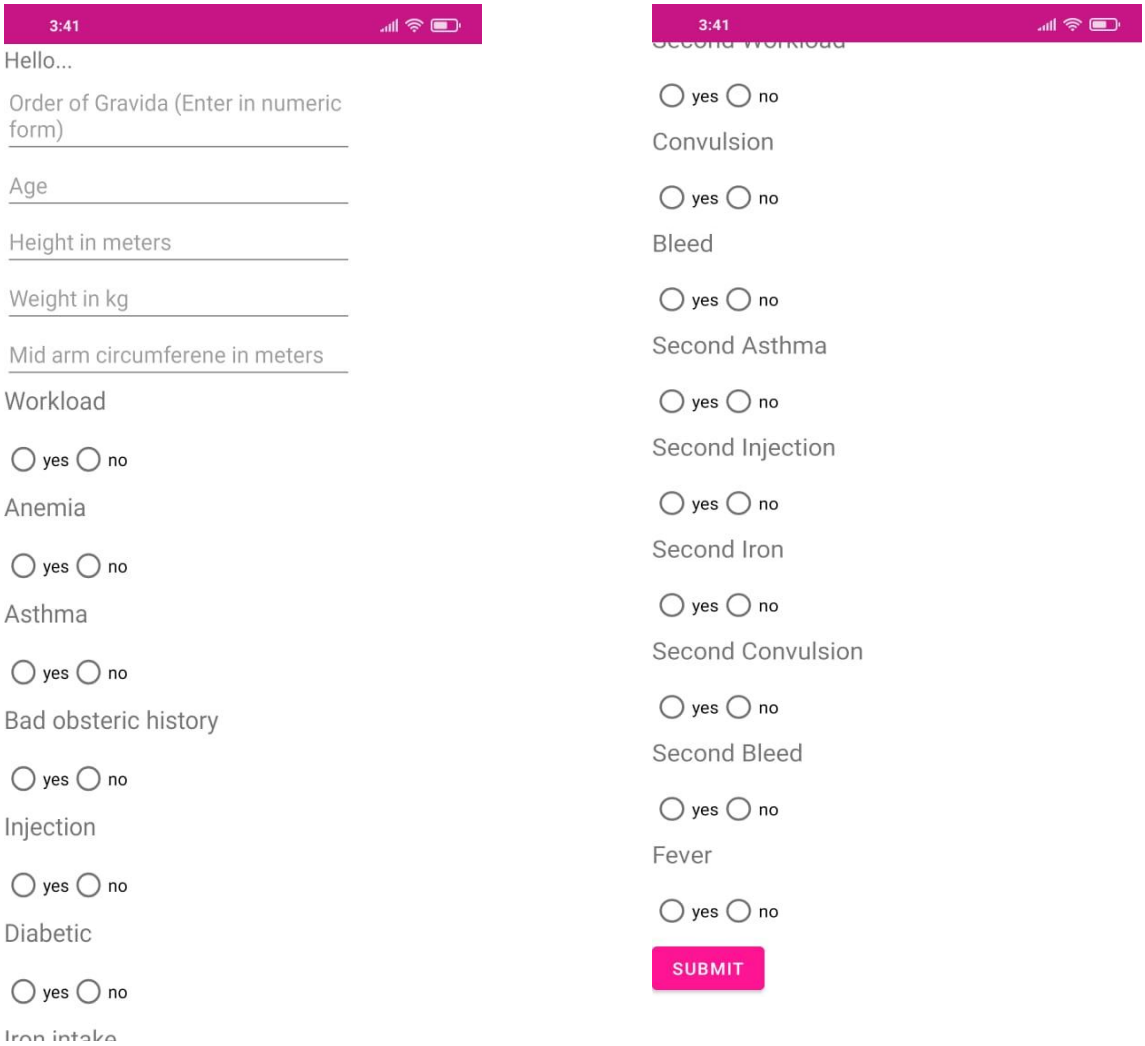

Fig 9.3(h) Inputs of LBW Model

Output Page
The output of the LBW Model will be displayed on the home page.

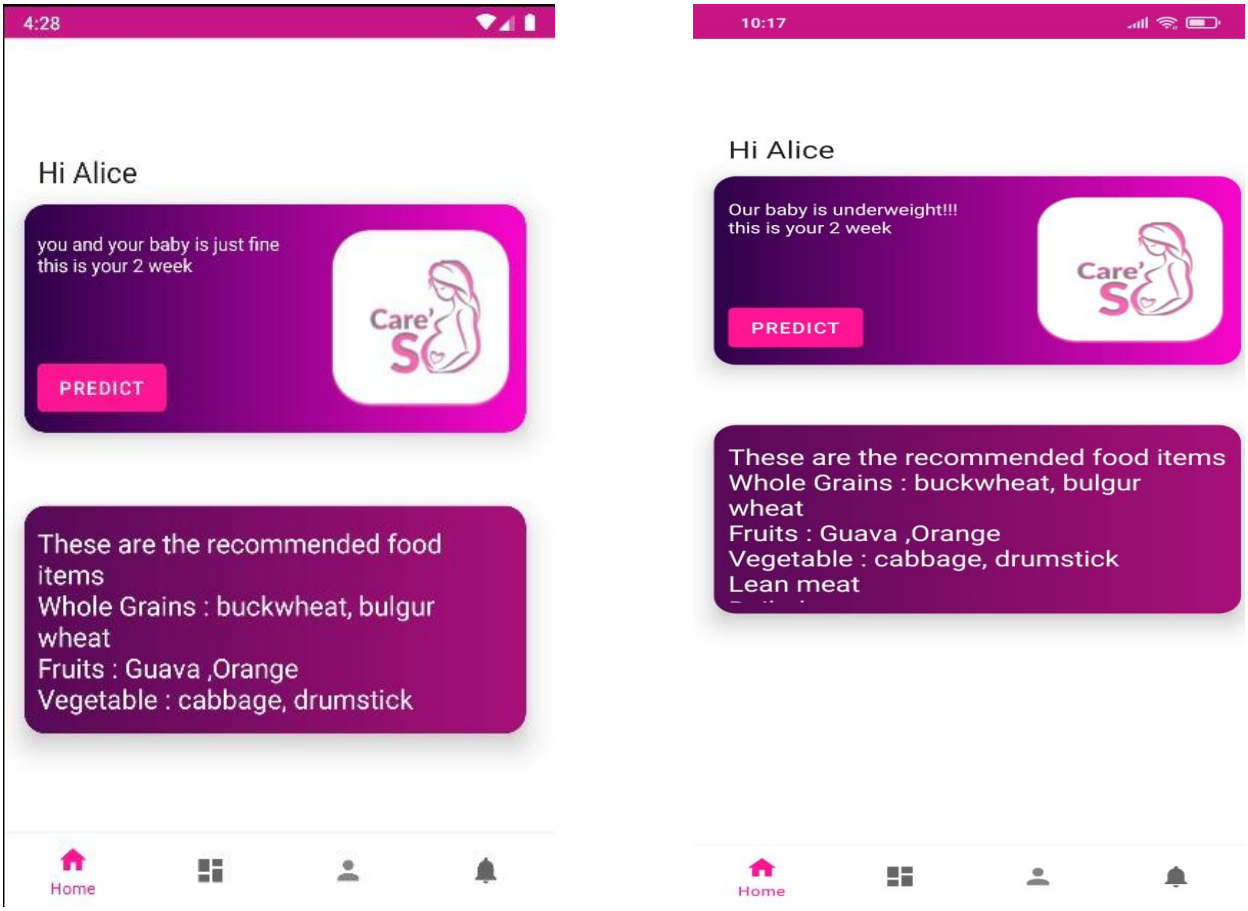

Fig 9.3(i) Output of LBW Model

10. RESULTS AND DISCUSSIONS

The result of this project is that it predicts the low birth weight of a foetus by analyzing and correlating 22 features which prove to have a significant impact on the mother's and foetus's health. Also, the system gives the food recommendation, exercise information and reminder functionality to the user. After a review of the literature, some demographic, obstetric, and clinical characteristics of the mother were collected at baseline and considered to predict low birth weight at term. The univariable analysis found several factors were eligible to be included in the prediction model. The android application interacts with the user; the user can input the values and obtain the output. The values and the predicted probability are saved in the database which can be viewed by the user. Food and exercise recommendations keep mothers' health also in very good condition. The machine learning model predicts the probability with an accuracy of 86% which proves to be very reliable.

Predicting the probability of LBW in pregnant women is essential to take appropriate measures accordingly. Giving better care and better food and exercise recommendations will provide better health conditions for mothers. Hence it will keep the foetus from becoming low birth weight and give better health conditions for the newborn child.

11. CONCLUSION AND SCOPE FOR FUTURE WORK

11.1 CONCLUSION

In this project, a pregnancy assistant has been implemented as an android application named “Care’S” which contains a machine learning model. The proposed system assists pregnant women in terms of her food intake, exercise to be followed and also provides reminders for her check ups.The system also contains a Low Birth Weight model implemented through random forest which predicts whether the fetus has a low birth weight or not.

11.2 SCOPE FOR FUTURE WORK

For future work, this project can be extended to predict the very low birth weight and extremely low birth weight of the unborn baby. Early disease prediction of the pregnant woman can be done by monitoring her health conditions. Furthermore, the project can be extended to Provide guidelines to care the newborn baby upto the age of two.

12. APPENDIX

1. LBW Machine Learning Model

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import pickle
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
lbw_data=pd.read_csv(r'C:\Users\Jpamb\Desktop\Low-Birth-Weight-Prediction-main\Total_data.csv')
lbw_data = lbw_data.drop(['Unnamed: 0','id','lda','relg','mwt','mwt2','mwt3'], axis = 'columns', inplace = False)
lbw_data.rename(columns = {'childwt': 'child_weight', 'logit': 'target', 'mage':'mother_age', 'mht': 'mother_height', 'mdarm': 'mid_arm_cir', 'head':'head_cir','occup':'occupation', 'educ':'education', 'f11':'proteinTime_12th_week','wm18':'adult_wom_18','ageyc':'age_at_pregnancy','boh':'bad_obs_history','ecostat':'econoic_status','belly1':'abdominal_girth_20wk','fand1':'Fundal_height_20wk','sis1':'1st_systolic','dis1':'1st_diastolic','inject':'injection','worklo':'2nd_workload','conv1':'1st_convolusion','belly2':'abdominal_girth_28wk','fand2':'Fundal_height_28wk','sis2':'2nd_systolic','dia2':'2nd_diastolic','workl':'3rd_workload','f13':'proteinTime_28th_week','conv2':'2nd_convolusion'}, inplace = True)
from sklearn.impute import SimpleImputer
impt = SimpleImputer(missing_values = np.nan, strategy = 'most_frequent')
lbw_data = impt.fit_transform(lbw_data)
lbw_data = pd.DataFrame(lbw_data, columns=['parity', 'child_weight', 'target', 'mother_age', 'mother_height','mid_arm_cir','head_cir','habit','occupation','education','workload','proteinTime_12th_week','f21','f31','f41','anemia','asthma','adult_wom_18','age_at_pregnancy','bad_obs_history','econoic_status','abdominal_girth_20wk','Fundal_height_20wk','1st_systolic', '1st_diastolic', 'injection',
```

```
'falif', 'iron','2nd_workload', 'f12', 'f22', 'f32', 'f42', 'rest', '1st_convolusion','d21', 'bleed1', 'blddef1', 'asthma2', 'fever1', 'lomot1', 'contab1','db1', 'abdominal_girth_28wk', 'Fundal_height_28wk', '2nd_systolic','2nd_diastolic', 'inject2', 'falif2', 'iron2', '3rd_workload','proteinTime_28th_week', 'f23', 'f33', 'f43', 'rest1', '2nd_convolusion', 'd', 'bleed', 'blddef', 'fever', 'lomot','contab', 'db', 'wg1', 'wg2', 'BMI'])
from sklearn.preprocessing import StandardScaler
StandardScaler = StandardScaler()
columns_to_scale=['mother_age','mid_arm_cir','head_cir',‘abdominal_girth_20wk','Fundal_height_20wk','1st_systolic','1st_diastolic','abdominal_girth_28wk','Fundal_height_28wk','2nd_systolic','2nd_diastolic',‘BMI']
lbw_data[columns_to_scale] = standardScaler.fit_transform(lbw_data[columns_to_scale])
lbw_data_sub=lbw_data[['parity','target','mother_age','mother_height','mid_arm_cir','workload','anemia','asthma','bad_obs_history','injection','falif','iron','2nd_workload','1st_convolusion','bleed1','asthma2','inject2', 'iron2', '2nd_convolusion', 'bleed','fever','BMI']]
X = lbw_data_sub.drop(['target'],axis=1)
y = lbw_data_sub['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.25,shuffle=True, random_state=5)
y = lbw_data_sub['target']
X = lbw_data_sub.drop('target', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)
rf_ent = RandomForestClassifier(criterion='entropy',n_estimators=100)
rf_ent.fit(X_train, y_train)
y_pred_rfe = rf_ent.predict(X_test)
print(X.columns)
print(type(X_test))
nparray = [np.array(data)]
```

```
pickle.dump(rf_ent, open('model.pkl','wb'))
model = pickle.load(open('model.pkl','rb'))
print(model.predict(nparray))
```

2. API for LBW

```
from flask import Flask, request
from flask_mysqldb import MySQL
import numpy as np
import pickle
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'cares'
mysql = MySQL(app)
@app.route('/predict',methods=['GET', 'POST'])
def predict():
    id = int(request.args.get('id'))
    cursor = mysql.connect.cursor()
    cursor.execute("""SELECT parity, mother_age, mother_height, mid_arm_cir, workload, anemia, asthma, bad_obs_history, injection, falif, iron, 2nd_workload, 1st_convolusion, bleed1, asthma2, inject2, iron2, 2nd_convolusion, bleed, fever, BMI FROM USERS WHERE id = %s""", (id,))
    int_features = cursor.fetchone()
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)
    str_pred = str(prediction[0])
    cursor.execute(''' UPDATE USERS SET TARGET=%s WHERE id = %s''', (str_pred,id,))
```

```
    mysql.connection.commit()
    cursor.close()
    return str_pred
if __name__ == "__main__":
    app.run(debug=True)
```

3. Backend Database ConnectionAPI of Android

```
<?php
    require_once 'DbConnect.php';
    $response = array();
    if(isset($_GET['apicall'])){
      switch($_GET['apicall']){
        case 'signup':
          if(isTheseParametersAvailable(array('username','email','password'))){
            $username = $_POST['username'];
            $email = $_POST['email'];
            $password = md5($_POST['password']);
            $stmt = $conn->prepare("SELECT id FROM users WHERE username = ? OR email = ?");
            $stmt->bind_param("ss", $username, $email);
            $stmt->execute();
            $stmt->store_result();
            if($stmt->num_rows > 0){
             $response['error'] = true;
             $response['message'] = 'User already registered';
             $stmt->close();
            }
            else{
```

```
            $stmt = $conn->prepare("INSERT INTO users (username, email, password) VALUES (?,
?, ?)");
            $stmt->bind_param("sss", $username, $email, $password);
            if($stmt->execute()){
                $stmt = $conn->prepare("SELECT id, username, email,TARGET,week FROM users
WHERE username = ?");
                $stmt->bind_param("s",$username);
                $stmt->execute();
                $stmt->bind_result($id, $username, $email,$target,$week);
                $stmt->fetch();
                $user = array(
                'id'=>$id,
                'username'=>$username,
                'email'=>$email,
                'target'=>$target,
                        'week'=>$week
                );
                $stmt->close();
                $response['error'] = false;
                $response['message'] = 'User registered successfully';
                $response['user'] = $user;
              }
             }

        }
        else{
          $response['error'] = true;
          $response['message'] = 'required parameters are not available';
```

```
        }
      break;
      case 'login':
       if(isTheseParametersAvailable(array('username', 'password'))){
        $username = $_POST['username'];
        $password = md5($_POST['password']);
              $stmt = $conn->prepare("SELECT id, username, email,TARGET,week FROM users
WHERE username = ? AND password = ?");
        $stmt->bind_param("ss",$username, $password);
        $stmt->execute();
        $stmt->store_result();
        if($stmt->num_rows > 0){
         $stmt->bind_result($id, $username, $email,$target,$week);
         $stmt->fetch();
         $user = array(
         'id'=>$id,
         'username'=>$username,
         'email'=>$email,
         'target'=>$target,
                  'week'=>$week
         );

         $response['error'] = false;
         $response['message'] = 'Login successfull';
         $response['user'] = $user;
        }
        else{
         $response['error'] = true;
```

```
            $response['message'] = 'Invalid username or password';
            }
        }
    case 'update':
                    if(isTheseParametersAvailable(array('id','parity', 'mother_age', 'mother_height', 'mid_arm_cir','workload','anemia', 'asthma', 'bad_obs_history', 'injection', 'falif','iron', 'workload2', 'convolusion','bleed','asthma2','inject2', 'iron2', 'convolusion2', 'bleed2','fever','BMI'))){
            $id = $_POST['id'];
            $parity = $_POST['parity'];
            $mother_age = $_POST['mother_age'];
            $mother_height = $_POST['mother_height'];
            $mid_arm_cir = $_POST['mid_arm_cir'];
            $workload = $_POST['workload'];
            $anemia = $_POST['anemia'];
            $asthma = $_POST['asthma'];
            $bad_obs_history = $_POST['bad_obs_history'];
            $injection = $_POST['injection'];
            $falif = $_POST['falif'];
            $iron = $_POST['iron'];
            $workload2 = $_POST['workload2'];
            $convolusion = $_POST['convolusion'];
            $bleed = $_POST['bleed'];
            $asthma2 = $_POST['asthma2'];
            $injection2 = $_POST['inject2'];
            $iron2 = $_POST['iron2'];
            $convolusion2 = $_POST['convolusion2'];
            $bleed2 = $_POST['bleed2'];
            $fever = $_POST['fever'];
```

```
$bmi = $_POST['BMI'];
$stmt = $conn->prepare("UPDATE users SET parity=?, mother_age=?, mother_height=?, mid_arm_cir=?,workload=?,anemia=?, asthma=?, bad_obs_history=?, injection=?, falif=?,iron=?, 2nd_workload=?, 1st_convolusion=?,bleed1=?,asthma2=?,inject2=?, iron2=?, 2nd_convolusion=?, bleed=?,fever=?,BMI=? WHERE id=?");
$stmt->bind_param("sssssssssssssssssssss",$parity,$mother_age,$mother_height,$mid_arm_cir,$workload,$anemia,$asthma,$bad_obs_history,$injection,$falif,$iron,$workload2,$convolusion,$bleed,$asthma2,$injection2,$iron2,$convolusion2,$bleed2,$fever,$bmi,$id);
if($stmt->execute()){
 $stmt->store_result();
  $response['error'] = false;
  $response['message'] = 'Updated successfully';
}
else{
  $response['error'] = true;
  $response['message'] = 'Unexpected error occurred';
}
}

break;
  case 'refresh':
 if(isTheseParametersAvailable(array('id'))){
  $id = $_POST['id'];
      $stmt = $conn->prepare("SELECT id, username, email,TARGET,week FROM users WHERE id = ?");
  $stmt->bind_param("s",$id);
  $stmt->execute();
  $stmt->bind_result($userid,$username,$email,$target,$week);
```

```
            $stmt->fetch();
            $user = array(
            'id'=>$userid,
            'username'=>$username,
            'email'=>$email,
            'target'=>$target,
                        'week'=>$week
            );
            $stmt->close();
            $response['error'] = false;
            $response['message'] = 'User refreshed successfully';
            $response['user'] = $user;
          }
          else{
            $response['error'] = true;
            $response['message'] = 'Unexpected error occurred';
          }
        break;
            case 'week':
          if(isTheseParametersAvailable(array('id','week'))){
            $id = $_POST['id'];
            $week = $_POST['week'];
            $stmt = $conn->prepare("UPDATE users SET week=? WHERE id=?");
            $stmt->bind_param("ss",$week,$id);
            if($stmt->execute()){
              $stmt->store_result();
                    $stmt = $conn->prepare("SELECT id, username, email,TARGET,week FROM users
WHERE id = ?");
```

```
    $stmt->bind_param("s",$id);
    $stmt->execute();
    $stmt->bind_result($userid,$username,$email,$target,$week);
    $stmt->fetch();
    $user = array(
    'id'=>$userid,
    'username'=>$username,
    'email'=>$email,
    'target'=>$target,
    'week'=>$week
    );
    $stmt->close();
    $response['error'] = false;
    $response['message'] = 'Week updated successfully';
    $response['user'] = $user;
   }
   else{
    $response['error'] = true;
    $response['message'] = 'Unexpected error occurred';
   }
  }
  else{
   $response['error'] = true;
   $response['message'] = 'Unexpected error occurred';
  }
 break;
   case 'food':
  if(isTheseParametersAvailable(array('week'))){
```

```
    $week = $_POST['week'];
    $stmt = $conn->prepare("SELECT Items FROM food WHERE Week = ?");
    $stmt->bind_param("s",$week);
    $stmt->execute();
    $stmt->bind_result($items);
    $stmt->fetch();
    $stmt->close();
    $response['error'] = false;
    $response['message'] = 'Items fetched successfully';
    $response['food'] = $items;
   }
   else{
    $response['error'] = true;
    $response['message'] = 'Unexpected error occurred';
   }
 break;
    case 'exc':
   if(isTheseParametersAvailable(array('week'))){
    $week = $_POST['week'];
    $stmt = $conn->prepare("SELECT items FROM exercises WHERE Week = ?");
    $stmt->bind_param("s",$week);
    $stmt->execute();
    $stmt->bind_result($items);
    $stmt->fetch();
    $stmt->close();
    $response['error'] = false;
    $response['message'] = 'Items fetched successfully';
    $response['exc'] = $items;
```

```
        }
        else{
          $response['error'] = true;
          $response['message'] = 'Unexpected error occurred';
        }
       break;
       default:
         $response['error'] = true;
         $response['message'] = 'Invalid Operation Called';
     }
   }
   else{
    $response['error'] = true;
    $response['message'] = 'Invalid API Call';
   }
   echo json_encode($response);
   function isTheseParametersAvailable($params){
     foreach($params as $param){
       if(!isset($_POST[$param])){
         return false;
       }
     }
   return true;
   }
?>
```

4. DATABASE CONNECTION

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "CareS";
 $conn = new mysqli($servername, $username, $password, $database);
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
?>
```

5. ANDROID

a. Android Manifests.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.cea.cares">
        <uses-permission android:name="android.permission.INTERNET" />
        <uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM" />
        <application
        android:allowBackup="true"
        android:icon="@mipmap/cares_logo"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/cares_logo_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.CareS"
        android:usesCleartextTraffic="true">
```

```
        <activity android:name=".MainActivity" />
        <activity android:name=".LoginActivity" />
        <activity android:name=".WeekActivity"/>
        <activity android:name=".ui.profile.UpdateProfile" />
        <activity
        android:name=".RegisterActivity"
        android:exported="true"
        android:label="@string/app_name">
        <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        </activity>
        <receiver android:name=".Background"
        android:enabled="true"
        android:exported="true">
        </receiver>
        </application>
</manifest>
```

App

b. Background.java

```
package com.cea.cares;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.BroadcastReceiver;
import android.content.Context;
```

```
import android.content.Intent;
import android.os.Build;
import android.widget.Toast;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import com.cea.cares.ui.activity.ActivityFragment;
public class Background extends BroadcastReceiver {
   @Override
   public void onReceive(Context arg0, Intent arg1)
      ActivityFragment activityFragment = new ActivityFragment();
      activityFragment.addNotification(arg0);
   }
}
```

c. LoginActivity.java

```
package com.cea.cares;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
```

```
public class LoginActivity extends AppCompatActivity {
   EditText editTextUsername, editTextPassword;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.login);
      editTextUsername = (EditText) findViewById(R.id.editTextUsername);
      editTextPassword = (EditText) findViewById(R.id.editTextPassword);
 findViewById(R.id.buttonLogin).setOnClickListener(new  View.OnClickListener()
{
         @Override
         public void onClick(View view) {
            userLogin();
         }
      });
findViewById(R.id.textViewRegister).setOnClickListener(newView.OnClickListen
er() {
         @Override
         public void onClick(View view) {
            finish();
           startActivity(new Intent(getApplicationContext(), RegisterActivity.class));
         }
      });
   }
   private void userLogin() {
      final String username = editTextUsername.getText().toString();
      final String password = editTextPassword.getText().toString();
      if (TextUtils.isEmpty(username)) {
```

```
    editTextUsername.setError("Please enter your username");
    editTextUsername.requestFocus();
    return;
}
if (TextUtils.isEmpty(password)) {
    editTextPassword.setError("Please enter your password");
    editTextPassword.requestFocus();
    return;
}
class UserLogin extends AsyncTask<Void, Void, String> {
    ProgressBar progressBar;
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        progressBar.setVisibility(View.VISIBLE);
    }
    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        progressBar.setVisibility(View.GONE);
        try {
            JSONObject obj = new JSONObject(s);
            if (!obj.getBoolean("error")) {
                                JSONObject userJson = obj.getJSONObject("user");
                User user = new User(
                        userJson.getInt("id"),
                        userJson.getInt("target"),
```

```
                    userJson.getString("username"),
                    userJson.getString("email"),
                    userJson.getInt("week")
                );
        SharedPrefManager.getInstance(getApplicationContext()).userLogin(user);
                finish();
            startActivity(new Intent(getApplicationContext(), MainActivity.class));
            } else {
                    }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    @Override
    protected String doInBackground(Void... voids) {
        RequestHandler requestHandler = new RequestHandler();
        HashMap<String, String> params = new HashMap<>();
        params.put("username", username);
        params.put("password", password);
        return requestHandler.sendPostRequest(URLs.URL_LOGIN, params);
    }}

    UserLogin ul = new UserLogin();
    ul.execute();
}
@Override
public void onBackPressed() {
    finishAffinity();
```

```
}}
```

d. MainActivity.java

```
package com.cea.cares;
import android.os.Bundle;
import com.google.android.material.bottomnavigation.BottomNavigationView;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import com.cea.cares.databinding.ActivityMainBinding;
public class MainActivity extends AppCompatActivity {
   private ActivityMainBinding binding;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      binding = ActivityMainBinding.inflate(getLayoutInflater());
      setContentView(binding.getRoot());
      BottomNavigationView navView = findViewById(R.id.nav_view);
  AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
            R.id.navigation_home, R.id.navigation_activity, R.id.navigation_profile,
R.id.navigation_moreactions)
          .build();
            NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_activity_main);
```

```
            // NavigationUI.setupActionBarWithNavController(this, navController,
appBarConfiguration);
        NavigationUI.setupWithNavController(binding.navView, navController);
    }}
```

e. RegisterActivity.java

```
package com.cea.cares;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
public class RegisterActivity extends AppCompatActivity {
    EditText editTextUsername, editTextEmail, editTextPassword;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.register);
        //if the user is already logged in we will directly start the profile activity
        if (SharedPrefManager.getInstance(this).isLoggedIn()) {
            finish();
```

```
            startActivity(new Intent(this, MainActivity.class));
            return;
        }
        editTextUsername = findViewById(R.id.editTextUsername);
        editTextEmail = findViewById(R.id.editTextEmail);
        editTextPassword = findViewById(R.id.editTextPassword);
                                findViewById(R.id.buttonRegister).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                registerUser();
            }
        });


                                findViewById(R.id.textViewLogin).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
                startActivity(new Intent(RegisterActivity.this, LoginActivity.class));
            }
        });
    }
    private void registerUser() {
        final String username = editTextUsername.getText().toString().trim();
        final String email = editTextEmail.getText().toString().trim();
        final String password = editTextPassword.getText().toString().trim();
            if (TextUtils.isEmpty(username)) {
```

```
        editTextUsername.setError("Please enter username");
        editTextUsername.requestFocus();
        return;
    }
    if (TextUtils.isEmpty(email)) {
        editTextEmail.setError("Please enter your email");
        editTextEmail.requestFocus();
        return;
    }
    if (!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        editTextEmail.setError("Enter a valid email");
        editTextEmail.requestFocus();
        return;
    }
    if (TextUtils.isEmpty(password)) {
        editTextPassword.setError("Enter a password");
        editTextPassword.requestFocus();
        return;
    }
    class RegisterUser extends AsyncTask<Void, Void, String> {
        private ProgressBar progressBar;
        @Override
        protected String doInBackground(Void... voids) {
            RequestHandler requestHandler = new RequestHandler();
            HashMap<String, String> params = new HashMap<>();
            params.put("username", username);
            params.put("email", email);
            params.put("password", password);
```

```
                return  requestHandler.sendPostRequest(URLs.URL_REGISTER,
params);
        }
        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            progressBar = (ProgressBar) findViewById(R.id.progressBar);
            progressBar.setVisibility(View.VISIBLE);
        }
        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            progressBar.setVisibility(View.GONE);
            try {
                JSONObject obj = new JSONObject(s);
                if (!obj.getBoolean("error")) {
                    Toast.makeText(getApplicationContext(), obj.getString("message"),
Toast.LENGTH_SHORT).show();

                    JSONObject userJson = obj.getJSONObject("user");
                    User user = new User(
                            userJson.getInt("id"),
                            userJson.getInt("target"),
                            userJson.getString("username"),
                            userJson.getString("email"),
                            userJson.getInt("week")
                    );
```

```
                    Toast.makeText(getApplicationContext(), "user created",
Toast.LENGTH_SHORT).show();
SharedPrefManager.getInstance(getApplicationContext()).userLogin(user);
                    Toast.makeText(getApplicationContext(), "saved shared pref",
Toast.LENGTH_SHORT).show();
                finish();
                                    startActivity(new Intent(getApplicationContext(),
WeekActivity.class));
            } else {
                    Toast.makeText(getApplicationContext(), "Some error occurred",
Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }}}
  RegisterUser ru = new RegisterUser();
    ru.execute();
  }
  @Override
  public void onBackPressed() {
    finishAffinity();
  }}
```

f. RequestHandler.java

```
package com.cea.cares;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
```

```
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;
import javax.net.ssl.HttpsURLConnection;
public class RequestHandler {
    public String sendPostRequest(String requestURL, HashMap<String, String>
postDataParams) {
        URL url;
        StringBuilder sb = new StringBuilder();
        try {
            url = new URL(requestURL);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(15000);
            conn.setConnectTimeout(15000);
            conn.setRequestMethod("POST");
            conn.setDoInput(true);
            conn.setDoOutput(true);
            OutputStream os = conn.getOutputStream();
            BufferedWriter writer = new BufferedWriter(
                    new OutputStreamWriter(os, "UTF-8"));
            writer.write(getPostDataString(postDataParams));
            writer.flush();
            writer.close();
```

```
            os.close();
            int responseCode = conn.getResponseCode();
            if (responseCode == HttpsURLConnection.HTTP_OK) {
                                    BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                sb = new StringBuilder();
                String response;
                while ((response = br.readLine()) != null) {
                   sb.append(response);
                }}
        } catch (Exception e) {
            e.printStackTrace();
        }
        return sb.toString();
    }
        private String getPostDataString(HashMap<String, String> params) throws
UnsupportedEncodingException {
        StringBuilder result = new StringBuilder();
        boolean first = true;
        for (Map.Entry<String, String> entry : params.entrySet()) {
            if (first)
                first = false;
            else
                result.append("&");
            result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
        }
```

```
        return result.toString();
    }}
```

g. SharedPrefManager.java

```
package com.cea.cares;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
public class SharedPrefManager {
    private static final String SHARED_PREF_NAME = "sharedprefCares";
    private static final String KEY_USERNAME = "keyusername";
    private static final String KEY_EMAIL = "keyemail";
    private static final String KEY_ID = "keyid";
    private static final String KEY_TARGET = "keytarget";
    private static final String KEY_WEEK = "keyweek";
    private static SharedPrefManager mInstance;
    private static Context mCtx;
    private SharedPrefManager(Context context) {
        mCtx = context;
    }
    public static synchronized SharedPrefManager getInstance(Context context) {
        if (mInstance == null) {
            mInstance = new SharedPrefManager(context);
        }
        return mInstance;
    }
    public void userLogin(User user) {
```

```
        SharedPreferences
sharedPreferences=mCtx.getSharedPreferences(SHARED_PREF_NAME,
Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putInt(KEY_ID, user.getId());
        editor.putInt(KEY_TARGET, user.getTarget());
        editor.putString(KEY_USERNAME, user.getUsername());
        editor.putString(KEY_EMAIL, user.getEmail());
        editor.putInt(KEY_WEEK, user.getWeek());
        editor.apply();
    }
    public boolean isLoggedIn() {
                                    SharedPreferences sharedPreferences =
mCtx.getSharedPreferences(SHARED_PREF_NAME,
Context.MODE_PRIVATE);
        return sharedPreferences.getString(KEY_USERNAME, null) != null;
    }
    public User getUser() {
                                        SharedPreferences sharedPreferences =
mCtx.getSharedPreferences(SHARED_PREF_NAME,
Context.MODE_PRIVATE);
        return new User(
                sharedPreferences.getInt(KEY_ID, -1),
                sharedPreferences.getInt(KEY_TARGET,-1),
                sharedPreferences.getString(KEY_USERNAME, null),
                sharedPreferences.getString(KEY_EMAIL, null),
                sharedPreferences.getInt(KEY_WEEK,-1)
        );
```

```
    }
    public void logout() {
                                        SharedPreferences sharedPreferences =
mCtx.getSharedPreferences(SHARED_PREF_NAME,
Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.clear();
        editor.apply();
        mCtx.startActivity(new Intent(mCtx, LoginActivity.class));
    }}
```

h. URLs.java

```
package com.cea.cares;
public class URLs {
                    private static final String ROOT_URL =
"http://10.0.2.2/project/Api.php?apicall=";
//private static final String ROOT_URL =
"http://192.168.209.231/project/Api.php?apicall=";
    public static final String URL_REGISTER = ROOT_URL + "signup";
    public static final String URL_LOGIN= ROOT_URL + "login";
    public static final String URL_UPDATE= ROOT_URL + "update";
    public static final String URL_PREDICT= "http://10.0.2.2:5000/predict?id=";
    public static final String URL_REFRESH= ROOT_URL + "refresh";
    public static final String URL_WEEK= ROOT_URL + "week";
    public static final String URL_FOOD= ROOT_URL + "food";
    public static final String URL_EXC= ROOT_URL + "exc";
}
```

i. User.java

```
package com.cea.cares;
public class User {
   private int id,target,week;
   private String username, email;
   public User(int id, int target, String username, String email, int week) {
      this.id = id;
      this.target = target;
      this.username = username;
      this.email = email;
      this.week = week;
   }
   public int getId() {
      return id;
   }
   public int getTarget() {
      return target;
   }
   public String getUsername() {
      return username;
   }
   public String getEmail() {
      return email;
   }
   public int getWeek() { return week; }
}
```

j. WeekActivity.java

```
package com.cea.cares;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Objects;
public class WeekActivity extends AppCompatActivity {
    User user;
    String id,weekNo,currentDate,selectedDate;
    TextView textView;
    private Calendar calendar;
    private int year, month, day;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.week_activity);
        EditText week = findViewById(R.id.mtrl_calendar_days_of_week);
        Button buttonWeek = findViewById(R.id.buttonweek);
        user = SharedPrefManager.getInstance(getApplicationContext()).getUser();
//        weekNo  = week.getText().toString().trim();
        id = String.valueOf(user.getId());
        textView = findViewById(R.id.textView3);
                                currentDate = new SimpleDateFormat("dd/M/yyyy",
Locale.getDefault()).format(new Date());
        calendar = Calendar.getInstance();
        year = calendar.get(Calendar.YEAR);
        month = calendar.get(Calendar.MONTH);
        day = calendar.get(Calendar.DAY_OF_MONTH);
        showDate(year, month+1, day);
        buttonWeek.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                weekNo  = week.getText().toString().trim();
                if(Objects.equals(weekNo, "")){
                    Date date1;
                    Date date2;
                    SimpleDateFormat dates = new SimpleDateFormat("dd/MM/yyyy");
                    try {
                        date1 = dates.parse(currentDate);
                        date2 = dates.parse(selectedDate);
```

```
                long difference =  Math.abs(date1.getTime() - date2.getTime());
                long differenceDates = difference / (24 * 60 * 60 * 1000);
                weekNo = String.valueOf(differenceDates / 7);
            } catch (ParseException e) {
                e.printStackTrace();
            }}
                                else if (Objects.equals(currentDate,selectedDate) &&
Objects.equals(weekNo, ""))
            weekNo = "1";
                                        Toast.makeText(getApplicationContext(),weekNo,
Toast.LENGTH_LONG).show();
            updateWeek();
        }});
    }
    public void updateWeek(){
        class UpdateWeek extends AsyncTask<Void, Void, String> {
            @Override
            protected String doInBackground(Void... voids) {
                RequestHandler requestHandler = new RequestHandler();
                HashMap<String, String> params = new HashMap<>();
                params.put("id",id);
                params.put("week",weekNo);
                return requestHandler.sendPostRequest(URLs.URL_WEEK,params);
            }
            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                try {
```

```
                JSONObject obj = new JSONObject(s);
                if (!obj.getBoolean("error")) {
                    Toast.makeText(getApplicationContext(), obj.getString("message"),
Toast.LENGTH_SHORT).show();
                    JSONObject userJson = obj.getJSONObject("user");
                    User user = new User(
                            userJson.getInt("id"),
                            userJson.getInt("target"),
                            userJson.getString("username"),
                            userJson.getString("email"),
                            userJson.getInt("week")
                    );
            SharedPrefManager.getInstance(getApplicationContext()).userLogin(user);
                    finish();
                                        startActivity(new  Intent(getApplicationContext(),
MainActivity.class));
                } else {
                        Toast.makeText(getApplicationContext(), "Some error occurred",
Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }}}
        UpdateWeek uw = new UpdateWeek();
        uw.execute();
    }
    @SuppressWarnings("deprecation")
    public void setDate(View view) {
```

```
        showDialog(999);
        Toast.makeText(getApplicationContext(), "ca",
                Toast.LENGTH_SHORT)
            .show();
    }
    @Override
    protected Dialog onCreateDialog(int id) {
        if (id == 999) {
            return new DatePickerDialog(this,
                    myDateListener, year, month, day);
        }
        return null;
    }
        private final DatePickerDialog.OnDateSetListener myDateListener =
this::onDateSet;
    private void onDateSet(DatePicker arg0, int arg1, int arg2, int arg3) {
        /*
         TODO Auto-generated method stub
         arg1 = year
         arg2 = month
         arg3 = day
        */
        showDate(arg1, arg2 + 1, arg3);
    }
    private void showDate(int year, int month, int day) {
        selectedDate = day + "/" + month + "/" + year;
        textView.setText(selectedDate);
    }}
```

UI

k. ActivityFragment.java

```
package com.cea.cares.ui.activity;
import android.annotation.SuppressLint;
import android.app.AlarmManager;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import androidx.fragment.app.Fragment;
import com.cea.cares.Background;
import com.cea.cares.R;
import com.cea.cares.RequestHandler;
```

```
import com.cea.cares.SharedPrefManager;
import com.cea.cares.URLs;
import com.cea.cares.User;
import com.cea.cares.databinding.FragmentActivityBinding;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Objects;
public class ActivityFragment extends Fragment {
   private FragmentActivityBinding binding;
   String week;
   TextView textViewExc;
   User user;
   public View onCreateView(@NonNull LayoutInflater inflater,
                            ViewGroup container, Bundle savedInstanceState) {
      binding = FragmentActivityBinding.inflate(inflater, container, false);
      View root = binding.getRoot();
      EditText drDate = binding.drDate;
      textViewExc = binding.exc;
      user = SharedPrefManager.getInstance(getActivity()).getUser();
      week = String.valueOf(user.getWeek());
      getExc();
      Button button = binding.notification;
      button.setOnClickListener(new View.OnClickListener() {
```

```
@SuppressLint("ShortAlarm")
@Override
public void onClick(View view) {
    String appDate = drDate.getText().toString().trim();
                String currentDate = new SimpleDateFormat("dd/M/yyyy",
Locale.getDefault()).format(new Date());
    if(Objects.equals(appDate, ""))
                    Toast.makeText(getActivity(),"Please enter the date",
Toast.LENGTH_LONG).show();
    else{
        Date date1;
        Date date2;
        SimpleDateFormat dates = new SimpleDateFormat("dd/MM/yyyy");
        try {
            date1 = dates.parse(currentDate);
            date2 = dates.parse(appDate);
            long difference =  Math.abs(date1.getTime() - date2.getTime());
            long differenceDates = (difference / (24 * 60 * 60 * 1000)) - 1;
        } catch (ParseException e) {
            e.printStackTrace();
        }
        Intent intent = new Intent(getActivity(), Background.class);
        PendingIntent pendingIntent = endingIntent.getBroadcast(getContext(),
0, intent, 0);
                    AlarmManager alarmManager = (AlarmManager)
getActivity().getSystemService(Context.ALARM_SERVICE);
        long interval = 1000 * 10;
```

```
alarmManager.setExact(AlarmManager.RTC_WAKEUP,interval,pendingIntent);
                //alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, 20000,
interval, pendingIntent);
                                    Toast.makeText(getContext(), "Notification Set",
Toast.LENGTH_SHORT).show();
    }}});
      return root;
   }
   @Override
   public void onDestroyView() {
      super.onDestroyView();
      binding = null;
   }
   public void addNotification( Context context) {
0);
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
          NotificationChannel channel = new NotificationChannel("my notification",
"my notification", NotificationManager.IMPORTANCE_HIGH);
                                        NotificationManager manager =
context.getSystemService(NotificationManager.class);
         manager.createNotificationChannel(channel);
      }
      NotificationCompat.Builder builder =
            new NotificationCompat.Builder(context,"my notification");
            builder.setContentTitle("New notification");
            builder.setContentText("Subject");
            builder.setSmallIcon(R.drawable.ic_home_black_24dp);
```

```
            builder.setAutoCancel(true);
                                NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(context);
        notificationManager.notify(0, builder.build());
    }
    public void getExc() {
        class GetExc extends AsyncTask<Void, Void, String> {
            @Override
            protected String doInBackground(Void... voids) {
                RequestHandler requestHandler = new RequestHandler();
                HashMap<String, String> params = new HashMap<>();
                params.put("week",week);
                                return requestHandler.sendPostRequest(URLs.URL_EXC,
params);
            }
            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                try {
                    JSONObject obj = new JSONObject(s);
                    if (!obj.getBoolean("error")) {
                                    Toast.makeText(getActivity(), obj.getString("message"),
Toast.LENGTH_SHORT).show();
                        String exc = obj.getString( "exc");
                            textViewExc.setText("These are the recommended exercises \n " +
exc);
                    } else {
```

```
                                    Toast.makeText(getActivity(), "Some error occurred",
Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }}}
        GetExc ge = new GetExc();
        ge.execute();
    }}
```

l. HomeFragment.java

```
package com.cea.cares.ui.home;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import com.cea.cares.RequestHandler;
import com.cea.cares.SharedPrefManager;
import com.cea.cares.URLs;
import com.cea.cares.User;
import com.cea.cares.databinding.FragmentHomeBinding;
import org.json.JSONException;
```

```
import org.json.JSONObject;
import java.util.HashMap;
public class HomeFragment extends Fragment {
   private FragmentHomeBinding binding;
   private  String id,week;
   TextView textViewpredict,textViewFood;
   User user;
   int target;
   public View onCreateView(@NonNull LayoutInflater inflater,
                            ViewGroup container, Bundle savedInstanceState) {
      binding = FragmentHomeBinding.inflate(inflater, container, false);
      View root = binding.getRoot();
      TextView textViewUsername = binding.textViewUsername;
      textViewpredict = binding.textViewpredict;
      textViewFood = binding.food;
      user = SharedPrefManager.getInstance(getActivity()).getUser();
      String text = "Hi " + user.getUsername();
      textViewUsername.setText(text);
      id = String.valueOf(user.getId());
      week = String.valueOf(user.getWeek());
      target = user.getTarget();
      if(target == 1)
         textViewpredict.setText("you and your baby is just fine" + "\nthis is your " +
week + " week");
      else if (target == 0)
          textViewpredict.setText("Our baby is underweight!!!"+ "\nthis is your " +
week + " week");
      getFood();
```

```
        Button button = binding.buttonpredict;
        button.setOnClickListener(view -> predict());
        return root;
    }
    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
    public void predict(){
        class Predict extends AsyncTask<Void, Void, String> {
            @Override
            protected String doInBackground(Void... voids) {
                        RequestHandler requestHandler = new RequestHandler();
                       HashMap<String, String> params = new HashMap<>();
                params.put("id",id);
                                                                    return
requestHandler.sendPostRequest(URLs.URL_PREDICT+id,params);
            }
            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                                                Toast.makeText(getActivity(),"Predicted
Successfully",Toast.LENGTH_SHORT).show();
                //HomeViewModel homeViewModel = new HomeViewModel();
                //homeViewModel.refreshUser(requireContext());
                refreshUser();
            }}
```

```
        Predict pr = new Predict();
        pr.execute();
    }
    public void refreshUser() {
        class RefreshUser extends AsyncTask<Void, Void, String> {
            @Override
            protected String doInBackground(Void... voids) {
                RequestHandler requestHandler = new RequestHandler();
                HashMap<String, String> params = new HashMap<>();
                params.put("id",id);
                return requestHandler.sendPostRequest(URLs.URL_REFRESH, params);
            }
            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                try {
                    JSONObject obj = new JSONObject(s);
                    if (!obj.getBoolean("error")) {
                                Toast.makeText(getActivity(),  obj.getString("message"),
Toast.LENGTH_SHORT).show();
                        JSONObject userJson = obj.getJSONObject("user");
                        User user = new User(
                                userJson.getInt("id"),
                                userJson.getInt("target"),
                                userJson.getString("username"),
                                userJson.getString("email"),
                                userJson.getInt("week")
                        );
```

```
                SharedPrefManager.getInstance(getActivity()).userLogin(user);
                                                        User usercpy =
SharedPrefManager.getInstance(getActivity()).getUser();
                int res = usercpy.getTarget();
                int week = usercpy.getWeek();
                if(res == 1)
                    textViewpredict.setText("you and your baby is just fine" + "\nthis
is your " + week + " week");
                else if (res == 0)
                     textViewpredict.setText("Our baby is underweight!!!"+ "\nthis is
your " + week + " week");
              } else {
                                Toast.makeText(getActivity(), "Some error occurred",
Toast.LENGTH_SHORT).show();
               }
            } catch (JSONException e) {
              e.printStackTrace();
            }}}
      RefreshUser ru = new RefreshUser();
      ru.execute();
   }
   public void getFood() {
      class GetFood extends AsyncTask<Void, Void, String> {
         @Override
         protected String doInBackground(Void... voids) {
            RequestHandler requestHandler = new RequestHandler();
            //creating request parameters
            HashMap<String, String> params = new HashMap<>();
```

```
            params.put("week",week);
            return requestHandler.sendPostRequest(URLs.URL_FOOD, params);
        }
        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
          try {
              JSONObject obj = new JSONObject(s);
              if (!obj.getBoolean("error")) {
                                  Toast.makeText(getActivity(),  obj.getString("message"),
Toast.LENGTH_SHORT).show();
                  String food = obj.getString( "food");
                   textViewFood.setText("These are the recommended food items\n" +
food);                } else {
                                    Toast.makeText(getActivity(),  "Some  error  occurred",
Toast.LENGTH_SHORT).show();
              }
            } catch (JSONException e) {
              e.printStackTrace();
            }}}
      GetFood gf = new GetFood();
      gf.execute();
    }}
```

m. MoreActionsFragment.java

```
package com.cea.cares.ui.moreactions;
import android.os.Bundle;
import android.view.LayoutInflater;
```

```
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import com.cea.cares.R;
import com.cea.cares.RequestHandler;
import com.cea.cares.SharedPrefManager;
import com.cea.cares.URLs;
import com.cea.cares.User;
import com.cea.cares.databinding.FragmentMoreactionsBinding;
import java.util.HashMap;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
public class MoreActionsFragment extends Fragment {
   private FragmentMoreactionsBinding binding;
   private ScheduledExecutorService scheduleTaskExecutor;
   int week;
   User user;
   public View onCreateView(@NonNull LayoutInflater inflater,
                            ViewGroup container, Bundle savedInstanceState) {
      binding = FragmentMoreactionsBinding.inflate(inflater, container, false);
      View root = binding.getRoot();
      final TextView textView = binding.textDashboard;
      textView.setText("More Actions");
```

```
        final Button button = binding.buttonLogout;
                                                button.setOnClickListener(view ->
    SharedPrefManager.getInstance(getActivity()).logout());
        scheduleTaskExecutor = Executors.newScheduledThreadPool(15);
        //Schedule a task to run every 5 seconds (or however long you want)
        scheduleTaskExecutor.scheduleAtFixedRate(new Runnable() {
            @Override
            public void run() {
                user = SharedPrefManager.getInstance(getActivity()).getUser();
                week = user.getWeek() + 1;
                //creating request handler object
                RequestHandler requestHandler = new RequestHandler();
                //creating request parameters
                HashMap<String, String> params = new HashMap<>();
                params.put("week",String.valueOf(week));
                //returing the response
                requestHandler.sendPostRequest(URLs.URL_WEEK, params);
            }
        }, 0, 15, TimeUnit.SECONDS); // or .MINUTES, .HOURS etc.
        return root;
    }
    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

n. NotificationsViewModel.java

```
package com.cea.cares.ui.profile;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;
import androidx.lifecycle.ViewModel;
public class NotificationsViewModel extends ViewModel {
   private final MutableLiveData<String> mText;
   public NotificationsViewModel() {
      mText = new MutableLiveData<>();
      mText.setValue("This is notifications fragment");
   }
   public LiveData<String> getText() {
      return mText;
   }}
```

o. ProfileFragment.java

```
package com.cea.cares.ui.profile;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.ViewModelProvider;
import com.cea.cares.SharedPrefManager;
```

```
import com.cea.cares.User;
import com.cea.cares.databinding.FragmentProfileBinding;
public class ProfileFragment extends Fragment {
    User user;
    String username,email;
    private FragmentProfileBinding binding;
    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        binding = FragmentProfileBinding.inflate(inflater, container, false);
        View root = binding.getRoot();
        final TextView textView = binding.textNotifications;
        textView.setText("This is profile Fragment");
        user = SharedPrefManager.getInstance(getActivity()).getUser();
        username = user.getUsername();
        email = user.getEmail();
        TextView usernameemail = binding.UsernameEmail;
        usernameemail.setText(username + "\n" + email);
        Button button = binding.update;
        button.setOnClickListener(this::profileupdate);
        return root;
    }
    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
    public void profileupdate(View view){
        Intent intent = new Intent(getActivity(), UpdateProfile.class);
```

```
        startActivity(intent);
    }}
```

p. UpdateProfile.java

```
package com.cea.cares.ui.profile;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.cea.cares.MainActivity;
import com.cea.cares.R;
import com.cea.cares.RequestHandler;
import com.cea.cares.SharedPrefManager;
import com.cea.cares.URLs;
import com.cea.cares.User;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Objects;
public class UpdateProfile extends AppCompatActivity {
    EditText orderGravida,age,height,weight,midarmcir;
```

RadioGroup workload,fever,anemia, iron2, bleed, asthma, convolusion2, bad_obs_history, injection,falif, iron, workload2, convolusion, bleed1, asthma2, inject2;

```
String temp ;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.update_profile);
    orderGravida = findViewById(R.id.orderGravida);
    age = findViewById(R.id.age);
    height = findViewById(R.id.height);
    weight = findViewById(R.id.weight);
    midarmcir = findViewById(R.id.midarmcir);
    workload = findViewById(R.id.radioWorkload);
    anemia = findViewById(R.id.radioAnemia);
    asthma = findViewById(R.id.radioAsthma);
    bad_obs_history = findViewById(R.id.radioBadh);
    injection = findViewById(R.id.radioInjection);
    falif = findViewById(R.id.radioFalif);
    iron = findViewById(R.id.radioIron);
    workload2 = findViewById(R.id.radioSecWrk);
    convolusion = findViewById(R.id.radioConv);
    bleed1 = findViewById(R.id.radioBleed);
    asthma2 = findViewById(R.id.radioSecAst);
    inject2 = findViewById(R.id.radioSecInj);
    iron2 = findViewById(R.id.radioSecIrn);
    convolusion2 = findViewById(R.id.radioSecCnv);
    bleed = findViewById(R.id.radioSecBld);
```

```
        fever = findViewById(R.id.radioFvr);
        Button submit = findViewById(R.id.submit);
        submit.setOnClickListener(view -> profileUpdate());
    }
    private void profileUpdate(){
User user = SharedPrefManager.getInstance(getApplicationContext()).getUser();
        String id = String.valueOf(user.getId());
        int wl = workload.getCheckedRadioButtonId();
        RadioButton radioButtonwl =  findViewById(wl);
        if(wl==-1){
        }
        else{
            if(Objects.equals(radioButtonwl.getText(), "yes"))
                temp = "1.0";
            else
                temp = "0.0";
        }
        final String wkload = temp ;
        int anm = anemia.getCheckedRadioButtonId();
        RadioButton radioButtonanm =  findViewById(anm);
        if(anm==-1){
        }
        else{
            if(Objects.equals(radioButtonanm.getText(), "yes"))
                temp = "1.0";
            else
                temp = "0.0";
        }
```

```
final String anmia = temp ;
int ast = asthma.getCheckedRadioButtonId();
RadioButton radioButtonast =  findViewById(ast);
if(ast==-1){
}
else{
   if(Objects.equals(radioButtonast.getText(), "yes"))
      temp = "1.0";
   else
      temp = "0.0";
}
final String astma = temp ;
int bahs = bad_obs_history.getCheckedRadioButtonId();
RadioButton radioButtonbahs =  findViewById(bahs);
if(bahs==-1){
}
else{
   if(Objects.equals(radioButtonbahs.getText(), "yes"))
      temp = "1.0";
   else
      temp = "0.0";
}
final String bad_histry = temp ;
int injct = injection.getCheckedRadioButtonId();
RadioButton radioButtoninjct =  findViewById(injct);
if(injct==-1){
}
else{
```

```
    if(Objects.equals(radioButtoninjct.getText(), "yes"))
        temp = "1.0";
    else
        temp = "0.0";
}
final String inj = temp ;
int flf = falif.getCheckedRadioButtonId();
RadioButton radioButtonflf =  findViewById(flf);
if(flf==-1){
}
else{
    if(Objects.equals(radioButtonflf.getText(), "yes"))
        temp = "1.0";
    else
        temp = "0.0";
}
final String flif = temp ;
int irn = iron.getCheckedRadioButtonId();
RadioButton radioButtonirn =  findViewById(irn);
if(irn==-1){
}
else{
    if(Objects.equals(radioButtonirn.getText(), "yes"))
        temp = "1.0";
    else
        temp = "0.0";
}
final String fe = temp ;
```

```
int wl2 = workload2.getCheckedRadioButtonId();
RadioButton radioButtonwl2 =  findViewById(wl2);
if(wl2==-1){
}
else{
   if(Objects.equals(radioButtonwl2.getText(), "yes"))
      temp = "1.0";
   else
      temp = "0.0";
}
final String wkload2 = temp ;
int clnv = convolusion.getCheckedRadioButtonId();
RadioButton radioButtonclnv =  findViewById(clnv);
if(clnv==-1){
}
else{
   if(Objects.equals(radioButtonclnv.getText(), "yes"))
      temp = "1.0";
   else
      temp = "0.0";
}
final String cnv = temp ;
int bled = bleed1.getCheckedRadioButtonId();
RadioButton radioButtonbled =  findViewById(bled);
if(bled==-1){
}
else{
   if(Objects.equals(radioButtonbled.getText(), "yes"))
```

```
        temp = "1.0";
    else
        temp = "0.0";
}
final String bld = temp ;
int ast2 = asthma2.getCheckedRadioButtonId();
RadioButton radioButtonast2 =  findViewById(ast2);
if(ast==-1){
}
else{
    if(Objects.equals(radioButtonast2.getText(), "yes"))
        temp = "1.0";
    else
        temp = "0.0";
}
final String astma2 = temp ;
int injct2 = inject2.getCheckedRadioButtonId();
RadioButton radioButtoninjct2 =  findViewById(injct2);
if(injct2==-1){
}
else{
    if(Objects.equals(radioButtoninjct2.getText(), "yes"))
        temp = "1.0";
    else
        temp = "0.0";
}
final String inj2 = temp ;
int irn2 = iron2.getCheckedRadioButtonId();
```

```
RadioButton radioButtonirn2 =  findViewById(irn2);
if(irn2==-1){
}
else{
   if(Objects.equals(radioButtonirn2.getText(), "yes"))
      temp = "1.0";
   else
      temp = "0.0";
}
final String fe2 = temp ;
int clnv2 = convolusion2.getCheckedRadioButtonId();
RadioButton radioButtonclnv2 =  findViewById(clnv2);
if(clnv2==-1){
}
else{
   if(Objects.equals(radioButtonclnv2.getText(), "yes"))
      temp = "1.0";
   else
      temp = "0.0";
}
final String cnv2 = temp ;
int bled2 = bleed.getCheckedRadioButtonId();
RadioButton radioButtonbled2 =  findViewById(bled2);
if(bled2==-1){
}
else{
   if(Objects.equals(radioButtonbled2.getText(), "yes"))
      temp = "1.0";
```

```
    else
        temp = "0.0";
}
final String bld2 = temp ;
int fvrr = fever.getCheckedRadioButtonId();
RadioButton radioButtonfvrr =  findViewById(fvrr);
if(fvrr==-1){
}
else{
    if(Objects.equals(radioButtonfvrr.getText(), "yes"))
        temp = "1.0";
    else
        temp = "0.0";
}
final String fvr = temp ;
final String OrderGravida = orderGravida.getText().toString().trim();
final String mage = age.getText().toString().trim();
final String hght = height.getText().toString().trim();
final String wght = weight.getText().toString().trim();
final String mid_arm_cir = midarmcir.getText().toString().trim();
float h = Float.parseFloat(hght);
int w = Integer.parseInt(wght);
final String bmi = String.valueOf(w/(h*h));
class ProfileUpdate extends AsyncTask<Void, Void, String> {
    private ProgressBar progressBar;
    @Override
    protected String doInBackground(Void... voids) {
        RequestHandler requestHandler = new RequestHandler();
```

```
                HashMap<String, String> params = new HashMap<>();
                params.put("id",id);
                params.put("parity", OrderGravida);
                params.put("mother_age", mage);
                params.put("mother_height", hght);
                params.put("mid_arm_cir", mid_arm_cir);
                params.put("workload", wkload);
                params.put("anemia", anmia);
                params.put("asthma", astma);
                params.put("bad_obs_history", bad_histry);
                params.put("injection", inj);
                params.put("falif", flif);
                params.put("iron", fe);
                params.put("workload2", wkload2);
                params.put("convolusion", cnv);
                params.put("bleed", bld);
                params.put("asthma2", astma2);
                params.put("inject2", inj2);
                params.put("iron2", fe2);
                params.put("convolusion2", cnv2);
                params.put("bleed2", bld2);
                params.put("fever", fvr);
                params.put("BMI", bmi);
                        return requestHandler.sendPostRequest(URLs.URL_UPDATE,
params);
        }
        @Override
        protected void onPreExecute() {
```

```
        super.onPreExecute();
                progressBar = (ProgressBar) findViewById(R.id.progressBar);
        progressBar.setVisibility(View.VISIBLE);
      }
      @Override
      protected void onPostExecute(String s) {
        super.onPostExecute(s);
                 progressBar.setVisibility(View.GONE);
              } }
    ProfileUpdate pu = new ProfileUpdate();
     pu.execute();
     startActivity(new Intent(this, MainActivity.class));
  }}
```

Layout

q. activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   android:id="@+id/container"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingTop="?attr/actionBarSize">
   <com.google.android.material.bottomnavigation.BottomNavigationView
      android:id="@+id/nav_view"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
```

```
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu" />
    <fragment
        android:id="@+id/nav_host_fragment_activity_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

r. fragment_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
tools:context=".ui.activity.ActivityFragment">
<TextView
   android:id="@+id/dateDr"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_marginStart="8dp"
   android:layout_marginEnd="8dp"
   android:layout_marginBottom="24dp"
   android:text="Please enter the next date of Doctor's appointment"
   android:textAlignment="center"
   app:layout_constraintBottom_toTopOf="@+id/drDate"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toTopOf="parent"
   tools:ignore="MissingConstraints" />
<EditText
   android:id="@+id/drDate"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_marginTop="24dp"
   android:hint="(dd/mm/yyyy)"
   app:layout_constraintBottom_toTopOf="@+id/notification"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toBottomOf="@+id/dateDr"
   tools:ignore="MissingConstraints" />
<Button
   android:id="@+id/notification"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="161dp"
        android:text="Notify"
        app:layout_constraintBottom_toTopOf="@+id/exc"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/drDate"
        tools:ignore="MissingConstraints" />
    <TextView
        android:id="@+id/exc"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="203dp"
        android:layout_marginBottom="60dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.49"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/notification"
        tools:ignore="MissingConstraints" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

s. fragment_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.home.HomeFragment">
    <TextView
        android:id="@+id/textViewUsername"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="20dp"
        android:padding="10dp"
        android:text="Username"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <androidx.cardview.widget.CardView
        android:id="@+id/cardView"
        android:layout_width="375dp"
        android:layout_height="173dp"
        android:layout_below="@id/textViewUsername"
        android:layout_alignParentTop="true"
        android:layout_marginStart="15dp"
        android:layout_marginTop="70dp"
        android:minHeight="200dp"
        app:cardCornerRadius="16dp"
```

```
app:cardElevation="@dimen/card_home_elevation"
app:cardPreventCornerOverlap="false"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
tools:ignore="MissingConstraints">
<androidx.constraintlayout.widget.ConstraintLayout
   android:layout_width="match_parent"
   android:layout_height="173dp"
   android:layout_gravity="center_vertical"
   android:background="@drawable/gradient_predict"
   android:orientation="vertical"
   android:padding="10dp"
   tools:layout_editor_absoluteX="4dp"
   tools:layout_editor_absoluteY="107dp">
   <TextView
      android:id="@+id/textViewpredict"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="10dp"
      android:text="prediction here"
      android:textColor="@color/white"
      app:layout_constraintTop_toTopOf="parent"
      tools:layout_editor_absoluteX="10dp" />
   <ImageView
      android:layout_width="228dp"
      android:layout_height="187dp"
      android:layout_alignParentTop="true"
      android:layout_alignParentEnd="true"
```

```
            android:layout_marginStart="172dp"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="5dp"
            android:adjustViewBounds="true"
            android:cropToPadding="true"
            android:src="@mipmap/cares_logo_foreground"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
        <Button
            android:id="@+id/buttonpredict"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Predict"
            app:layout_constraintBottom_toBottomOf="parent"
            tools:layout_editor_absoluteX="10dp" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:layout_width="375dp"
    android:layout_height="173dp"
    android:layout_alignParentTop="true"
    android:layout_marginStart="15dp"
    android:layout_marginBottom="-700dp"
    android:minHeight="200dp"
    app:cardCornerRadius="16dp"
    app:cardElevation="@dimen/card_home_elevation"
    app:cardPreventCornerOverlap="false"
```

```
        app:layout_constraintBottom_toTopOf="@+id/cardView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints">
        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="173dp"
            android:layout_gravity="center_vertical"
            android:background="@drawable/gradient_food"
            android:orientation="vertical"
            android:padding="10dp"
            tools:layout_editor_absoluteX="4dp"
            tools:layout_editor_absoluteY="107dp">
            <TextView
                android:id="@+id/food"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="5dp"
                android:textColor="@color/white"
                android:textSize="20sp"
                app:layout_constraintTop_toTopOf="parent"
                tools:ignore="MissingConstraints"
                tools:layout_editor_absoluteX="0dp" />
        </androidx.constraintlayout.widget.ConstraintLayout>
    </androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

t. fragment_moreactions.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.moreactions.MoreActionsFragment">
    <TextView
        android:id="@+id/text_dashboard"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Logout"
        tools:ignore="MissingConstraints"
        android:id="@+id/buttonLogout" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

u. fragment_profile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".ui.profile.ProfileFragment">
   <TextView
      android:id="@+id/text_notifications"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:layout_marginEnd="8dp"
      android:textAlignment="center"
      android:textSize="20sp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />
   <Button
      android:id="@+id/update"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginBottom="56dp"
      android:text="Update"
```

```
        android:textAlignment="center"
        app:layout_constraintBottom_toBottomOf="parent"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="150dp" />
    <TextView
        android:id="@+id/UsernameEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="68dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="0dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

v. login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:orientation="vertical"
```

```
android:padding="10dp">
<EditText
    android:id="@+id/editTextUsername"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:hint="Username"
    android:inputType="text" />
<EditText
    android:id="@+id/editTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:fontFamily="sans-serif"
    android:hint="Password"
    android:inputType="textPassword" />
<Button
    android:id="@+id/buttonLogin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:text="Login" />
<TextView
    android:id="@+id/textViewRegister"
    android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:layout_marginTop="8dp"
            android:text="Dont' have an account?\nRegister Here"
            android:textAlignment="center"
          android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"
/>   </LinearLayout>
   <ProgressBar
      android:visibility="gone"
      android:id="@+id/progressBar"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_centerVertical="true" />
</RelativeLayout>
```

w. register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_centerVertical="true"
      android:orientation="vertical"
```

```
android:padding="10dp">
<EditText
    android:id="@+id/editTextUsername"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:hint="Username"
    android:inputType="text" />
<EditText
    android:id="@+id/editTextEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:hint="Email"
    android:inputType="textEmailAddress" />
<EditText
    android:id="@+id/editTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:fontFamily="sans-serif"
    android:hint="Password"
    android:inputType="textPassword" />
<Button
    android:id="@+id/buttonRegister"
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:layout_marginTop="8dp"
            android:text="Register" />
        <TextView
            android:id="@+id/textViewLogin"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:layout_marginTop="8dp"
            android:text="Already Registered?\nLogin Here"
            android:textAlignment="center"
         android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"
/>    </LinearLayout>
    <ProgressBar
        android:visibility="gone"
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true" />
</RelativeLayout>
```

x. update_profile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".ui.profile.UpdateProfile">
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginEnd="4dp"
    app:layout_constraintEnd_toEndOf="parent"
    tools:layout_editor_absoluteY="66dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:text="Hello" />
        <EditText
            android:id="@+id/orderGravida"
            android:layout_width="287dp"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Order of Gravida (Enter in numeric form)"
            app:layout_constraintStart_toStartOf="parent"
            tools:layout_editor_absoluteY="20dp" />
        <EditText
```

```
    android:id="@+id/age"
    android:layout_width="287dp"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Age"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="20dp" />
<EditText
    android:id="@+id/height"
    android:layout_width="287dp"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Height in meters"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="20dp" />
<EditText
    android:id="@+id/weight"
    android:layout_width="287dp"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Weight in kg"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="20dp" />
<EditText
    android:id="@+id/midarmcir"
    android:layout_width="287dp"
    android:layout_height="wrap_content"
    android:ems="10"
```

```
    android:hint="Mid arm circumferene in meters"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="20dp" />
<!-- Workload -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Workload"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioWorkload"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/one"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
```

```
        <RadioButton
            android:id="@+id/two"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Anemia -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Anemia"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioAnemia"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/three"
```

```
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/four"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Asthma -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Asthma"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioAsthma"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <RadioButton
                android:id="@+id/five"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:checked="false"
                android:text="yes" />
            <RadioButton
                android:id="@+id/six"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:checked="false"
                android:text="no" />
        </RadioGroup>
    </LinearLayout>
    <!-- Bad obsteric history -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Bad obsteric history"
```

```
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioBadh"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/seven"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/eight"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Injection -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:orientation="vertical">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Injection"
    android:textSize="20dp"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="20dp" />
<RadioGroup
    android:id="@+id/radioInjection"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <RadioButton
        android:id="@+id/nine"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:checked="false"
        android:text="yes" />
    <RadioButton
        android:id="@+id/ten"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:checked="false"
        android:text="no" />
</RadioGroup>
```

```
</LinearLayout>
<!-- Falif -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Diabetic"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioFalif"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/eleven"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/twelve"
            android:layout_width="fill_parent"
```

```
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Iron -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Iron intake"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioIron"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/thirteen"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
```

```
            android:checked="false"
            android:text="yes"
            tools:ignore="DuplicateIds" />
        <RadioButton
            android:id="@+id/fourteen"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Second Workload -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Workload"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioSecWrk"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/fifteen"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/sixteen"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Convolusion -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Convolusion"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
```

```
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioConv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/seventeen"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/eighteen"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Bleed -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bleed"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioBleed"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/ninteen"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/twenty"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Second Asthma -->
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Asthma"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioSecAst"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/twentyone"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/twentytwo"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
```

```
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Second Injection -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Injection"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />

    <RadioGroup
        android:id="@+id/radioSecInj"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/twentythree"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
```

```
            android:text="yes" />
        <RadioButton
            android:id="@+id/twentyfour"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Second Iron -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Iron"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioSecIrn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
     <RadioButton
```

```
            android:id="@+id/twentyfive"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/twentysix"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<!-- Second Convulsion -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Convulsion"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
```

```
            android:id="@+id/radioSecCnv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <RadioButton
                android:id="@+id/twentyseven"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:checked="false"
                android:text="yes" />
            <RadioButton
                android:id="@+id/twentyeight"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:checked="false"
                android:text="no" />
        </RadioGroup>
    </LinearLayout>
    <!-- Second Bleed -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:text="Second Bleed"
        android:textSize="20dp"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="20dp" />
    <RadioGroup
        android:id="@+id/radioSecBld"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/twentynine"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="yes" />
        <RadioButton
            android:id="@+id/thirty"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:checked="false"
            android:text="no" />
    </RadioGroup>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:orientation="vertical">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Fever"
    android:textSize="20dp"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="20dp" />
<RadioGroup
    android:id="@+id/radioFvr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <RadioButton
        android:id="@+id/thirtyone"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:checked="false"
        android:text="yes" />
    <RadioButton
        android:id="@+id/thirytwo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:checked="false"
        android:text="no" />
</RadioGroup>
```

```
        </LinearLayout>
        <Button
            android:id="@+id/submit"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Submit" />
        <ProgressBar
            android:id="@+id/progressBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:visibility="gone" />
    </LinearLayout>
  </ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

y. week_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:text="Please let us know the first day of your last menstruation"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="-16dp" />
<Button
    android:id="@+id/buttondate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:onClick="setDate"
    android:text="SET date"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="16dp" />
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="92dp"
    android:text="or"
    app:layout_constraintTop_toBottomOf="@+id/buttondate"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="-16dp" />
<TextView
```

```
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Let us know which week is this"
        app:layout_constraintBottom_toTopOf="@+id/mtrl_calendar_days_of_week"
        app:layout_constraintTop_toBottomOf="@+id/textView3"
        app:layout_constraintVertical_bias="0.807"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="0dp" />
    <EditText
        android:id="@+id/mtrl_calendar_days_of_week"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@+id/buttonweek"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="0.815"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="0dp" />
    <Button
        android:id="@+id/buttonweek"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="228dp"
        app:layout_constraintBottom_toBottomOf="parent"
        tools:layout_editor_absoluteX="-16dp"
        android:text="Submit"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Menu

z. Bottom_nav_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="@string/title_home" />
    <item
        android:id="@+id/navigation_activity"
        android:icon="@drawable/ic_dashboard_black_24dp"
        android:title="@string/title_activity" />
    <item
        android:id="@+id/navigation_profile"
        android:icon="@drawable/ic_notifications_black_24dp"
        android:title="@string/title_profile" />
    <item
        android:id="@+id/navigation_moreactions"
        android:icon="@drawable/ic_notifications_black_24dp"
        android:title="@string/title_moreactions" />
</menu>
```

Navigation

aa. mobile_navigation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:id="@+id/mobile_navigation"
    app:startDestination="@+id/navigation_home">
    <fragment
        android:id="@+id/navigation_home"
        android:name="com.cea.cares.ui.home.HomeFragment"
        android:label="@string/title_home"
        tools:layout="@layout/fragment_home" />
    <fragment
        android:id="@+id/navigation_activity"
        android:name="com.cea.cares.ui.activity.ActivityFragment"
        android:label="@string/title_activity"
        tools:layout="@layout/fragment_activity" />
    <fragment
        android:id="@+id/navigation_profile"
        android:name="com.cea.cares.ui.profile.ProfileFragment"
        android:label="@string/title_profile"
        tools:layout="@layout/fragment_profile" />
    <fragment
        android:id="@+id/navigation_moreactions"
        android:name="com.cea.cares.ui.moreactions.MoreActionsFragment"
        android:label="@string/title_moreactions"
        tools:layout="@layout/fragment_moreactions" />
</navigation>
```

REFERENCES

[1] "Prediction and Classification of Low Birth Weight Data Using Machine Learning Techniques" Indonesian Journal of Science & Technology, Alfensi Faruk, Endro Setyo Cahyono, Ning Eliyati, IkaArifieni(2018)

[2] "Risk factors for low birth weight in nigeria: evidence from the 2013 Nigeria demographic and health survey. Global Health Action",2016 , Dahlui, M., Azahar, N., Oche, O. C., and Aziz, N. A.

[3] Machine Learning mastery with Python, Jason Brownlee.

[4] "Random Decision Forests", Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Ho, Tin Kam

[5] "Android Application Development using Android Studio and PHP Framework", International Journal Of Computer Application: Recent Trends in Future Prospective in Engineering & Management Technology 2016 Akshay Singh, Sakshi Sharma, Shashwat Singh, IMS Engineering College.

[6] "Personal Health Assistant On Android Mobile Device: Sleeping, Nutrition and Exercise" Kiatateeti, Sangsuree Vasupongayya and Sinchai, Center of Network Research, Prince of Songkla University, Thailand.

[7] Niti Aayog Annual Report 2021-22 (Published Date: 22/02/2022).

[8] "Random Forest and Decision Trees", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012, Sarhad University of Science and Information Technology, Peshawar.