DIAGNOSIS OF PCOS USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

Shanmuhapriyaa.R
Priya.K
Preethi.Jai
Sangeetha.D

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

MARCH & 2020

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"DIAGNOSIS OF PCOS USING MACHINE LEARNING"** is the bonafide work of "**SHANMUHAPRIYAA. R, PRIYA. K, PREETHI. JAI, SANGEETHA. D"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
DR.D SHANTHI	DR.D SHANTHI
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	Professor and Head
Computer science and Engineering	Computer Science and Engineering
PSNA College of Engineering and Technology	PSNA College of Engineering and Technology
Dindigul-624622	Dindigul-624622

Acknowledgement

With warm hearts and immense pleasure, I thank the almighty for his grace and blessings which drove us to the successful completion of the project. I would like to express our gratitude towards our parents for their kind cooperation and encouragement which help me in completion of this project.

We take the opportunity to express our sincere thanks to the respected chairperson **Tmt K. DHANALAKSHMI AMMAL,** who is the guiding light for all the activities in our college. I would like to express deep gratitude to our Pro-chairman **Rtn. Thiru R.S.K RAGURAM, D.A.E., M.COM.,** and Vice-chairman **Thiru R.S.K SUKUMARAN, B.A** for their continuous support towards the development of the students.

We would like to thank our principal **Dr. D. VASUDEVAN M.E, Ph.D.,** for being a beacon in guiding every one of us and infusing us with the strength and enthusiasm to work over successfully.

We express our sincere thanks and heartfelt gratitude to **Dr. D. SHANTHI M.E, Ph.D.** Professor and Head, Department of Computer Science and Engineering for her valuable suggestions and continuous encouragement in the completion of the project work.

This project would not have been possible without the motivation and guidance of **Dr. T. Hemalatha M.E., Ph.D.,** Professor of the Department of Computer Science and Engineering.

ABSTRACT

Polycystic ovary syndrome affects women at large without a known cause and is the order of the day. The main aim is to develop a system based on Machine Learning to periodically monitor women's health and provide apt suggestions. Around 1 in 10 women are affected by Polycystic Ovary Syndrome and more than 1 million cases have been registered in India. To diagnose the presence of this syndrome, PCOS System can be used. This system is an application for women who want to monitor their health. It uses supervised learning algorithms to classify the data modal based on the input. The algorithm is developed based on the symptoms of the patient affected. Users need to answer a set of questions to determine their possibility of getting affected. Based on the information collected from the user, data will be analysed and a suitable suggestion will be provided by the system.

TABLE OF CONTENTS

LIST OF TABLES

LISTS OF FIGURES

LISTS OF ABBREVIATIONS

S.No	Acronym	Expansion
1.	PCOS	PolyCystic Ovary Syndrome
2.	ID3	Iterative Dichotomiser 3
3.	ANN	Artificial Neural Network
4.	IVD	InVitro medical Diagnostics
5.	SDLC	Software Development Life Cycle
6.	IDE	Integrated Development Environment
7.	BP	Blood Pressure
8.	HDL	High-level Data Link

1. INTRODUCTION

1.1 Problem Definition

Polycystic ovarian syndrome is a hormonal disorder common among women of reproductive age. Women associated with this abnormality are highly prone to infertility, anovulation, cardiovascular disease, type 2 diabetes, mellitus, obesity, gynecological cancer etc. Some of the common symptoms include oily skin and darkened acne marks, weight gain, hypertension, mood and anxiety disorder, irregularity in menstrual cycle etc. Nearly 5-10% of reproductive age women are affected by this ailment.

Computer and communication technology can make all the difference in the world for the women with such disorders. Our approach is to develop a system based on Machine Learning to periodically monitor women's health and provide appropriate suggestions. The challenge is to develop a system that should be reliable and user friendly.

To diagnose the presence of PCOS, This system can be used. This system is an application for women who want to monitor their health. It uses supervised learning algorithms to classify the data modal based on the input. The algorithm is developed based on the symptoms of the patient affected. Users need to answer a set of questions to determine their possibility of getting affected. Based on the information collected from the user, data will be analysed and a suitable suggestion will be provided by the system.

1.2 Algorithms and Techniques for classification

The main motive behind opting classification is to accurately predict the target class for each of the objects and the data in entirety.

1.2.1 Decision Tree Induction

The Decision Trees have the sole purpose of generating rules. These rules are nothing but conditional statements or parameters which can be easily interpreted by the users and in turn can be used within a dataset to recognize a set of records. A researcher under the name of J. Ross Quinlan evolved a decision tree algorithm in 1980 called ID3 (Iterative Dichotomiser). Thereafter, he also developed the C4.5 algorithm, which was the apparent inheritor of ID3. Both the algorithms make use of the greedy approach. Here no backtracking procedures have been inculcated as the trees are constructed following a top-down, recursive, divide-and-conquer mannerism.

1.2.2 Naive Bayes Algorithm

Naive Bayes algorithm is a simple classifier based on probability. The Bayesian principle is to allocate cases to the class that has the highest posterior probability. By counting the frequency and the permutation of values, this classifier calculates a set of probabilities from the given data set. An assumption is made, such that all the fields are independent given the value of the class variable. This algorithm is called naïve because the above assumption rarely holds true in real life applications. Nevertheless, this algorithm's performance is good and also it has the ability to learn quickly in supervised classification problems.

1.2.3 Artificial Neural Networks Algorithm

The idea of ANN is based on the belief that working of the human brain can replicate silicon and wires as living neurons and dendrites. The neurons are organized in layers with one layer of input and output layer each and hidden layer(s). An ANN learns the relationships between the input and output data sets. During the training phase, training data is introduced into the neural network. The difference between the actual output values of the network and the training output values is then calculated. This difference is nothing but an error value which is decreased during the training by modifying the weight values of the connections. This is repeated until the error value has reached the predetermined training goal. ANN's have the ability to learn, generalize and most importantly they place no restrictions on the input variables.

1.3 Machine Learning on Diagnostics

Medical diagnostics are a category of medical tests designed to detect infections, conditions and diseases. These medical diagnostics fall under the category of IVD (in vitro medical diagnostics) which are purchased by consumers or used in laboratory settings. Today, AI is playing an integral role in the evolution of the field of medical diagnostics.

Applications of AI in medical diagnostics are in the early adoption phase across multiple specialties with limited data currently available on patient outcomes. These applications have the potential to impact how clinicians and health care systems approach diagnostics and the ability for individuals to understand changes to their health in real-time.

2. LITERATURE SURVEY

2.1 Review of the Literature

The earliest work on PCOS health applications has been started in 2015. The aim of all researchers was to make the system as automated and accurate as possible through various types of inputs. So as to increase its application in the real world.

Humanite Corp (2018) created an application - PCOS Guide, that provides the user to understand about PCOS and provide diet and lifestyle tips. It provides basic information about PCOS and its symptoms to make the users have better insight about their problem.

TrailX (2019) created an application - PCOS Tracker. This application helps the user to keep track of their symptoms that they notice every month at the time of their periods and provides data insights and information about PCOS.

The existing system only provides the information about PCOS and their symptoms. It does not help the users to diagnose or assist them. They provide the tips in common to all users.

Our system helps the user to diagnose and also provide information about PCOS. In addition to that, it also helps the user to contact the doctor to ask queries about their problems and even make an appointment with them. It diagnoses the risk of having PCOS and helps them by providing some useful tips respectively to improve their lifestyle .

3. DEVELOPMENT PROCESS

A software development process is a structure imposed on the development of a software product. The activities concerned with the development of a software are collectively known as Software Development Life Cycle (SDLC). SDLC is any logical process used by a systems analyst to develop an information system, including requirements, validation , training and ownership. An SLDC should result in a high quality system that meets or exceeds customer expectations, reaches completion within time and cost estimates, works effectively and efficiently in the current and planned.

3.1 Requirement Analysis and Specifications

The requirement engineering process consists of feasibility study, requirements elicitations and analysis, requirements specification, requirements validation and requirements management. Requirements elicitation and analysis is an iterative process that can be represented as a spiral of activities, namely requirements classification and organization, requirements negotiation and requirements classification and organization, requirements and requirements documentation.

3.1.1 Input Requirements

The input to the system is by answering the questions that have been asked to the users. The questions are generally of two types: yes/no questions and numbered questions that are based on clinical tests. These questions are formed based on the symptoms that an affected person could have.

3.1.2 Output Requirement

The output will be displayed on the screen. The output of this system falls under three categories: low risk, mid risk, high risk. Along with this, the suggestions and tips to be followed by the users are provided by this system.

3.1.3 Functional Requirements

The answers to the given questions are fed to the classifier algorithm. The algorithm analyses the data using the datasets and provides the appropriate suggestions to the users as an output.

3.1.4 User Requirements

Users need to provide proper details while registering. They should answer all the questions that have been asked for diagnosis. Users need to take clinical tests prior to using this system.

3.2 Resource Requirements

Software requirements is a sub-field of software engineering that deals with the elicitation, analysis, specification and validation of requirements for software. Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirements analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities and defined to a level of detail sufficient for system design.

3.2.1 Hardware

The minimum hardware requirements of this project are listed in Table 1.

Hardware	Requirement
Processor	Intel(R) Core(TM) i5
Memory	8.00 GB
Hard disk capacity	400 GB
Monitor type	64-bit os, x64-based processor

Table 1

3.2.2 Software

The minimum software requirements of this project are listed in table 2.

Software	Requirement
Operating System	Windows 10
Front end	Android Studio
Back end	Anaconda-Spyder
IDE	Json-Server Connectivity (Heroku)

Table 2

3.3 System Description

The PCOS diagnosis system has various phases throughout the completion of process and can be accessed by the user. The user must be signed

up before accessing this system. Login permission must be required for the system to be used.

To diagnose PCOS, users must answer the questions, they are either yes/no questions or numbered questions (Personnel questions and clinical tests). Once they answer the questions, the data are fed into the analyser to analyse the data to provide the result.

The predicted result will be updated to the analysis database and user database for result generation. The results are generated on the basis of the answers that are provided by the users. Other features like emergency contacts are provided to the user so that they can contact and get help from the doctors.

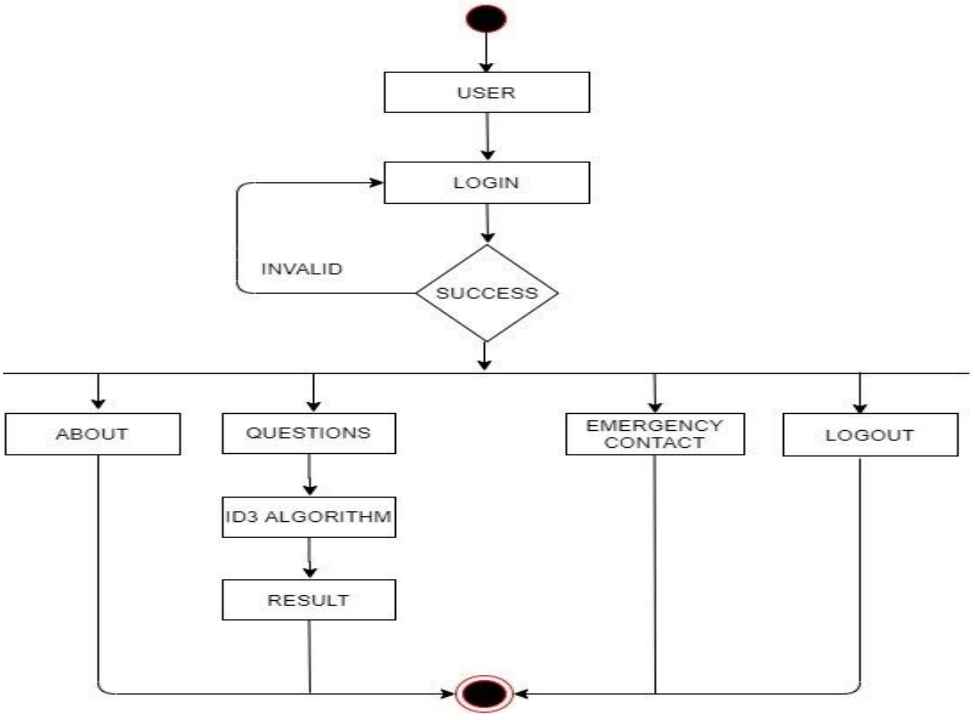


Fig 3.1 System Flow

3.4 Design

Software design is a process of problem -solving and planning for a software solution. After the purpose and specifications of software are

determined, software developers will design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view.

3.4.1 System Architecture

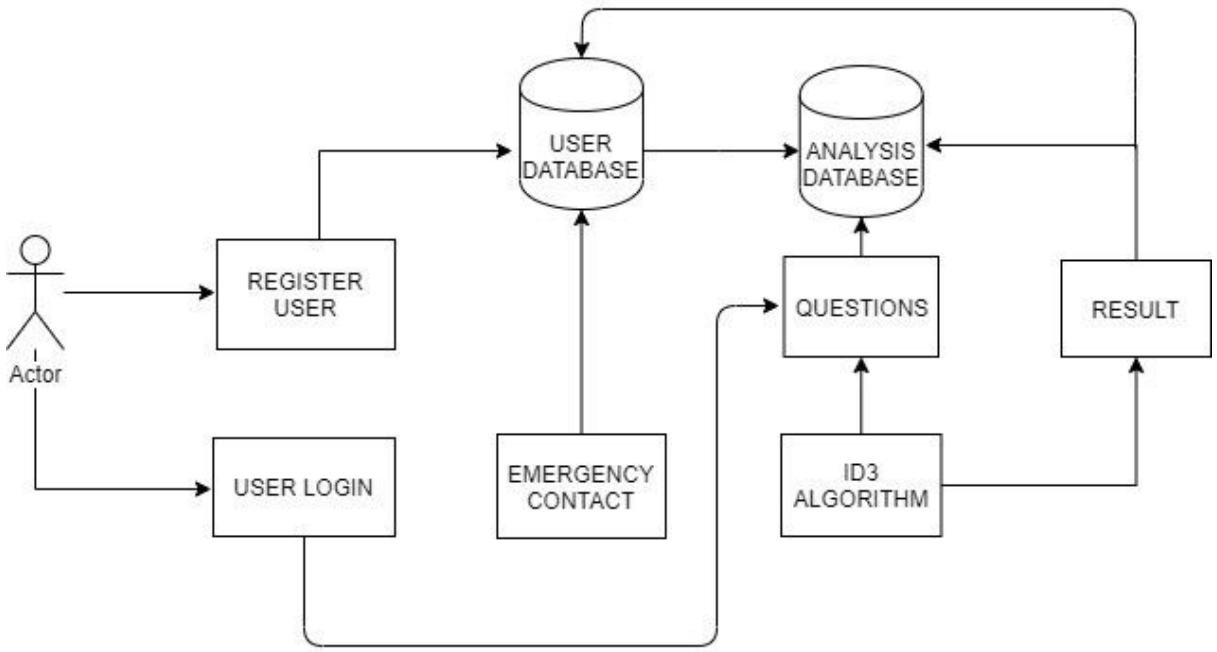


Figure 3.2 Architecture diagram

The architecture of the system is represented in Figure 3.1. This system gets the input from the user by allowing them to answer the set of predefined questions. The data from the user is collected and analysed using the classifier algorithm.

The classifier algorithm already has the dataset, the collected data from the user is analysed using this dataset and provides the result as a suggested output to the user. The output is updated regularly to the user's profile so that the user can monitor their health periodically.

3.4.2 Detailed Design

Our system provides an efficient way of providing appropriate suggestions to the users. The main tasks to be accomplished in the development of the proposed system are designing the modules and database.

3.4.2.1 Module Design

The modules of the proposed system are as follows:

1. Authentication
2. Questionnaires and Datasets
3. ID3 Algorithm

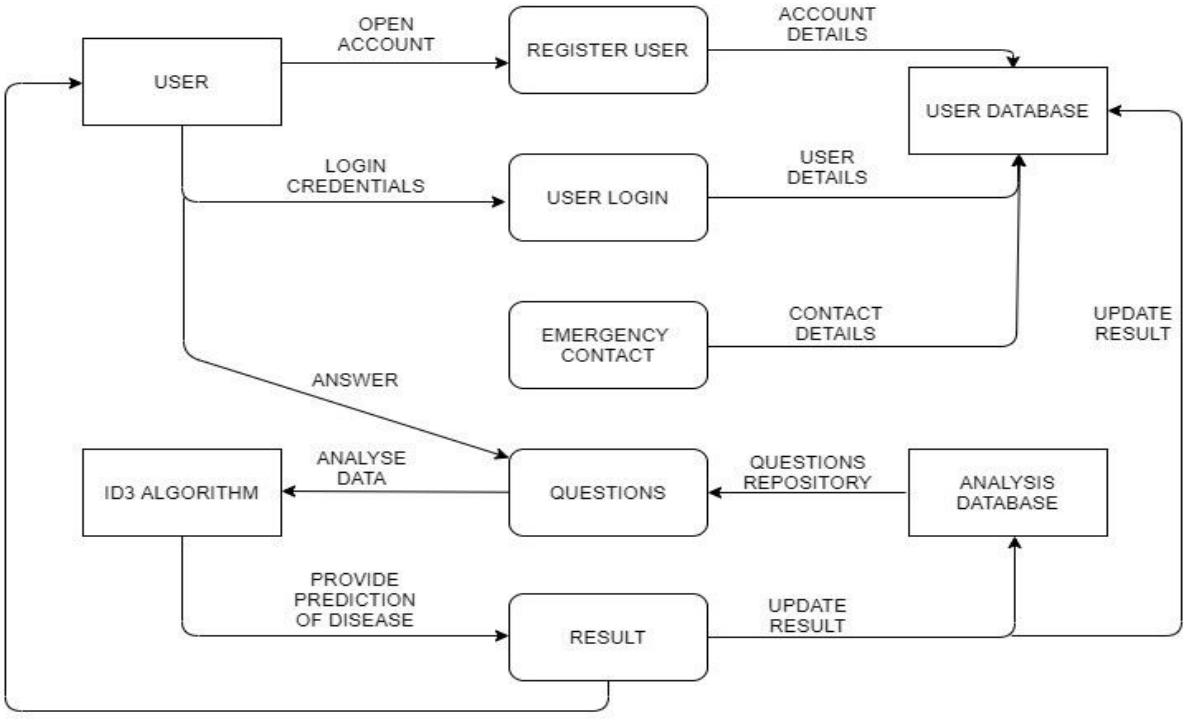


Fig 3.3 High Level Diagram

Authentication:

The system allows the user to register their mobile number and email id, verifies the user credentials that are stored in the user database. The authenticated users are allowed to access the system. The diagnosis process is carried on only when the authentication is validated, if the user is not authenticated the system cannot proceed with the execution. The authenticated users are maintained and allowed to request the system to attain their requirements.

Questionnaires and Dataset:

To diagnose PCOS, Questionnaires are used. Generally, these questions are formed based on the symptoms that an affected person could have. After the authentication process, users are requested to answer the set of questions. Normally, the questions are YES or NO type and some questions are based on clinical tests. Dataset has the record of patients.

ID3 Algorithm:

ID3 Algorithm is used to generate the decision tree from a dataset. It uses a greedy strategy by selecting the locally best attribute to split the dataset on each iteration. The ID3 algorithm is used by training on a data set *S* to produce a decision tree which is stored in memory. At runtime, this decision tree is used to classify new test cases by traversing the decision tree using the features of the datum to arrive at a leaf node. The class of this terminal node is the class the test case is classified as.

3.4.2.3 Database Design

This project has two databases: User Database and Dataset. User database stores details of the users (Table 3). Dataset has the patient's record (Table 4)

.

Table 3: User Database

SNO	FIELDNAME	DATATYPE
1	Name	Varchar
2	User_id	Number
3	Password	Varchar
4.	Email_id	Varchar
5.	Mobile_no	Number
6.	Age	Number

Table 4: Dataset

SNO	ATTRIBUTE	VALUE
1	Crave oily foods and sweets	Yes(y),No(n)
2	Have had continuous weight gain	Yes(y),No(n)
3	Have irregular periods	Yes(y),No(n)
4	Periods last longer than a week	Yes(y),No(n)
5	Feel extremely irritable/stressed/hungry	Yes(y),No(n)
6	Noticed any dark patches on your skin recently	Yes(y),No(n)
7	Have an excess/unusual amount of hair growth on your breasts/upper thighs/stomach	Yes(y),No(n)
8	Have the habit of consuming drugs	Yes(y),No(n)
9	Have thyroid problem	Yes(y),No(n)
10	Take any treatment for getting pregnant	Yes(y),No(n)

11	Age	Number
12	Weight	Number
13	Height	Number
14	Sugar level	Number
15	Androgen level	Number
16	BP level	Number
17	Sleeping hours per day	Number
18	Children do you have	Number
19	Year gap between marriage and first child	Number

3.5 Implementation

Software implementation involves compilation of the designed system. Modular and subsystem programming code will be accomplished during this stage. Unit testing and module testing are done in this stage .This stage is intermingled with the next in their individual modules and will need testing before integration to the main project. Planning in the software life cycle involves setting goals, defining targets, establishing schedules and estimating budget for an entire software project.

3.5.1 Tools and Technologies

Tools:

Spyder: Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language.

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

Technologies:

Python: Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development.

Pandas: A Python package which provides fast, flexible and expensive data structures and data analysis tools designed to make working with various types of data both easy and intuitively.

NumPy: A library for the python programming language , adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate with.

Java: Java is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development.

3.6 Testing

Testing is the process of evaluating the correctness, the quality and the completeness of the system developed. Our system was tested across a variety of applicants. It was found that the system was able to provide appropriate suggestions successfully to all cases. Since the error in the software can be injured at any stage. We had carried out the testing process at different levels during the development. The basic levels of testing are,

- Unit testing
- Integration testing
- Validation testing
- Functional testing
- Structural testing

Unit Testing

Unit testing was used to test individual units in the system and ensure that they operate correctly. Alternate logic analysis and screen validations were tested in this to ensure optimum efficiency in the system. The procedures and functions used and their association with data were tested.

Integration Testing

This testing process focuses on identifying the interfaces between components and their functionality. The bottom up approach was adopted during this testing. Low_level modules are integrated and combined as a cluster before testing. This allows identifying any wrong linkages or parameters passing early in the development process as it just can be passed in the set of data and checked if the result returned is an accepted one.

Validation Testing

Software testing and validation is achieved through a series of black box test that demonstrate conformity with requirements. A test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both, the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met. After each validation test case has been conducted, one of the two possible exists.

Functional Testing

Functional testing, also known as block box or closed box testing, is normally applied to HDL (High-Level Data Link) code that operates concurrently and concentrates on checking the interaction between modules,

blocks or functional boundaries. The objective here is to ensure that correct results are obtained, when inputs are applied, in a predictable manner. Functional testing can therefore be considered as concentrating on checking that the data paths operate correctly.

Structural Testing

Structural testing is known as white box or open box testing is normally applied to sequential HDL (High-Level Data Link) code and concentrates on checking that all executable statements within each module have been exercised and the corresponding branches and paths through that module have been covered. If there is a section of HDL code that has never been exercised then there is a high possibility that it could contain an error that will remain undetected.

4. FUTURE ENHANCEMENT

This system can be improvised by letting the user book their appointment directly through our application via an email sent to the doctor by a single click in app. The accuracy can be enhanced for about greater than 90% for better prediction. High class design along with better UI can be facilitated. Profile page will be made more user-friendly. Any other enhancements for better implementation will be done.

BENEFITS

- Women can check their risk of being affected by PCOS.
- They can obtain guidelines regarding lifestyle, treatment at the right time for their aid.

- It is simple and easy to use(User friendly) by common people.
- Accuracy of about 80% in our diagnosis system which helps in identifying risk at a cheaper cost.
- Users have their own Login data and profile page that stores their information.

Limitations of the system

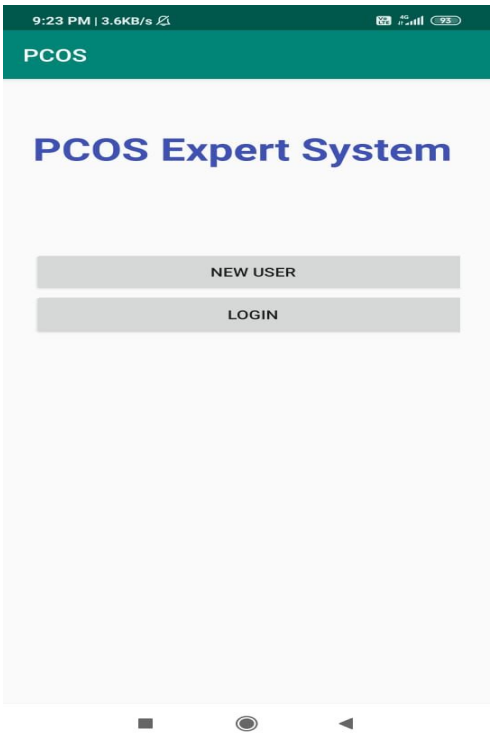
- Cannot book appointments directly. Only contact details would be provided.
- The accuracy of the prediction can be further enhanced to provide better accuracy.
- UI design can be made more user-friendly.
- Profile page can be enhanced more.

5. CONCLUSION

The objective of this project is to provide a system that helps the users to diagnose PCOS and provide suggestions and tips to the user. This project is designed as a mobile application and runs well on android os. It incorporates all requirements and has been developed as versatile and user friendly as possible. It has been designed using advanced features of new technologies. The modular design and constructed system is user oriented in which users can easily understand and use this application.

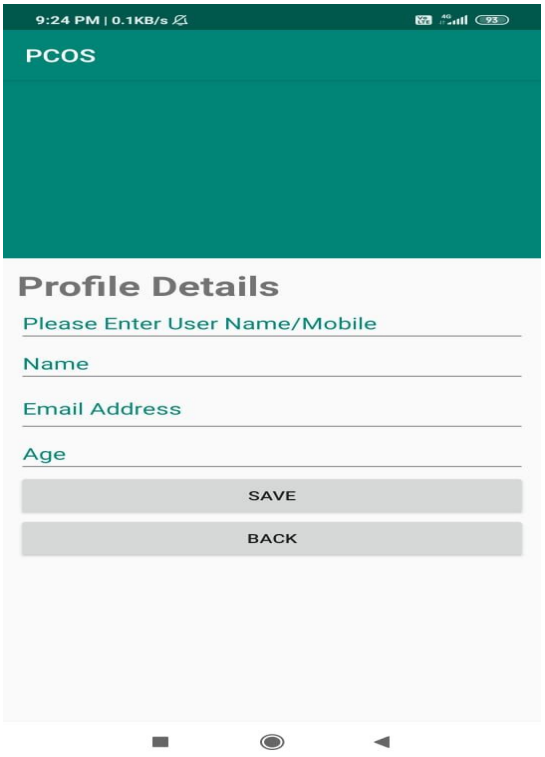
APPENDIX - SCREENSHOTS

LOGIN PAGE 1



LOGIN PAGE 2

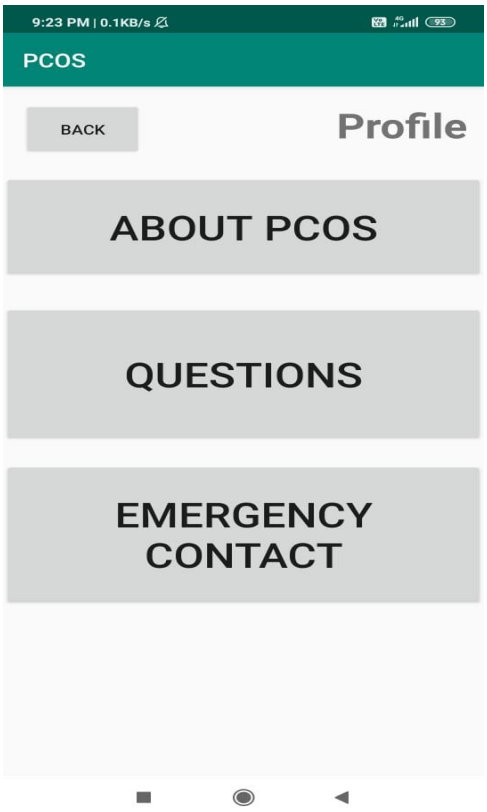
LOGIN PAGE 3


9:24 PM | 0.1KB/s
PCOS
Profile Details
Please Enter User Name/Mobile
Name
Email Address
Age
SAVE
BACK

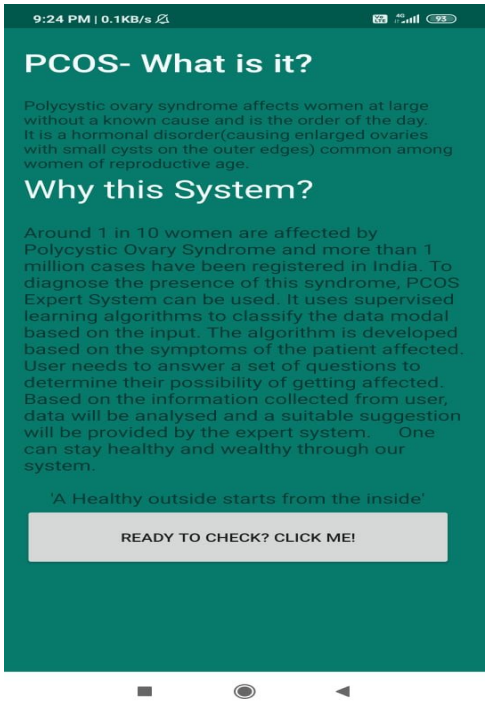
LOGIN PAGE 4


9:23 PM | 0.1KB/s
PCOS
8056214001
LOGIN
NO ID? SIGN UP

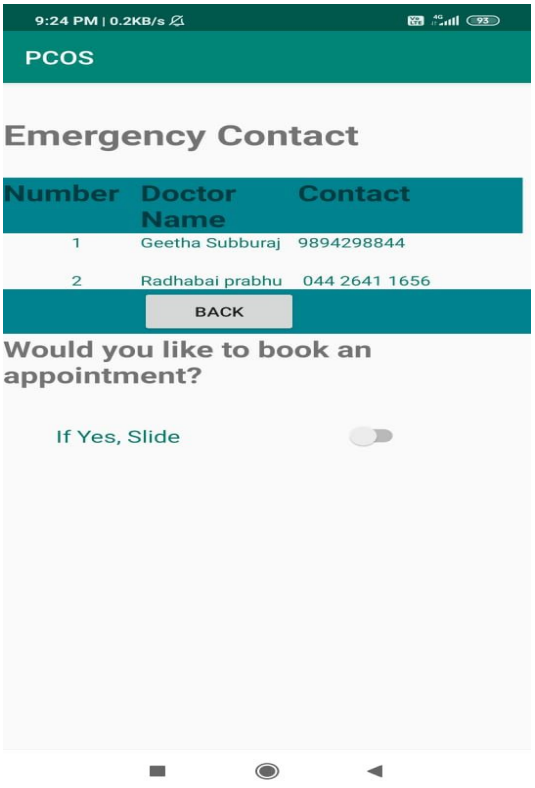
MENU PAGE



ABOUT PAGE

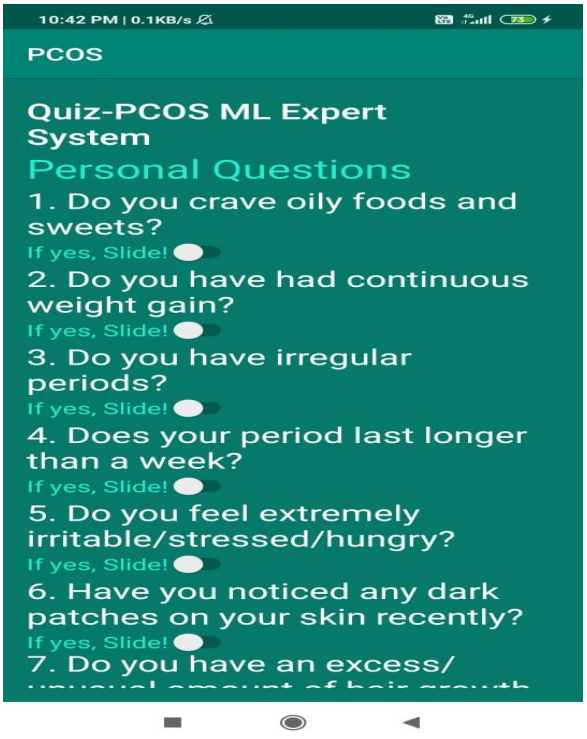


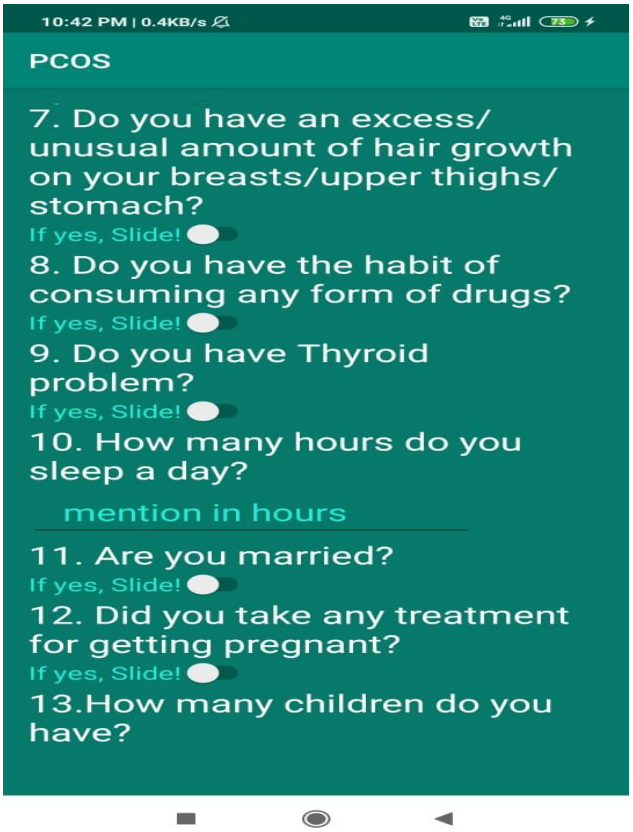
EMERGENCY CONTACT PAGE

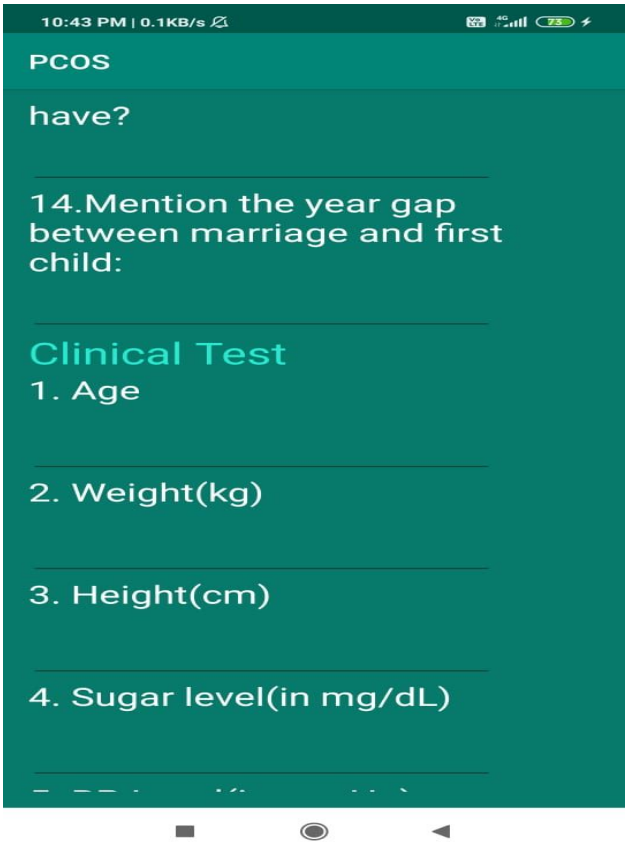


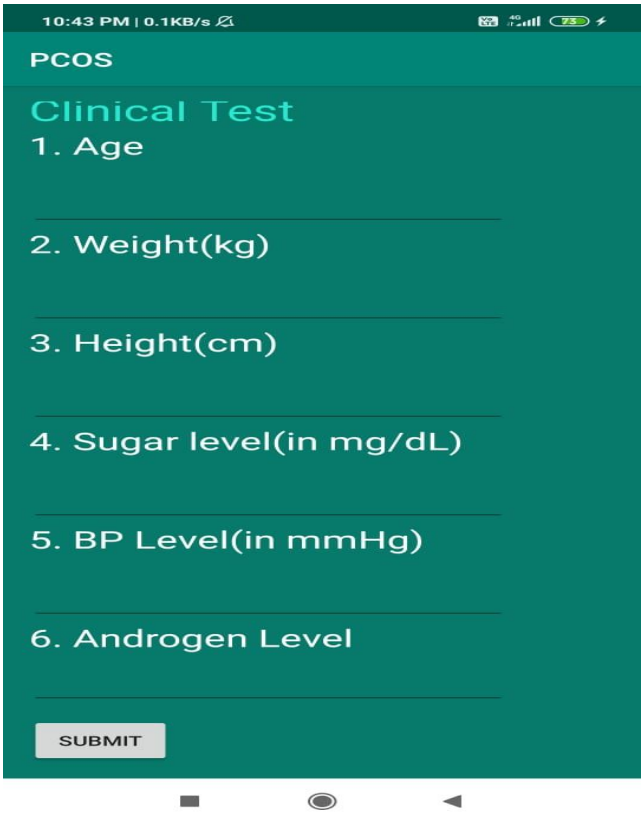
QUESTIONS:

Single Page(Scroll View)

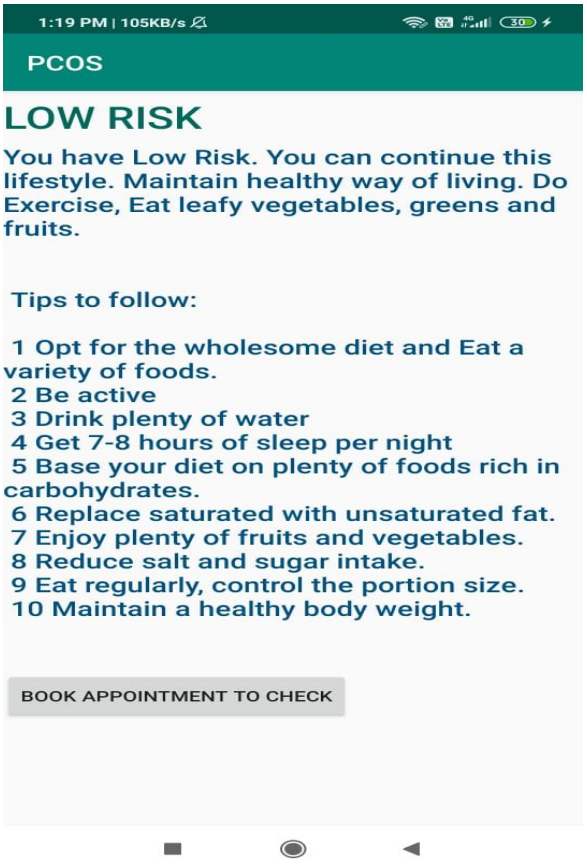

10:42 PM | 0.1KB/s
PCOS
Quiz-PCOS ML Expert System
Personal Questions
1. Do you crave oily foods and sweets?
If yes, Slide!
2. Do you have had continuous weight gain?
If yes, Slide!
3. Do you have irregular periods?
If yes, Slide!
4. Does your period last longer than a week?
If yes, Slide!
5. Do you feel extremely irritable/stressed/hungry?
If yes, Slide!
6. Have you noticed any dark patches on your skin recently?
If yes, Slide!
7. Do you have an excess/


10:42 PM | 0.4KB/s
PCOS
7. Do you have an excess/ unusual amount of hair growth on your breasts/upper thighs/ stomach?
If yes, Slide!
8. Do you have the habit of consuming any form of drugs?
If yes, Slide!
9. Do you have Thyroid problem?
If yes, Slide!
10. How many hours do you sleep a day?
mention in hours
11. Are you married?
If yes, Slide!
12. Did you take any treatment for getting pregnant?
If yes, Slide!
13.How many children do you have?

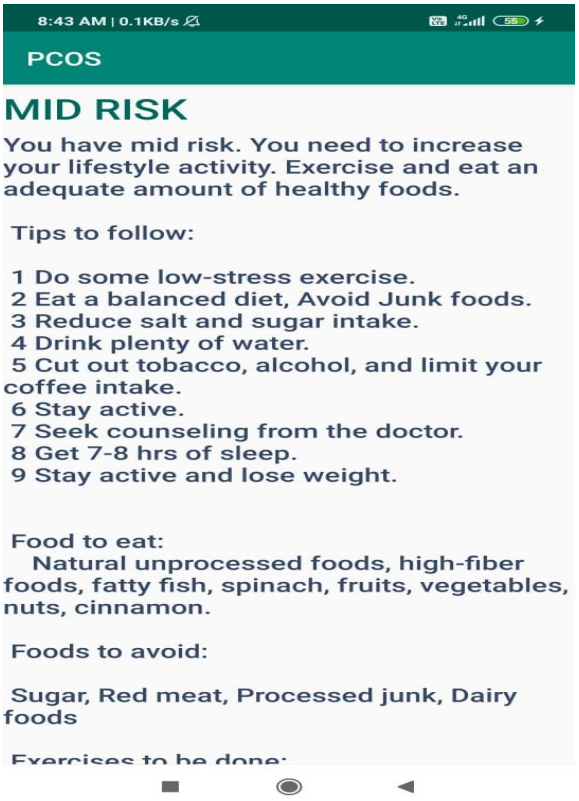

10:43 PM | 0.1KB/s
PCOS
have?
14.Mention the year gap between marriage and first child:
Clinical Test
1. Age
2. Weight(kg)
3. Height(cm)
4. Sugar level(in mg/dL)

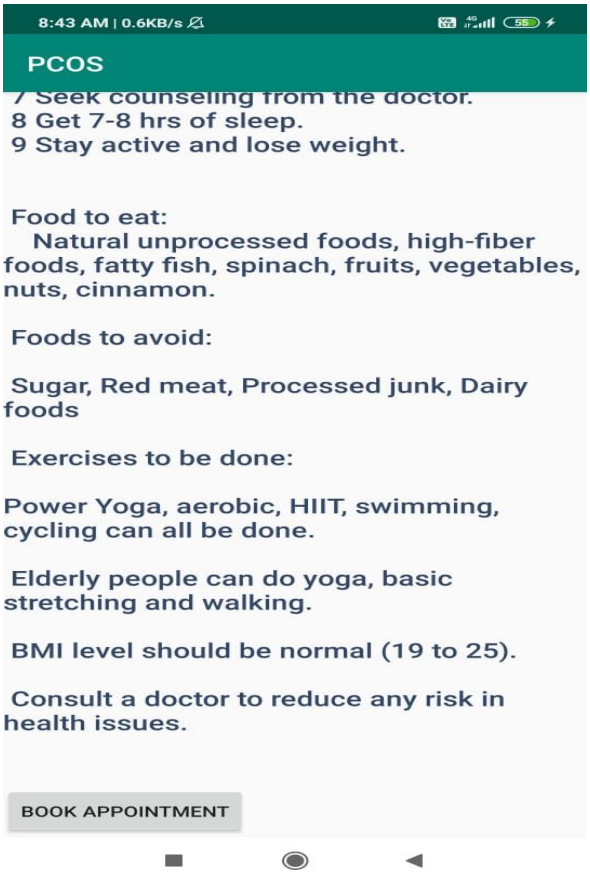

10:43 PM | 0.1KB/s
PCOS
Clinical Test
1. Age
2. Weight(kg)
3. Height(cm)
4. Sugar level(in mg/dL)
5. BP Level(in mmHg)
6. Androgen Level
SUBMIT

OUTPUT SCREEN 1 - LOW RISK

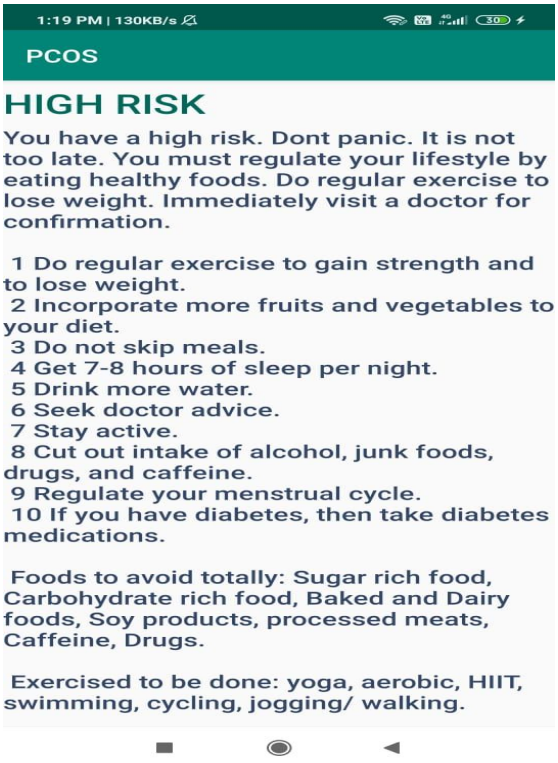


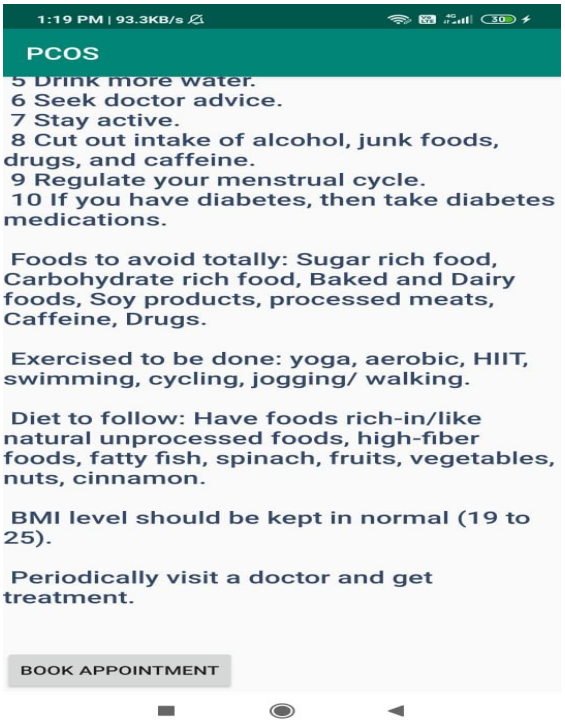
OUTPUT SCREEN 2 - MID RISK





OUTPUT SCREEN 3 - HIGH RISK





Code:

FRONT END:

Login Connectivity:

```
package com.example.pcos;


import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;


public class LoginActivity extends MainActivity {
```

```
    Button btn_lregister, btn_llogin;
    EditText et_lusername, et_lpassword;
 String TAG="login";
    DatabaseHelper databaseHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        databaseHelper = new DatabaseHelper(this);

        et_lusername = findViewById(R.id.input_phone);
        et_lpassword = findViewById(R.id.input_password);

        btn_llogin = findViewById(R.id.login);
        btn_lregister = findViewById(R.id.register);

        btn_lregister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(LoginActivity.this, UserActivity.class);
                startActivity(intent);
            }
        });

        btn_llogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
               String username = et_lusername.getText().toString();
                String password = et_lpassword.getText().toString();

                Boolean checklogin = databaseHelper.CheckLogin(username,
password);
                Log.e(TAG, "onClick: "+checklogin);
```

```
            if(checklogin==true) {
                Intent intent = new Intent(LoginActivity.this, MenuActivity.class);
                startActivity(intent);
             }
          }
       });
   }}
```

DatabaseHelper.java

```
package com.example.pcos;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {
   public static final String DATABASE_NAME = "login.db";



   public DatabaseHelper(Context context) {
      super(context, DATABASE_NAME, null, 1);
   }

   @Override
   public void onCreate(SQLiteDatabase db) {
      db.execSQL("CREATE TABLE user(ID INTEGER primary key,
username text, password text)");
   }

   @Override
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
```

```
        db.execSQL("DROP TABLE IF EXISTS user");
    }

    public boolean Insert(String username, String password) {
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("username", username);
        contentValues.put("password", password);
        long result = sqLiteDatabase.insert("user", null, contentValues);
        return result != -1;
    }

    public Boolean CheckUsername(String username) {
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
        Cursor cursor = sqLiteDatabase.rawQuery("SELECT * FROM user
WHERE username=?", new String[]{username});
        return cursor.getCount() <= 0;
    }

    public Boolean CheckLogin(String username, String password) {
        SQLiteDatabase sqLiteDatabase = this.getReadableDatabase();
        Cursor cursor = sqLiteDatabase.rawQuery("SELECT * FROM user
WHERE username=? AND password=?", new String[]{username, password});
        return cursor.getCount() > 0;
    }
}
```

ProfileActivity.java:

```
package com.example.pcos;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

public class ProfileActivity extends AppCompatActivity {
    EditText editMobile;
     EditText editName;
    EditText editEmail;
     EditText editAge;

    Button save;

    String profileText1;
    String profileText2;
     String profileText3;
   String profileText4;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);
        save = findViewById(R.id.save);
        editMobile = findViewById(R.id.mobile);
        editAge=findViewById(R.id.age);
        editEmail=findViewById(R.id.email);
        editName=findViewById(R.id.name);
        /*
        editAge=(EditText)findViewById(R.id.age);
        editEmail=(EditText)findViewById(R.id.email);
        editName=(EditText)findViewById(R.id.name);

        profileText1=editMobile.getText().toString();
        profileText2=editAge.getText().toString();
        profileText3=editEmail.getText().toString();
        profileText4=editName.getText().toString();
```

```
        if (savedInstanceState != null) {
            editMobile.setText(savedInstanceState.getString("text"));
            editAge.setText(savedInstanceState.getString(profileText2));
            //setting the saved value to the TextView
        }
/*
        editMobile.setText(profileText1);
        editAge.setText(profileText2);
        editEmail.setText(profileText3);
        editName.setText(profileText4);

 */

        save.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
           /* profileText2 = editAge.getText().toString();
            editAge.setText(profileText2);
            profileText1=editMobile.getText().toString();
            editMobile.setText(profileText1);
            profileText3=editEmail.getText().toString();
            editEmail.setText(profileText3);
            profileText4=editName.getText().toString();
            editName.setText(profileText4);
            */
                Intent i = new Intent(ProfileActivity.this, SaveActivity.class);
                profileText1 = editMobile.getText().toString();
                profileText2=editName.getText().toString();
                profileText3=editEmail.getText().toString();

                profileText4=editAge.getText().toString();

                i.putExtra("value", profileText1);
```

```
            i.putExtra("value1",profileText2);
            i.putExtra("value2",profileText3);
            i.putExtra("value3",profileText4);
            startActivity(i);
            finish();
        }
    });
}
// });


public void profile(View v) {
    startActivity(new Intent(ProfileActivity.this, MenuActivity.class));
}


}
```

BACKEND:

Backend Connectivity:

Train.py:

```
#data preprocessing

#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix,accuracy_score
from sklearn.externals import joblib
```

```
def train_model():
    #importing dataset
    dataset =pd.read_csv(r"C:\Users\priya\Project\model\PcodMini.csv")
    X = dataset.iloc[:, :-1].values
    Y = dataset.iloc[:, 19].values

    #encoding categorical data
    labelencoder_y=LabelEncoder()
    Y=labelencoder_y.fit_transform(Y)

    #splitting training and test set

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state
=0)



    #fitting the classifier to the training set
    classifier = DecisionTreeClassifier(criterion='entropy',random_state=0)
    classifier.fit(X_train, Y_train)

    #predicting the test set results
    Y_pred=classifier.predict(X_test)

    # making the confusion matrix and calculating accuracy
    cm=confusion_matrix(Y_test,Y_pred)
    accuracy = accuracy_score(Y_test, Y_pred)

    #dumping the classifier into model
    joblib.dump(classifier, 'pcod-model.model')
    print('Model Training Finished.\n\tAccuracy obtained: {}'.format(accuracy))
```

App.py:

```
import os
```

```
from flask import Flask, jsonify, request
from flask_restful import Api, Resource
from Train import train_model
from sklearn.externals import joblib

app = Flask(__name__)
api = Api(app)
if not os.path.isfile('pcod-model.model'):
    train_model()

model = joblib.load('pcod-model.model')

class MakePrediction(Resource):
    @staticmethod
    def post():
        posted_data = request.get_json()
        age = posted_data['age']
        weight = posted_data['weight']
        height = posted_data['height']
        sugar_Level = posted_data['sugar_Level']
        bp_Level = posted_data['bp_Level']
        androgen_Level = posted_data['androgen_Level']
        sleep = posted_data['sleep']
        child_Count = posted_data['child_Count']
        gap_Mrg_Child = posted_data['gap_Mrg_Child']
        periods_long_week = posted_data['periods_long_week']
        irregular_periods = posted_data['irregular_periods']
        fast_food = posted_data['fast_food']
        loose_Weight = posted_data['loose_Weight']
        Hair_Growth = posted_data['Hair_Growth']
        dark_Patches = posted_data['dark_Patches']
        stress = posted_data['stress']
        any_Drugs = posted_data['any_Drugs']
        thyroid_problem = posted_data['thyroid_problem']
        treatment_Taken = posted_data['treatment_Taken']
```

```
        prediction =
model.predict([[age,weight,height,sugar_Level,bp_Level,androgen_Level,sleep,
child_Count,gap_Mrg_Child,periods_long_week,irregular_periods,fast_food,lo
ose_Weight,Hair_Growth,dark_Patches,stress,any_Drugs,thyroid_problem,treat
ment_Taken]])[0]
        if prediction == 0:
            predicted_class = 'high_risk'
        elif prediction == 1:
            predicted_class = 'mid_risk'
        else:
            predicted_class = 'low_risk'

        return jsonify({
            'Prediction': predicted_class
        })

api.add_resource(MakePrediction, '/predict')
if __name__ == '__main__':
    app.run(debug=True)
```

Requirements for Python:

```
aniso8601==8.0.0
astroid==2.1.0
backcall==0.1.0
binarytree==4.0.0
bleach==3.1.0
boto==2.49.0
boto3==1.8.1
botocore==1.11.1
bz2file==0.98
certifi==2018.8.24
chardet==3.0.4
```

click==7.1.1
colorama==0.4.1
cycler==0.10.0
decorator==4.3.2
defusedxml==0.5.0
docutils==0.14
entrypoints==0.3
Flask==1.1.1
Flask-RESTful==0.3.8
gensim==3.5.0
gunicorn==19.9.0
idna==2.7
ipykernel==5.1.0
ipython==7.3.0
ipython-genutils==0.2.0
ipywidgets==7.4.2
isort==4.3.4
itsdangerous==1.1.0
jedi==0.13.2
Jinja2==2.11.1
jmespath==0.9.3
jsonschema==2.6.0
jupyter==1.0.0
jupyter-client==5.2.4
jupyter-console==6.0.0
jupyter-core==4.4.0
kiwisolver==1.0.1
lazy-object-proxy==1.3.1
MarkupSafe==1.1.0
matplotlib==3.0.2
mccabe==0.6.1
mistune==0.8.4
nbconvert==5.4.1
nbformat==4.4.0
notebook==5.7.4

numpy==1.15.1
panda==0.3.1
pandas==0.24.0
pandocfilters==1.4.2
parso==0.3.4
pickleshare==0.7.5
prometheus-client==0.6.0
prompt-toolkit==2.0.9
Pygments==2.3.1
pylint==2.2.2
pyparsing==2.3.1
python-dateutil==2.7.3
pytz==2018.9
pywinpty==0.5.5
pyzmq==18.0.0
qtconsole==4.4.3
requests==2.19.1
s3transfer==0.1.13
scikit-learn==0.19.2
scipy==1.1.0
Send2Trash==1.5.0
six==1.12.0
sklearn==0.0
smart-open==1.6.0
terminado==0.8.1
testpath==0.4.2
tornado==5.1.1
traitlets==4.3.2
typed-ast==1.1.1
urllib3==1.23
wcwidth==0.1.7
webencodings==0.5.1
Werkzeug==1.0.0
widgetsnbextension==3.4.2
wrapt==1.10.11

Intermediate Connection(Heroku server with Android Code):

```
package com.example.pcos;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.Switch;

import androidx.appcompat.app.AppCompatActivity;

import org.apache.commons.io.IOUtils;
import org.json.JSONObject;

import java.io.BufferedInputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;


public class Questionactivity extends AppCompatActivity {
  public static int is1 = 0, is2 = 0, is3 = 0, is4 = 0, is5 = 0, is6 = 0, is7 = 0, is8 =
0, is9 = 0, is11 = 0, is12 = 0;
  public static Switch s1, s2, s3, s4, s5, s6, s7, s8, s9, s11, s12;
  public static String se10, se13, se14, sce1, sce2, sce3, sce4, sce5, sce6;
  public static EditText e10, e13, e14, ec1, ec2, ec3, ec4, ec5, ec6;
  public static String prediction;
  Button submit;
  // public static String prediction, result, json;
```

```
  public static String query_url =
"https://pcod-disease-prediction.herokuapp.com/predict";

  @Override


  public void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.all_page);
     s1 = findViewById(R.id.spcoq1);
     s2 = findViewById(R.id.spcoq2);
     s3 = findViewById(R.id.spcoq3);
     s4 = findViewById(R.id.spcoq4);
     s5 = findViewById(R.id.spcoq5);
     s6 = findViewById(R.id.spcoq6);
     s7 = findViewById(R.id.spcoq7);
     s8 = findViewById(R.id.spcoq8);
     s9 = findViewById(R.id.spcoq9);
     s11 = findViewById(R.id.spcoq11);
     s12 = findViewById(R.id.spcoq12);

     e10 = findViewById(R.id.epcoq10);
     e13 = findViewById(R.id.epcoq13);
     e14 = findViewById(R.id.epcoq14);

     ec1 = findViewById(R.id.cepcoq1);
     ec2 = findViewById(R.id.cepcoq2);
     ec3 = findViewById(R.id.cepcoq3);
     ec4 = findViewById(R.id.cepcoq4);
     ec5 = findViewById(R.id.cepcoq5);
     ec6 = findViewById(R.id.cepcoq6);
     {
        s1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
```

```
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is1 = 1;
                    // The toggle is enabled
                }
                else{
                    is1 = 0;
                }
            }
        });
        s2.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is2 = 1;
                    // The toggle is enabled
                }
                else{
                    is2 = 0;
                }
            }
        });
        s3.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is3 = 1;
                    // The toggle is enabled
                }
                else{
                    is3 = 0;
                }
```

```
            }
        });
        s4.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is4 = 1;
                    // The toggle is enabled
                }
                else{
                    is4 = 0;
                }
            }
        });
        s5.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is5 = 1;
                    // The toggle is enabled
                }
                else{
                    is5 = 0;
                }
            }
        });
        s6.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is6 = 1;
                    // The toggle is enabled
```

```
            }
            else{
                is6 = 0;
            }
        }
    });
    s7.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
            if (isChecked) {
                is7 = 1;
                // The toggle is enabled
            }
            else{
                is7 = 0;
            }
        }
    });
    s8.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
            if (isChecked) {
                is8 = 1;
                // The toggle is enabled
            }
            else{
                is8 = 0;
            }
        }
    });
    s9.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
```

```
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is9 = 1;
                    // The toggle is enabled
                }
                else{
                    is9 = 0;
                }
            }
        });
        s11.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is11 = 1;
                    // The toggle is enabled
                }
                else{
                    is11 = 0;
                }
            }
        });
        s12.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
                    is12 = 1;
                    // The toggle is enabled
                }
                else{
                    is12 = 0;
                }
```

```
            }
        });

    public void onClick(View submit)  {
        se10 = e10.getText().toString();
        se13 = e13.getText().toString();
        se14 = e14.getText().toString();
        sce1 = ec1.getText().toString();
        sce2 = ec2.getText().toString();
        sce3 = ec3.getText().toString();
        sce4 = ec4.getText().toString();
        sce5 = ec5.getText().toString();
        sce6 = ec6.getText().toString();
        new MyTask().execute();

    }

    private class MyTask extends AsyncTask<Void,Void,Void> {

        @Override
        protected Void doInBackground(Void... voids) {
            String query_url =
"https://pcod-disease-prediction.herokuapp.com/predict";
            try {
                URL url = new URL(query_url);
                HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
                conn.setConnectTimeout(5000);
                conn.setRequestProperty("Content-Type", "application/json");
                conn.setRequestProperty("Accept","*/*");
                conn.setRequestProperty("User-Agent","Mozilla/5.0 ( compatible ) ");
                conn.setDoOutput(true);
                conn.setDoInput(true);
                conn.setRequestMethod("POST");
                JSONObject json1 = new JSONObject();
```

```
            json1.put("age",sce1);
            json1.put("weight",sce2);
            json1.put("height",sce3);
            json1.put("sugar_Level",sce4);
            json1.put("bp_Level",sce5);
            json1.put("androgen_Level", sce6);
            json1.put("sleep", se10);
            json1.put("child_Count", se13);
            json1.put("gap_Mrg_Child", se14);
            json1.put("periods_long_week", is4);
            json1.put("irregular_periods", is3);
            json1.put("fast_food", is1);
            json1.put("loose_Weight", is2);
            json1.put("Hair_Growth", is7);
            json1.put("dark_Patches", is6);
            json1.put("stress", is5);
            json1.put("any_Drugs", is8);
            json1.put("thyroid_problem", is9);
            json1.put("treatment_Taken", is12);
            String json = json1.toString();
            OutputStream os = conn.getOutputStream();
            os.write(json.getBytes());
            // read the response
            int code =conn.getResponseCode();
            InputStream in = new BufferedInputStream(conn.getInputStream());
            String result = IOUtils.toString(in);
            System.out.println(result);
            System.out.println("result after Reading JSON Response");
            JSONObject myResponse = new JSONObject(result);
            prediction=myResponse.getString("Prediction");

            os.close();
            in.close();
            conn.disconnect();
        } catch (Exception e) {
```

```
            System.out.println(e);
        }
        return null;
    }

    protected void onPostExecute(Void aVoid) {
        if (prediction.equals("low_risk")) {
            startActivity(new Intent(Questionactivity.this, LowRisk.class));
        } else if (prediction.equals("mid_risk")) {
            startActivity(new Intent(Questionactivity.this, Mid_Risk.class));
        } else if (prediction.equals("high_risk")) {
            startActivity(new Intent(Questionactivity.this, HighRisk.class));
        }
        else {
            //Add a new class for no risk.
            startActivity(new Intent(Questionactivity.this, AboutActivity.class));
        }
    }
```

REFERENCES:

1. Fisher, D.H.(1987). Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2, 139-172.
2. Quinlan, J.R (1987). Induction of decision lists. Machine Learning, 1, 81-106.
3. Rivest, R.L. (1987). Learning decision lists. Machine Learning, 2, 229-246.
4. A. Saravanan, S. Sathiamoorthy. Detection of Polycystic Ovarian Syndrome (Asian journal of Engineering and applied technology).
5. Polycystic Ovary Syndrome: An Updated Overview. Samer El Hayek, Lynn Bitar and Georges Daoud.
6. Supervised Machine Learning: A Review of Classification Techniques. S.B. Kotsians.
7. A Patient's Guide To PCOS. Walter Futterweit, George Ryan.
8. Internet: Pages like Google, Wikipedia, Heroku Guide etc.