INDEX

1. ABSTRACT

The "Health Cart" application project is a comprehensive solution designed to assist users in effectively managing their healthcare needs. By leveraging various features, the application aims to provide a seamless experience for users in accessing essential health-related information and services. One of the key functionalities of the application is its ability to offer access to comprehensive checkup details. Users can retrieve detailed information about various medical checkup packages available. Additionally, the application facilitates easy access to information about doctors categorized by their specific fields. This feature enables users to find doctors who specialize in their specific medical needs, ensuring they receive the most appropriate care. By integrating a booking system, users can conveniently schedule appointments based on their own preferences. To ensure efficient management of healthcare-related activities, the application incorporates a cart feature. Through this feature, users can keep track of their purchased checkup packages and booked appointments in a centralized location. This enables users to have a clear overview of their upcoming medical commitments and previously availed services. Recognizing the importance of data security, the project includes a login page and a dedicated login database. This ensures that user information is protected and accessed securely.

1. INTRODUCTION

One of the key components of the "Health Cart" Android app is the user login feature, which ensures personalized access to a range of health services. By creating an account, users can securely store their health-related information, medical history, and preferences, enabling them to have a tailored and seamless experience within the app.

A significant aspect of the app is the lab test module, which empowers users to conveniently book health checkup packages. This feature eliminates the hassle of physical visits to diagnostic centers by allowing users to browse through a variety of test packages, select the one that suits their needs, and schedule appointments with partner laboratories directly from the app.

Another essential feature is the "Find Doctor" module, which simplifies the process of scheduling appointments with healthcare professionals. Users can search for doctors based on specialties, location,making it easy to find the right healthcare provider for their specific needs. The module also offers real-time appointment availability, allowing users to book appointments instantly without the need for phone calls or manual coordination.

To facilitate efficient management of bookings, the app includes an "Order Details" module. This module serves as a centralized hub for users to view their appointments and lab test bookings.

The "Health Article" module features a collection of informative articles and images on a wide range of health subjects. The module aims to empower users with valuable health-related information, promoting proactive engagement in maintaining a healthy lifestyle.

2. SYSTEM ANALYSIS

3.1Existing System

Existing System Challenges:

- **Outdated Information:**

Many existing applications that provide information about doctors and hospitals suffer from the problem of outdated information. Users often encounter difficulties when the information they access has not been updated for a long time. This can lead to incorrect or incomplete details, such as outdated contact information or changes in doctors' availability.

- **Fragmented Booking Experience:**

Several applications exist that allow users to book appointments, but they are often limited to specific doctors or hospitals. Users need to install separate apps for each doctor or hospital they wish to book an appointment with, resulting in a fragmented and cumbersome booking experience.

- **Lack of Comprehensive Doctor Listings:**

The existing system lacks a comprehensive application that provides a consolidated list of doctors based on their medical specialty. Users may have to rely on multiple sources or platforms to gather information about doctors in different specialties, making the search process time-consuming and inefficient.

Addressing these limitations can lead to the development of an improved system that offers a more efficient and user-friendly experience for users seeking doctor information and appointment booking services.

3.2Proposed System

Proposed System Features:

- **Updated and Comprehensive Information:**

The proposed system will address the issue of outdated information by ensuring that the data about doctors, hospitals, and their availability is regularly updated. Users can rely on accurate and current information regarding doctors' locations, contact details, and specialties.

- **Consolidated Doctor Listings:**

The proposed system will provide a comprehensive list of doctors based on their medical specialty. Users can easily search and browse through the list to find doctors in their desired field, eliminating the need to rely on multiple sources or applications.

- **Appointment Booking for Specific Doctors:**

The proposed system will allow users to book appointments with specific doctors from the list based on their specialty. Users can select a doctor of their choice and choose an available time slot for the appointment, ensuring a convenient and personalized booking experience.

- **User-Friendly Interface:**

The proposed system will prioritize a user-friendly interface that is easy to navigate and provides a seamless experience. Users will be able to access the desired information, book appointments, browse articles, and interact with the application effortlessly.

4. SYSTEM SPECIFICATION

4.1 Hardware Specification

1. for Computer System/Laptop:

- 8 GB of RAM or more.
- 8 GB of available disk space minimum.
- Minimum screen resolution 1280 x 720.

2. for Mobile phone:

- Android mobile with Lollipop (5.0) and above versions
- Minimum of 2 GB of RAM or More.
- Minimum of 512MB of available storage.

4.2 Software Specifications

Frontend:

- XML

Backend:

- Java
- SQLite

Basic Requirements:

- IDE
- Android SDK
- Android Emulator

5. SYSTEM DESIGN

5.1 Database Design:

Users Table:

Sl.no	Name	Type	Constraints
1	Username	Text	Primary key, Unique
2	Email	Text	Not Null
3	Password	Text	Not Null

Orderplaced Table:

Sl.no	Name	Type	Constraints
1	Username	Text	Foreign key References users(Username)
2	Fullname	Text	Not Null
3	Address	Text	Not Null
4	Contactno	Text	Not Null
5	Pincode	Int	Not Null
6	Date	Text	Not Null
7	Time	Text	Not Null
8	Amount	Float	Not Null
9	Otype	Float	Not Null

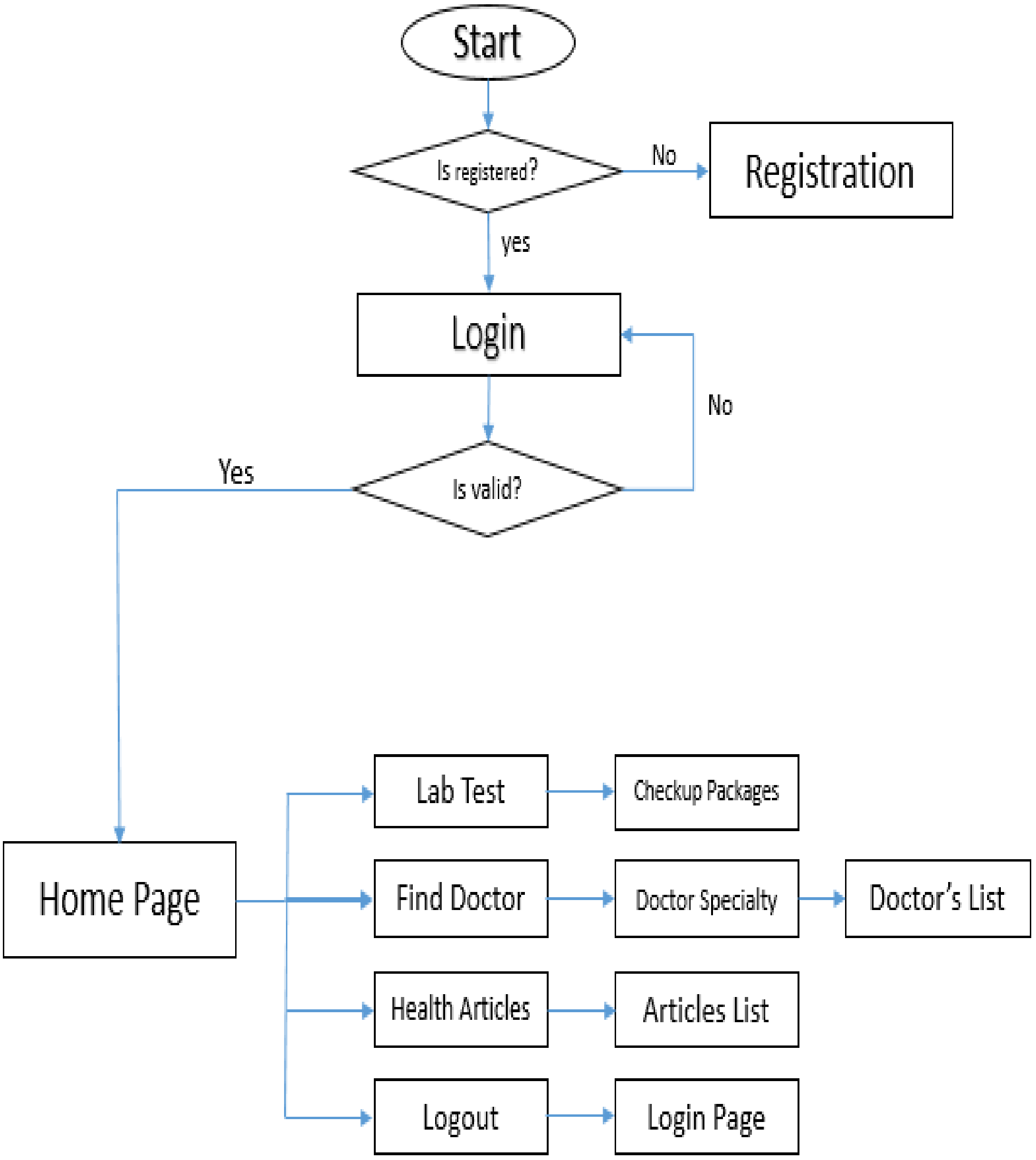
Healthcare View:

Sl.no	Name	Type	Constraints
1	Username	Text	From users(Username)
2	Email	Text	From Users(Email)
3	Fullname	Text	From Orderplaced(Fullname)
4	Address	Text	From Orderplaced(Address)
5	Contactno	Text	From Orderplaced(Contactno)
6	Pincode	Int	From Orderplaced(Pincode)
7	Date	Text	From Orderplaced(Date)
8	Time	Text	From Orderplaced(Time)
9	Amount	Text	From Orderplaced(Amount)
10	Otype	Text	From Orderplaced(Otype)

Cart Table:

Sl.no	Name	Type	Constrints
1	Username	Text	Foreign key References users(Username)
2	Product	Text	Not Null
3	Price	Float	Not Null
4	Otype	Text	Not Null

5.2 Dataflow Diagram:



6. PROJECT DESCRIPTION

6.1 List of Modules:

- Login page
- Checkup packages
- Find doctors
- Doctor’s specialty
- Health articles

6.2 Module Description:

i. Login Page:

The Login Page module is an essential component of an Android application that provides a secure and user-friendly way for users to access the app. Its main functionality revolves around user authentication and registration. Here's an expanded description of the Login Page module:

- **User Registration:**

The Login Page allows new users to register by providing their necessary details such as username, email address, and password. Input validation ensures that the provided information meets the specified criteria (e.g., password must contain 8 characters with a mixture of uppercase, lowercase, numbers and special characters.)

- **User Login:**

Registered users can use the Login Page to authenticate themselves and gain access to the application's features and data. Users typically enter their username/email and password to log in.Input validation ensures that the entered credentials are correct and match an existing user account.

ii. Checkup Packages Page:

The Checkup Packages module is a feature within an Android application that allows users to browse and select from a variety of health checkup packages. The module facilitates a seamless process for users to add their desired package to a cart and proceed to checkout.

iii. Find Doctors Page:

The "Find Doctors" module aims to simplify the process of finding and booking appointments with doctors. It provides users with the ability to search for doctors based on specialties, view detailed profiles, and conveniently book appointments. This module provides a convenient and streamlined way for users to find healthcare professionals who specialize in specific medical fields. The integration with backend systems ensures accurate and up-to-date information improves the overall user experience.

iv. Health Articles Page:

The "Health Articles" module is a valuable component of an Android application that offers users access to a collection of health-related articles. This module provides a platform for users to explore informative and educational content related to various health topics. This module aims to provide users with a valuable resource for accessing reliable and informative health-related content. The module enhances users' knowledge and promotes health awareness within the Android application.

7. SYSTEM TESTING:-

System testing is the testing of the behaviour of a complete and fully integrated software product based on the software requirements specification (SRS) document. In the main focus of this testing is to evaluate Business / Functional/End-user requirements. This testing is to be carried out only after System Integration Testing is completed where both Functional & Non-Functional requirements are verified. In the integration testing testers are concentrated on finding bugs/defects on integrated modules. But in the Software System Testing testers are concentrated on finding bugs/defects based on software application behaviour, software design and expectation of end user.

System Testing is a crucial phase in the software development life cycle (SDLC) that focuses on evaluating the complete and integrated system to ensure that it meets the specified requirements and functions as expected. It involves testing the system as a whole rather than individual components or units.

Importance of system testing:

- In Software Development Life Cycle the System Testing is performed as the first level of testing where the System is tested as a whole.
- In this step of testing check if system meets functional requirements or not.
- System Testing enables you to test, validate and verify both the Application Architecture and Business requirements.
- The application/System is tested in an environment that particularly resembles the effective production environment.

Unit Testing:

This type of testing focuses on testing individual units or components of the application, such as methods or classes. Unit tests help identify bugs and ensure that each unit of code works as intended.

Integration Testing:

Integration testing verifies the interaction between different modules or components of the application. It ensures that the integrated parts work together correctly and that data flows smoothly between them.

Functional Testing:

Functional testing ensures that the application functions as expected from a user's perspective. It involves testing various features, user interactions, and user interface elements to validate their correctness and usability.

UI Testing:

User Interface (UI) testing validates the graphical user interface of the application. It checks the layout, appearance, and responsiveness of the UI elements across different devices and screen sizes.

Performance Testing:

Performance testing measures the responsiveness, speed, stability, and resource usage of the application under different conditions. It helps identify bottlenecks, memory leaks, and performance issues that could degrade the user experience.

Security Testing:

It includes testing for authentication, authorization, data encryption, and protection against common security threats like SQL injection or cross-site scripting.

8. CONCLUSION & FUTURE ENCHANCEMENT

Conclusion:

The Health Cart app offers a comprehensive solution for users' health-related needs. By providing features like user login, lab test booking, doctor appointment booking, order details, and health articles, the app aims to simplify the process of accessing health services. The centralized approach and user-friendly interface enhance user convenience and provide a seamless experience.

Future Work:

While the current version of the Health Cart app offers several valuable features, there is always room for improvement and future enhancements. Some potential areas for further development include:

- **Real-time doctor availability:** Integrate a real-time availability feature for doctors, allowing users to see doctors' current availability and book appointments accordingly.
- **Health tracker:** Incorporate a health tracker feature to help users monitor their health parameters, track progress, and set goals.
- **Reminder system:** Implement a reminder system to send notifications to users regarding upcoming lab tests, doctor appointments, or health-related activities.
- **Telemedicine:** Introduce a telemedicine feature that allows users to consult doctors remotely through video calls or chat, providing a convenient alternative to in-person visits.
- **Personalized recommendations**: Utilize machine learning algorithms to provide personalized health recommendations based on users' health history and preferences.

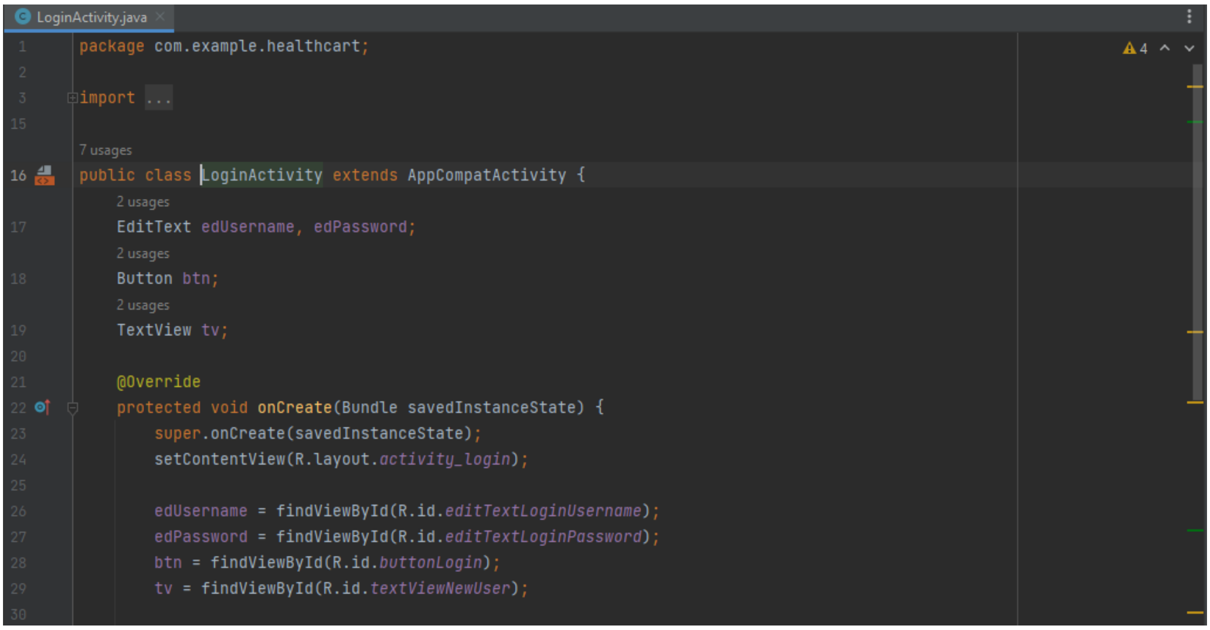
9. BIBLIOGRAPHY & REFERENCE:

- https://stackoverflow.com/questions/34353220/how-do-i-select-android-sdk-in-android-studio
- https://stackoverflow.com/questions/5350624/set-icon-for-android-application
- https://stackoverflow.com/questions/2091465/how-do-i-pass-data-between-activities-in-android-application
- https://www.youtube.com/watch?v=HagZBlNevLQ
- https://github.com/hussien89aa/AndroidTutorialForBeginners

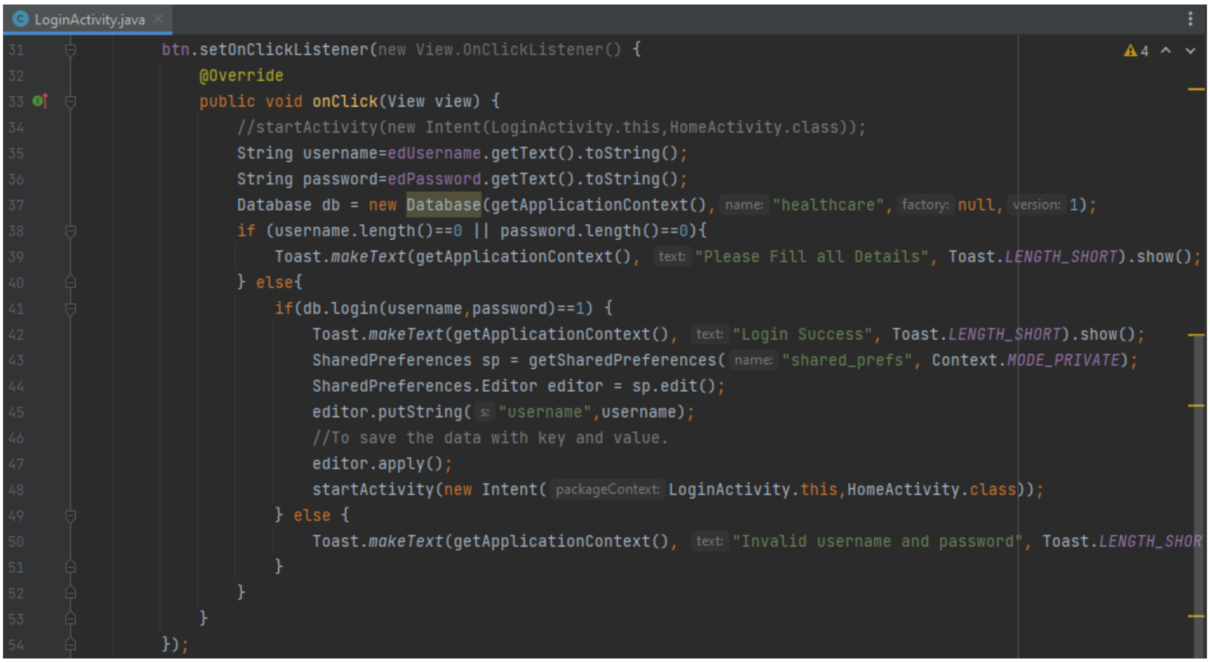
10. APPENDICES:-

10.1 Sample Coding:

Login Page:

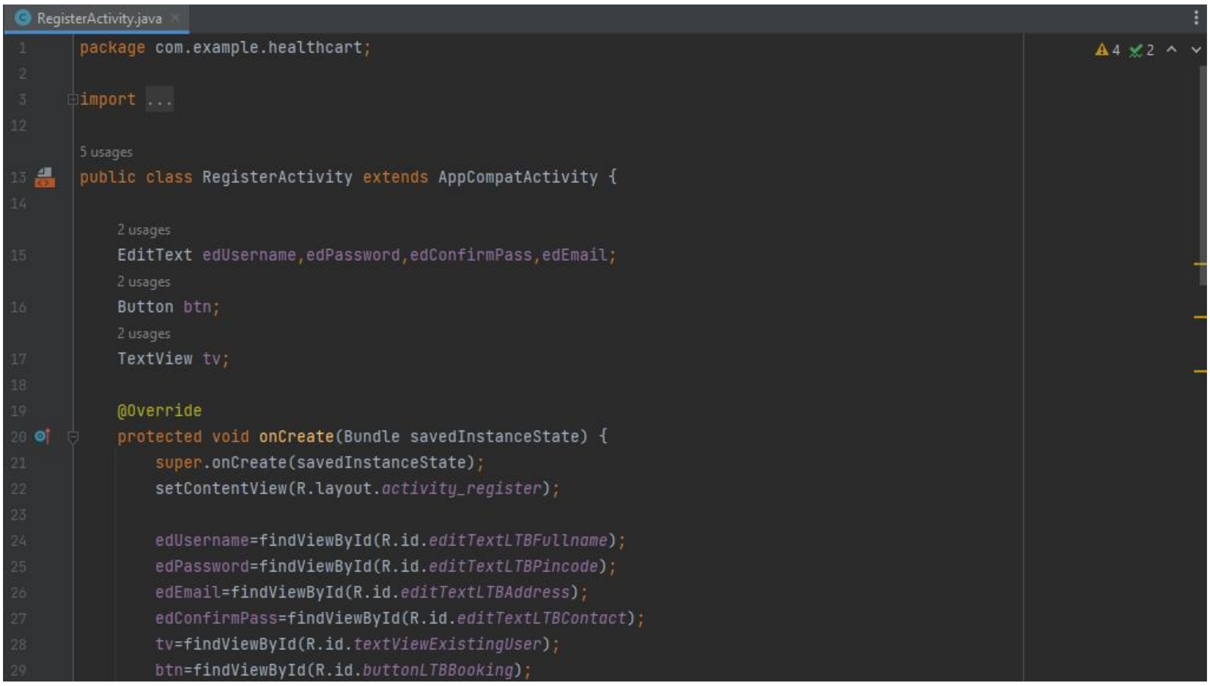


```
package com.example.healthcart;

import ...

7 usages
public class LoginActivity extends AppCompatActivity {
    2 usages
    EditText edUsername, edPassword;
    2 usages
    Button btn;
    2 usages
    TextView tv;

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

        edUsername = findViewById(R.id.editTextLoginUsername);
        edPassword = findViewById(R.id.editTextLoginPassword);
        btn = findViewById(R.id.buttonLogin);
        tv = findViewById(R.id.textViewNewUser);

```

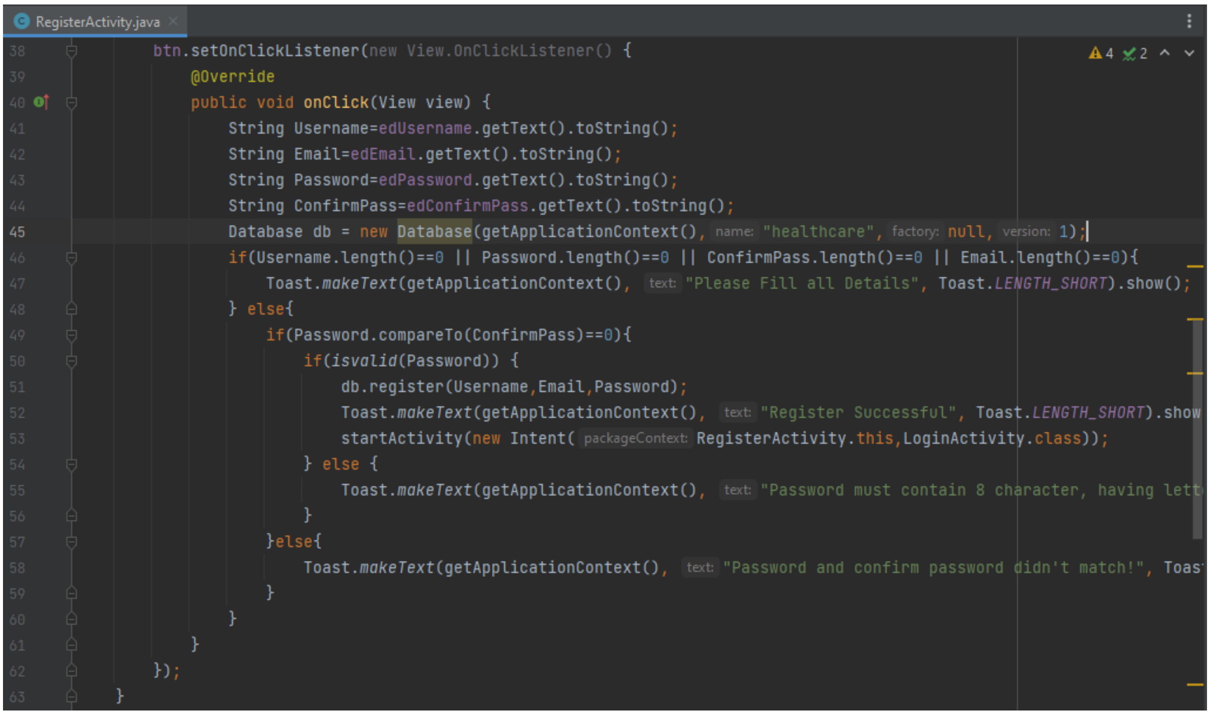


```
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //startActivity(new Intent(LoginActivity.this,HomeActivity.class));
                String username=edUsername.getText().toString();
                String password=edPassword.getText().toString();
                Database db = new Database(getApplicationContext(), name: "healthcare", factory: null, version: 1);
                if (username.length()==0 || password.length()==0){
                    Toast.makeText(getApplicationContext(), text: "Please Fill all Details", Toast.LENGTH_SHORT).show();
                } else{
                    if(db.login(username,password)==1) {
                        Toast.makeText(getApplicationContext(), text: "Login Success", Toast.LENGTH_SHORT).show();
                        SharedPreferences sp = getSharedPreferences( name: "shared_prefs", Context.MODE_PRIVATE);
                        SharedPreferences.Editor editor = sp.edit();
                        editor.putString( s: "username",username);
                        //To save the data with key and value.
                        editor.apply();
                        startActivity(new Intent( packageContext: LoginActivity.this,HomeActivity.class));
                    } else {
                        Toast.makeText(getApplicationContext(), text: "Invalid username and password", Toast.LENGTH_SHOR
                    }
                }
            }
        });
```

Register Page:

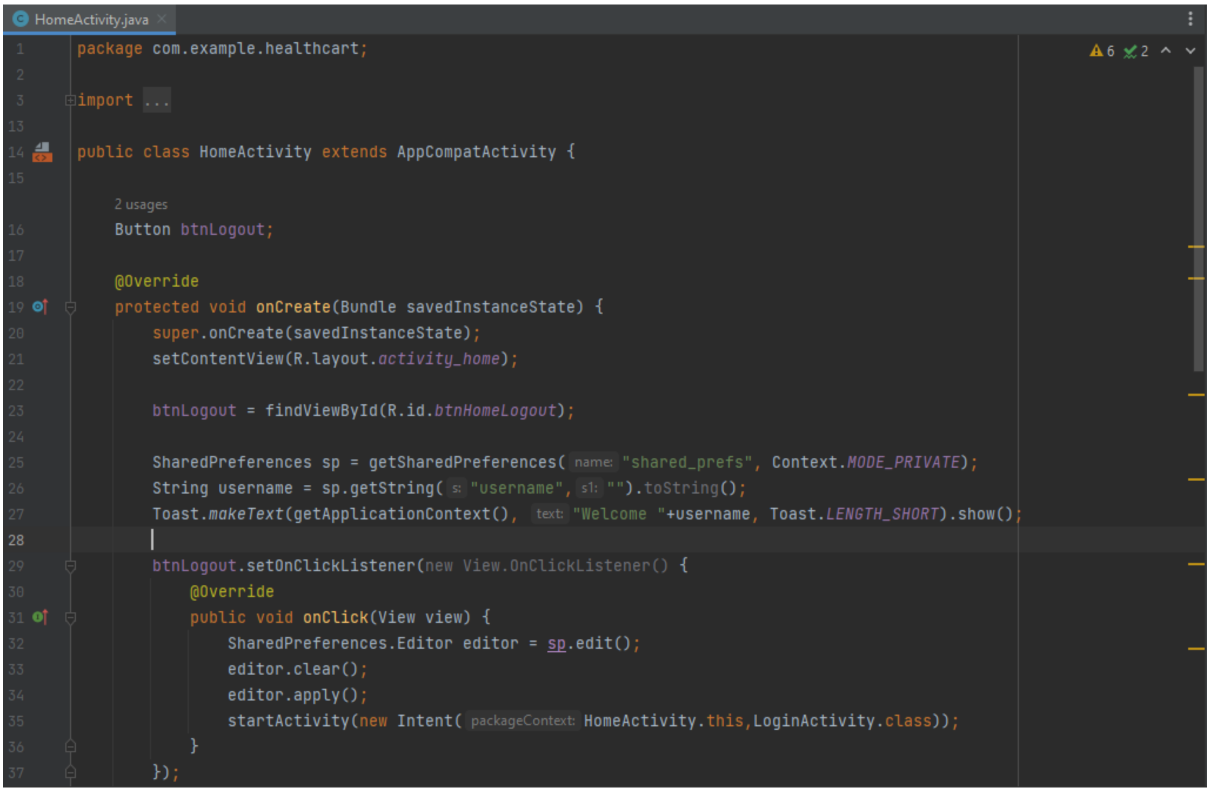


```
package com.example.healthcart;

import ...

5 usages
public class RegisterActivity extends AppCompatActivity {

    2 usages
    EditText edUsername,edPassword,edConfirmPass,edEmail;
    2 usages
    Button btn;
    2 usages
    TextView tv;

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

        edUsername=findViewById(R.id.editTextLTBFullname);
        edPassword=findViewById(R.id.editTextLTBPincode);
        edEmail=findViewById(R.id.editTextLTBAddress);
        edConfirmPass=findViewById(R.id.editTextLTBContact);
        tv=findViewById(R.id.textViewExistingUser);
        btn=findViewById(R.id.buttonLTBBooking);
```



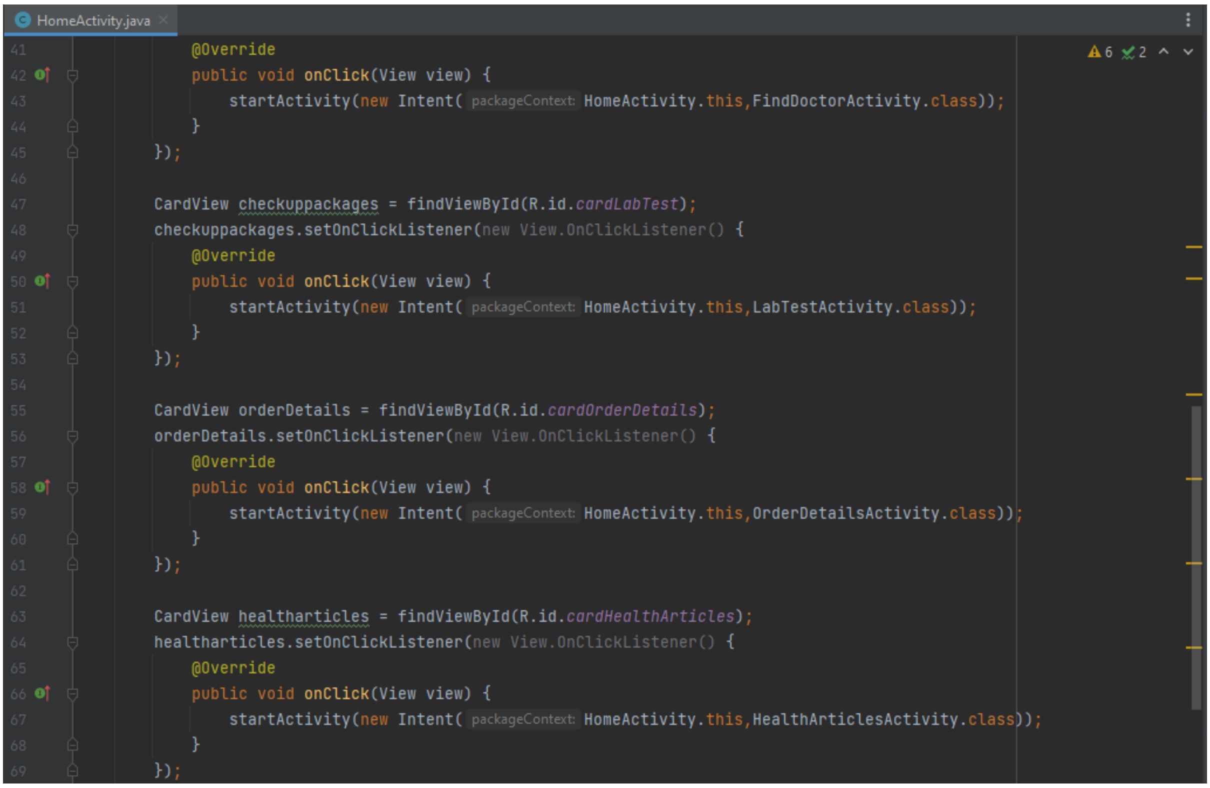
```
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String Username=edUsername.getText().toString();
                String Email=edEmail.getText().toString();
                String Password=edPassword.getText().toString();
                String ConfirmPass=edConfirmPass.getText().toString();
                Database db = new Database(getApplicationContext(), name: "healthcare", factory: null, version: 1);
                if(Username.length()==0 || Password.length()==0 || ConfirmPass.length()==0 || Email.length()==0){
                    Toast.makeText(getApplicationContext(), text: "Please Fill all Details", Toast.LENGTH_SHORT).show();
                } else{
                    if(Password.compareTo(ConfirmPass)==0){
                        if(isvalid(Password)) {
                            db.register(Username,Email,Password);
                            Toast.makeText(getApplicationContext(), text: "Register Successful", Toast.LENGTH_SHORT).show
                            startActivity(new Intent( packageContext: RegisterActivity.this,LoginActivity.class));
                        } else {
                            Toast.makeText(getApplicationContext(), text: "Password must contain 8 character, having lett
                        }
                    }else{
                        Toast.makeText(getApplicationContext(), text: "Password and confirm password didn't match!", Toas
                    }
                }
            }
        });
    }
```

Home Page:



HomeActivity.java

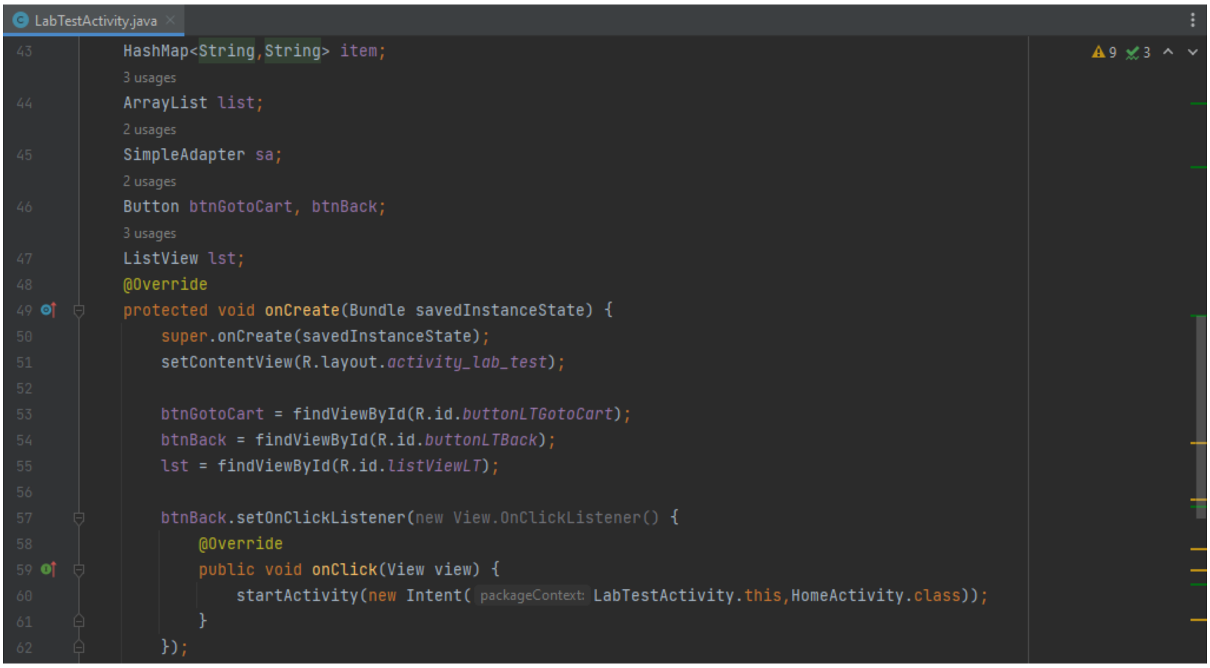
```
package com.example.healthcart;

import ...

public class HomeActivity extends AppCompatActivity {

    2 usages
    Button btnLogout;

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

        btnLogout = findViewById(R.id.btnHomeLogout);

        SharedPreferences sp = getSharedPreferences( name: "shared_prefs", Context.MODE_PRIVATE);
        String username = sp.getString( s: "username", s1: "").toString();
        Toast.makeText(getApplicationContext(), text: "Welcome "+username, Toast.LENGTH_SHORT).show();

        btnLogout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                SharedPreferences.Editor editor = sp.edit();
                editor.clear();
                editor.apply();
                startActivity(new Intent( packageContext: HomeActivity.this,LoginActivity.class));
            }
        });
```



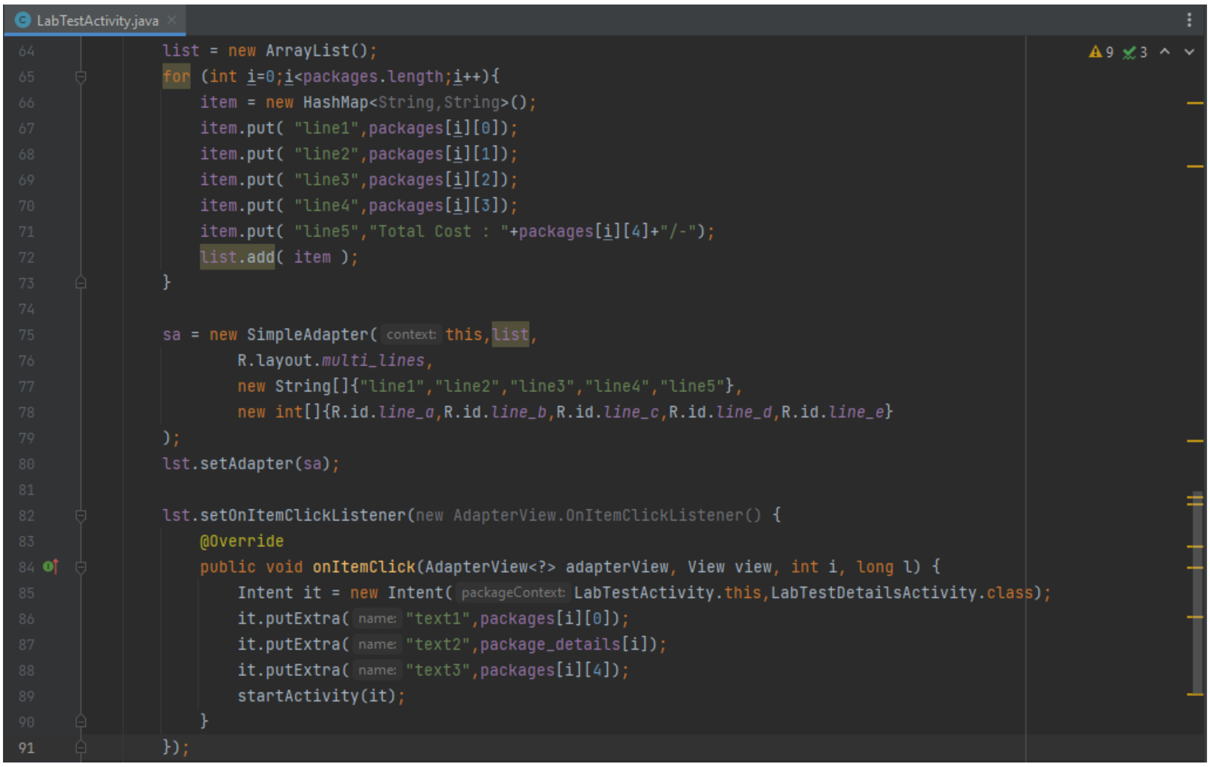
HomeActivity.java

```
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: HomeActivity.this,FindDoctorActivity.class));
            }
        });

        CardView checkuppackages = findViewById(R.id.cardLabTest);
        checkuppackages.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: HomeActivity.this,LabTestActivity.class));
            }
        });

        CardView orderDetails = findViewById(R.id.cardOrderDetails);
        orderDetails.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: HomeActivity.this,OrderDetailsActivity.class));
            }
        });

        CardView healtharticles = findViewById(R.id.cardHealthArticles);
        healtharticles.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: HomeActivity.this,HealthArticlesActivity.class));
            }
        });
```

Lab Test Page:

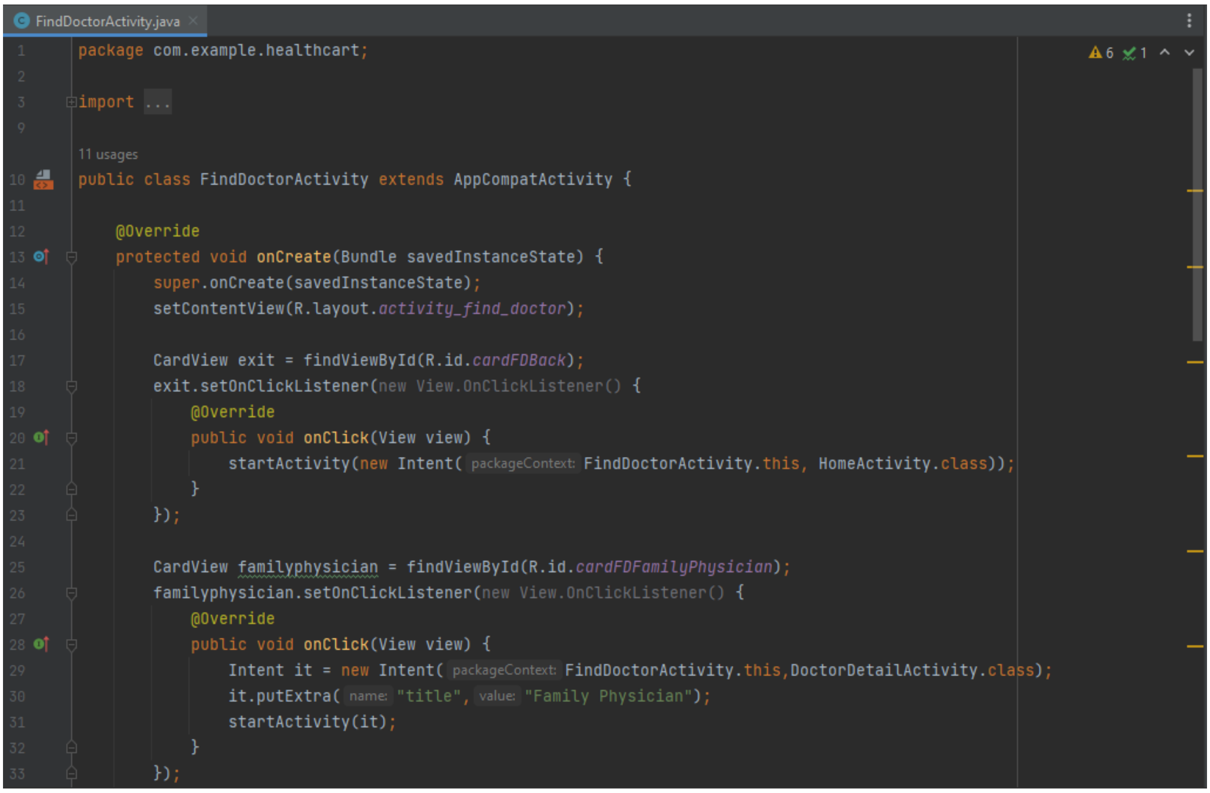


```java
    HashMap<String,String> item;
    3 usages
    ArrayList list;
    2 usages
    SimpleAdapter sa;
    2 usages
    Button btnGotoCart, btnBack;
    3 usages
    ListView lst;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lab_test);

        btnGotoCart = findViewById(R.id.buttonLTGotoCart);
        btnBack = findViewById(R.id.buttonLTBack);
        lst = findViewById(R.id.listViewLT);

        btnBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: LabTestActivity.this,HomeActivity.class));
            }
        });
```

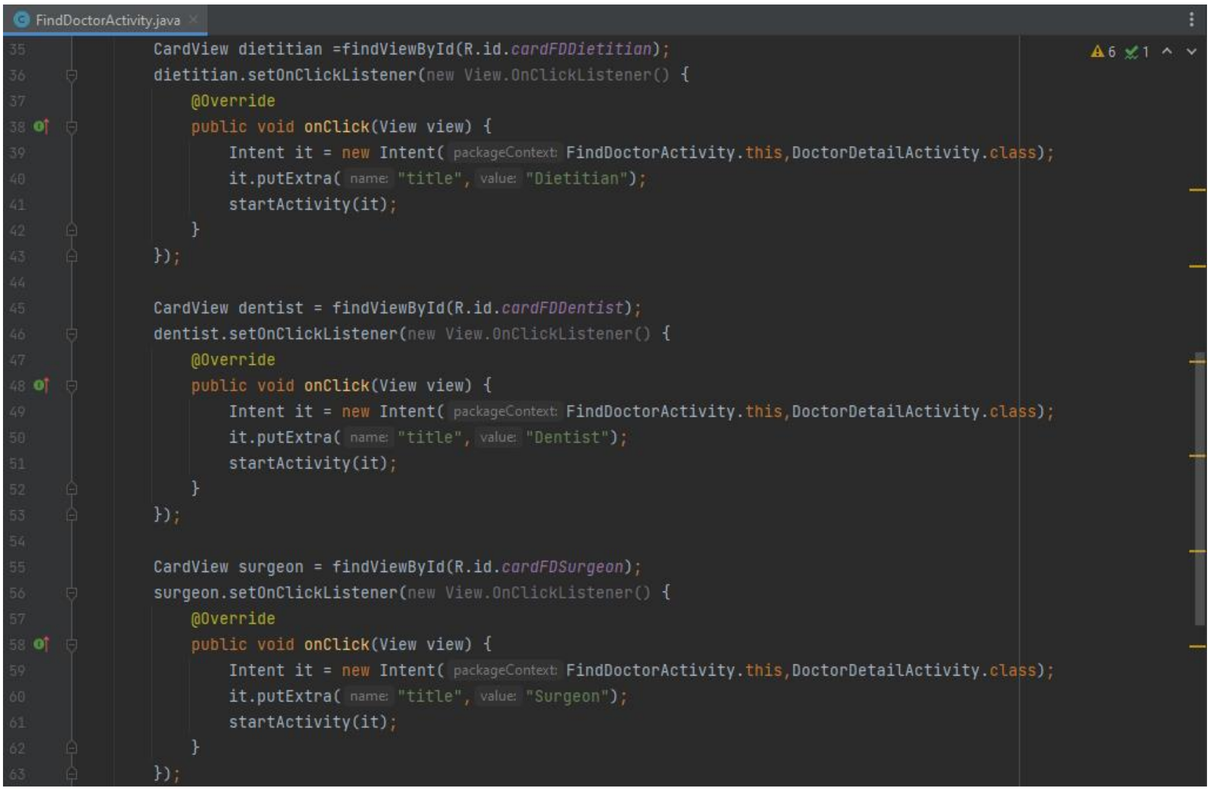


```java
        list = new ArrayList();
        for (int i=0;i<packages.length;i++){
            item = new HashMap<String,String>();
            item.put( "line1",packages[i][0]);
            item.put( "line2",packages[i][1]);
            item.put( "line3",packages[i][2]);
            item.put( "line4",packages[i][3]);
            item.put( "line5","Total Cost : "+packages[i][4]+"/-");
            list.add( item );
        }

        sa = new SimpleAdapter( context: this,list,
                R.layout.multi_lines,
                new String[]{"line1","line2","line3","line4","line5"},
                new int[]{R.id.line_a,R.id.line_b,R.id.line_c,R.id.line_d,R.id.line_e}
        );
        lst.setAdapter(sa);

        lst.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                Intent it = new Intent( packageContext: LabTestActivity.this,LabTestDetailsActivity.class);
                it.putExtra( name: "text1",packages[i][0]);
                it.putExtra( name: "text2",package_details[i]);
                it.putExtra( name: "text3",packages[i][4]);
                startActivity(it);
            }
        });
```

Find Doctor Page:

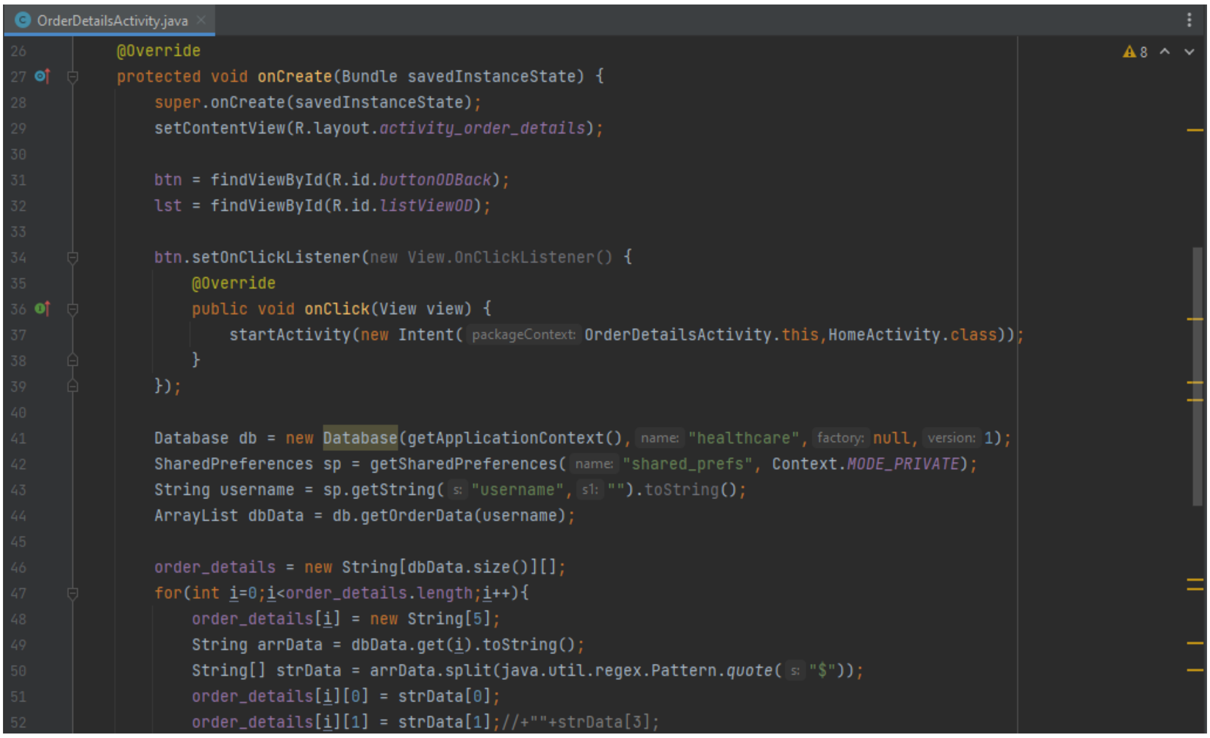


```java
package com.example.healthcart;

import ...

11 usages
public class FindDoctorActivity extends AppCompatActivity {

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

        CardView exit = findViewById(R.id.cardFDBack);
        exit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: FindDoctorActivity.this, HomeActivity.class));
            }
        });

        CardView familyphysician = findViewById(R.id.cardFDFamilyPhysician);
        familyphysician.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent it = new Intent( packageContext: FindDoctorActivity.this,DoctorDetailActivity.class);
                it.putExtra( name: "title", value: "Family Physician");
                startActivity(it);
            }
        });
```



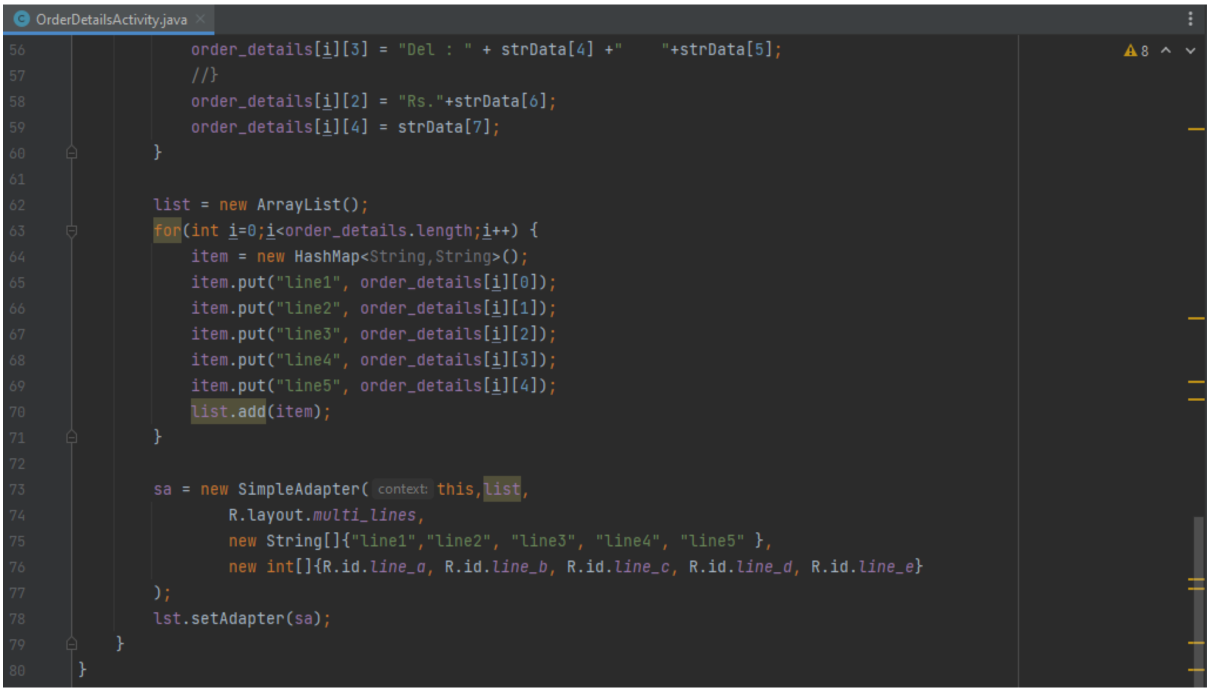
```java
        CardView dietitian =findViewById(R.id.cardFDDietitian);
        dietitian.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent it = new Intent( packageContext: FindDoctorActivity.this,DoctorDetailActivity.class);
                it.putExtra( name: "title", value: "Dietitian");
                startActivity(it);
            }
        });

        CardView dentist = findViewById(R.id.cardFDDentist);
        dentist.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent it = new Intent( packageContext: FindDoctorActivity.this,DoctorDetailActivity.class);
                it.putExtra( name: "title", value: "Dentist");
                startActivity(it);
            }
        });

        CardView surgeon = findViewById(R.id.cardFDSurgeon);
        surgeon.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent it = new Intent( packageContext: FindDoctorActivity.this,DoctorDetailActivity.class);
                it.putExtra( name: "title", value: "Surgeon");
                startActivity(it);
            }
        });
```

Order Details Page:



OrderDetailsActivity.java

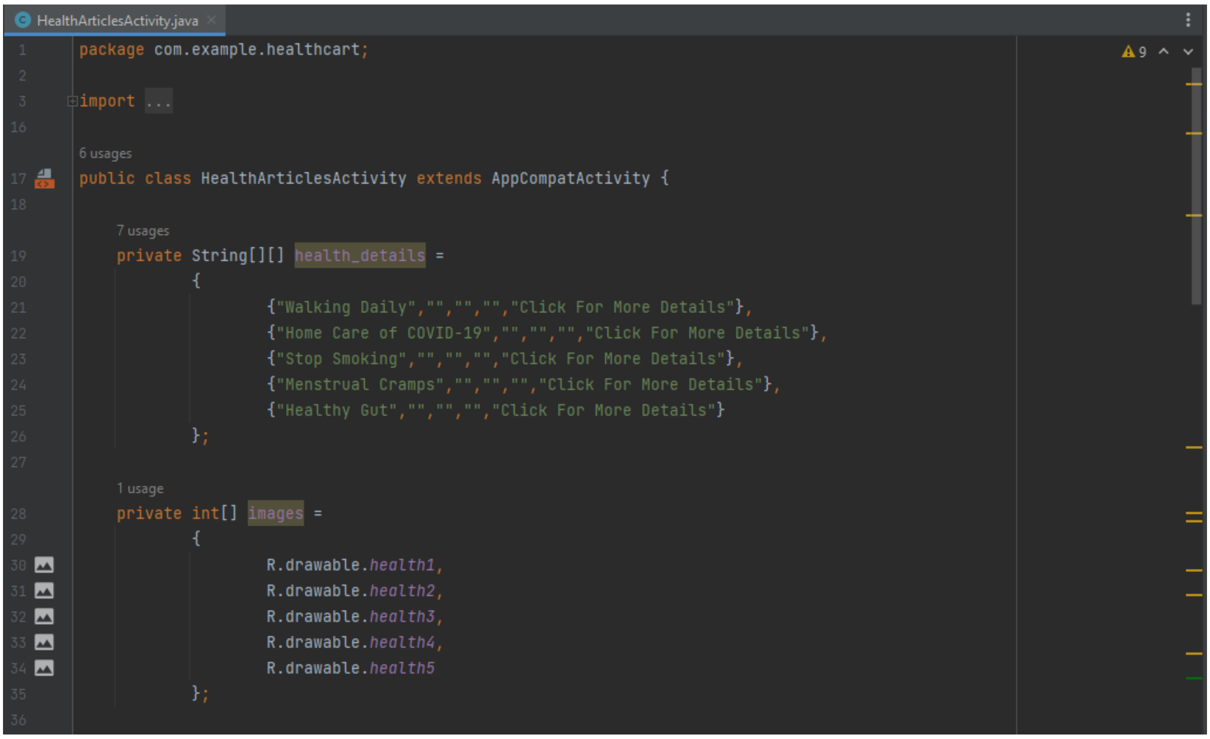
```java
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_details);

        btn = findViewById(R.id.buttonODBack);
        lst = findViewById(R.id.listViewOD);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: OrderDetailsActivity.this,HomeActivity.class));
            }
        });

        Database db = new Database(getApplicationContext(), name: "healthcare", factory: null, version: 1);
        SharedPreferences sp = getSharedPreferences( name: "shared_prefs", Context.MODE_PRIVATE);
        String username = sp.getString( s: "username", s1: "").toString();
        ArrayList dbData = db.getOrderData(username);

        order_details = new String[dbData.size()][];
        for(int i=0;i<order_details.length;i++){
            order_details[i] = new String[5];
            String arrData = dbData.get(i).toString();
            String[] strData = arrData.split(java.util.regex.Pattern.quote( s: "$"));
            order_details[i][0] = strData[0];
            order_details[i][1] = strData[1];//+""+strData[3];
```



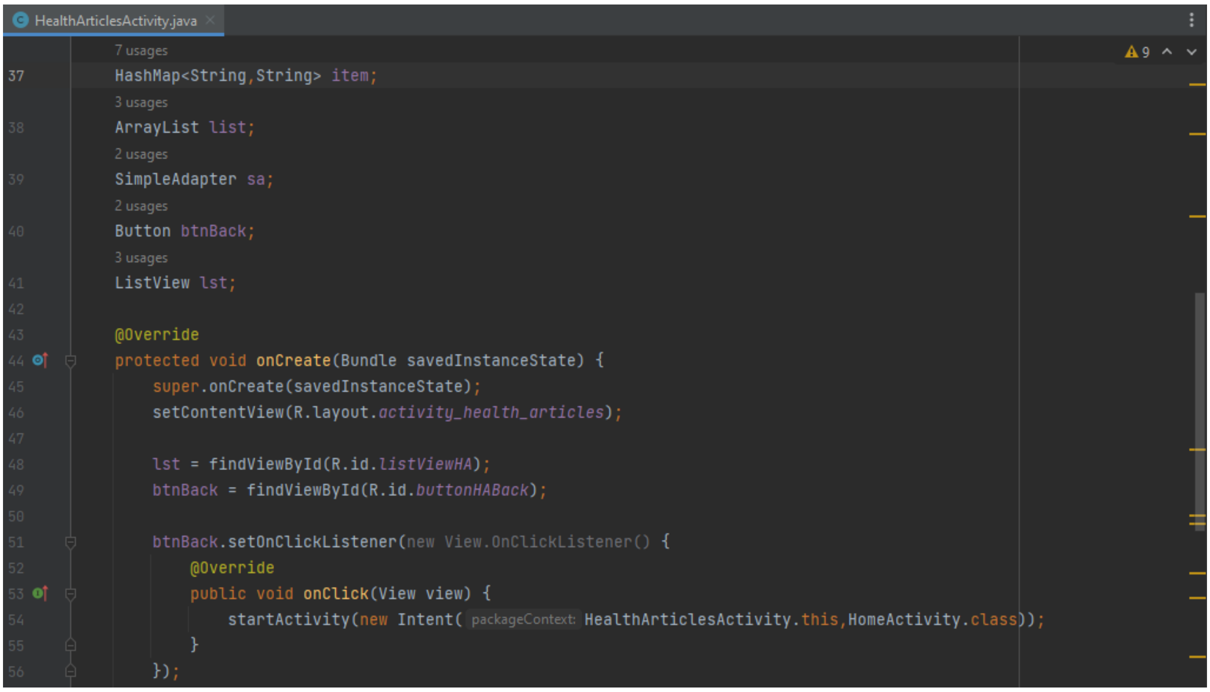
OrderDetailsActivity.java

```java
            order_details[i][3] = "Del : " + strData[4] +"     "+strData[5];
            //}
            order_details[i][2] = "Rs."+strData[6];
            order_details[i][4] = strData[7];
        }

        list = new ArrayList();
        for(int i=0;i<order_details.length;i++) {
            item = new HashMap<String,String>();
            item.put("line1", order_details[i][0]);
            item.put("line2", order_details[i][1]);
            item.put("line3", order_details[i][2]);
            item.put("line4", order_details[i][3]);
            item.put("line5", order_details[i][4]);
            list.add(item);
        }

        sa = new SimpleAdapter( context: this,list,
                R.layout.multi_lines,
                new String[]{"line1","line2", "line3", "line4", "line5" },
                new int[]{R.id.line_a, R.id.line_b, R.id.line_c, R.id.line_d, R.id.line_e}
        );
        lst.setAdapter(sa);
    }
}
```

Health Articles Page:

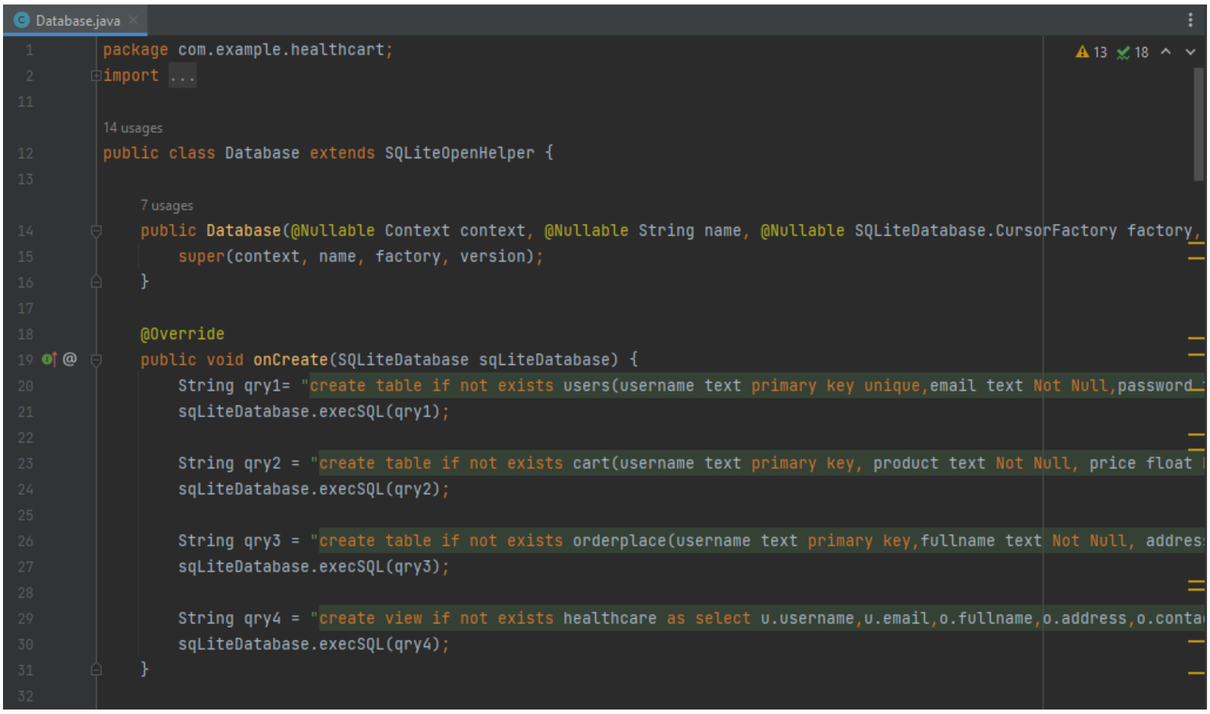


```
package com.example.healthcart;

import ...

6 usages
public class HealthArticlesActivity extends AppCompatActivity {

    7 usages
    private String[][] health_details =
            {
                    {"Walking Daily","","","","Click For More Details"},
                    {"Home Care of COVID-19","","","","Click For More Details"},
                    {"Stop Smoking","","","","Click For More Details"},
                    {"Menstrual Cramps","","","","Click For More Details"},
                    {"Healthy Gut","","","","Click For More Details"}
            };

    1 usage
    private int[] images =
            {
                    R.drawable.health1,
                    R.drawable.health2,
                    R.drawable.health3,
                    R.drawable.health4,
                    R.drawable.health5
            };

```

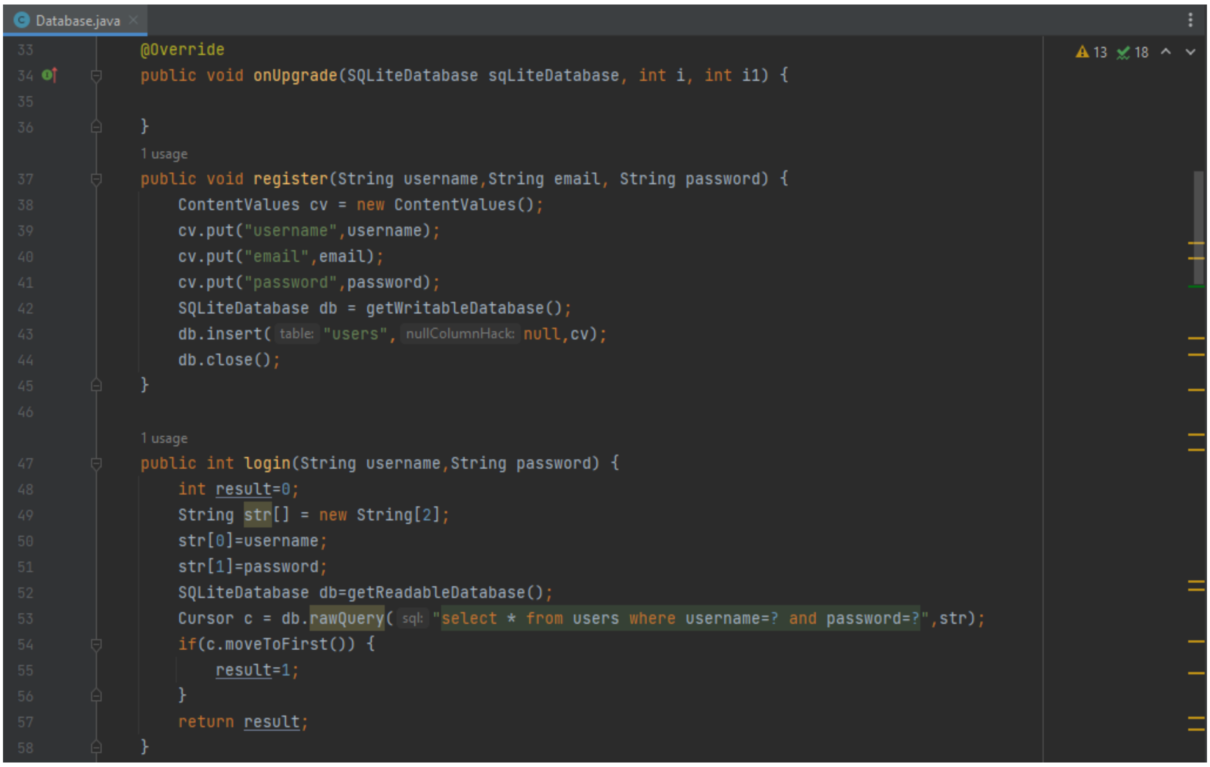


```
    7 usages
    HashMap<String,String> item;
    3 usages
    ArrayList list;
    2 usages
    SimpleAdapter sa;
    2 usages
    Button btnBack;
    3 usages
    ListView lst;

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

        lst = findViewById(R.id.listViewHA);
        btnBack = findViewById(R.id.buttonHABack);

        btnBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent( packageContext: HealthArticlesActivity.this,HomeActivity.class));
            }
        });
```

Database Code:



```java
package com.example.healthcart;
import ...

14 usages
public class Database extends SQLiteOpenHelper {

    7 usages
    public Database(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory,
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String qry1= "create table if not exists users(username text primary key unique,email text Not Null,password
        sqLiteDatabase.execSQL(qry1);

        String qry2 = "create table if not exists cart(username text primary key, product text Not Null, price float
        sqLiteDatabase.execSQL(qry2);

        String qry3 = "create table if not exists orderplace(username text primary key,fullname text Not Null, addres
        sqLiteDatabase.execSQL(qry3);

        String qry4 = "create view if not exists healthcare as select u.username,u.email,o.fullname,o.address,o.conta
        sqLiteDatabase.execSQL(qry4);
    }

```

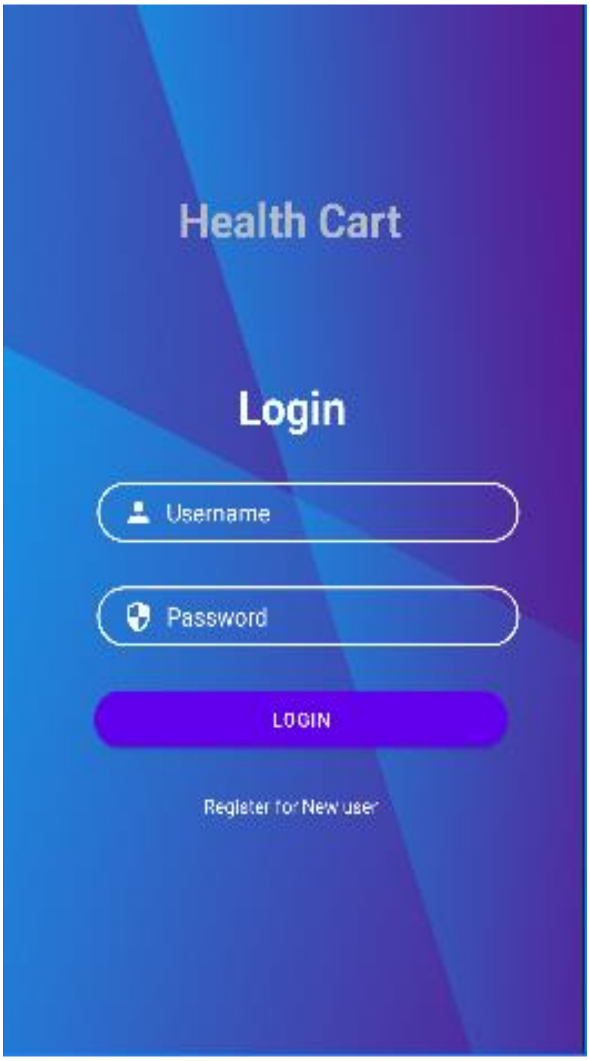


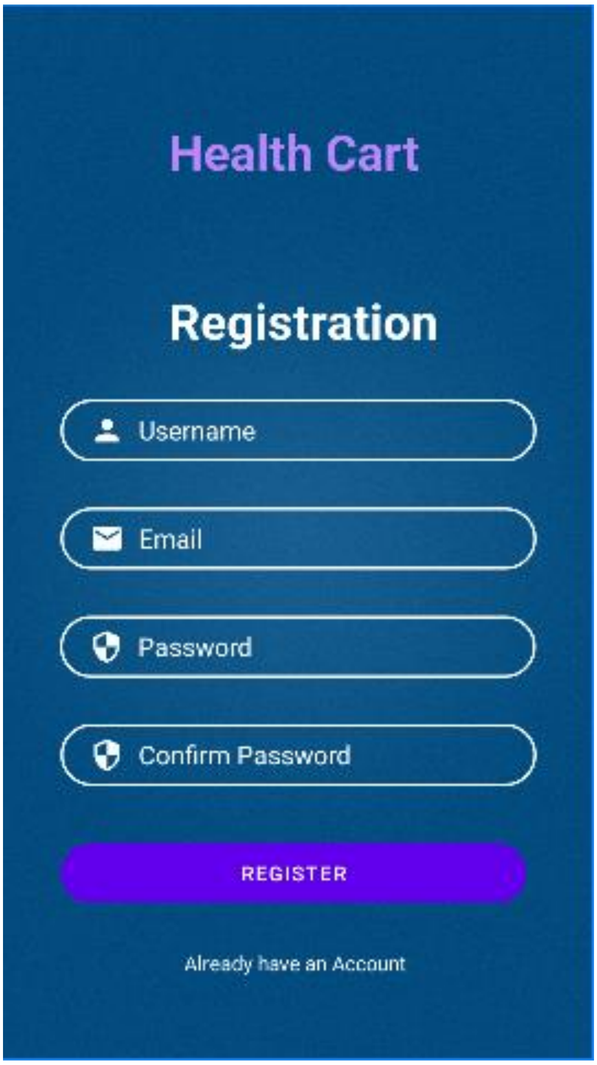
```java
    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }
    1 usage
    public void register(String username,String email, String password) {
        ContentValues cv = new ContentValues();
        cv.put("username",username);
        cv.put("email",email);
        cv.put("password",password);
        SQLiteDatabase db = getWritableDatabase();
        db.insert( table: "users", nullColumnHack: null,cv);
        db.close();
    }

    1 usage
    public int login(String username,String password) {
        int result=0;
        String str[] = new String[2];
        str[0]=username;
        str[1]=password;
        SQLiteDatabase db=getReadableDatabase();
        Cursor c = db.rawQuery( sql: "select * from users where username=? and password=?",str);
        if(c.moveToFirst()) {
            result=1;
        }
        return result;
    }
```

10.2 Screenshots

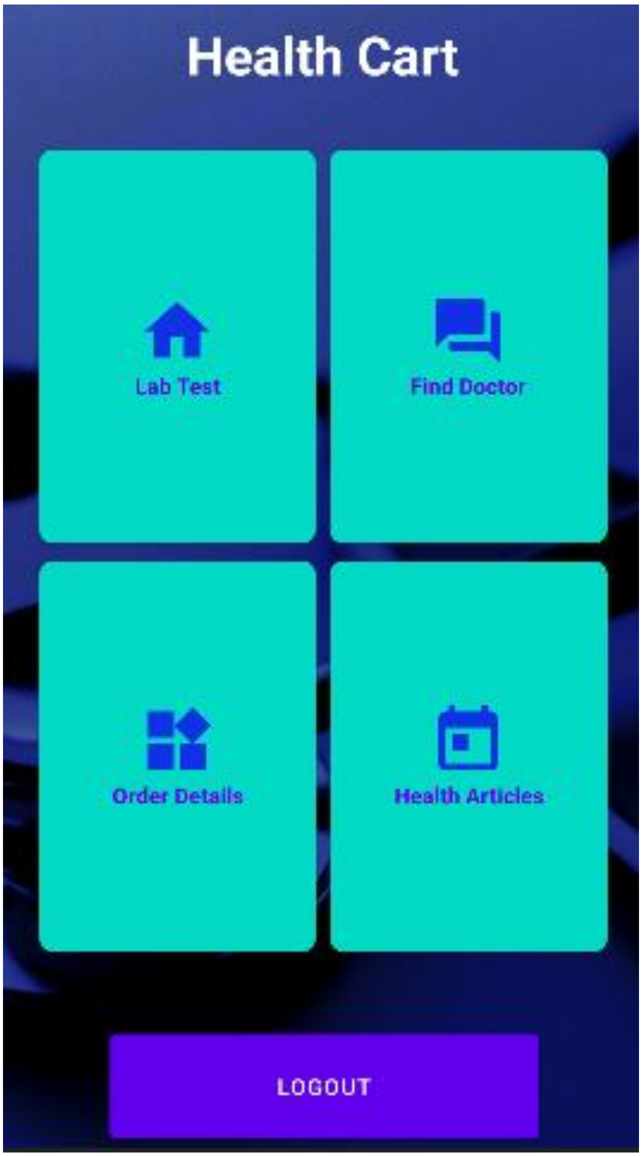
1. Login Page:

2. Registration Page:



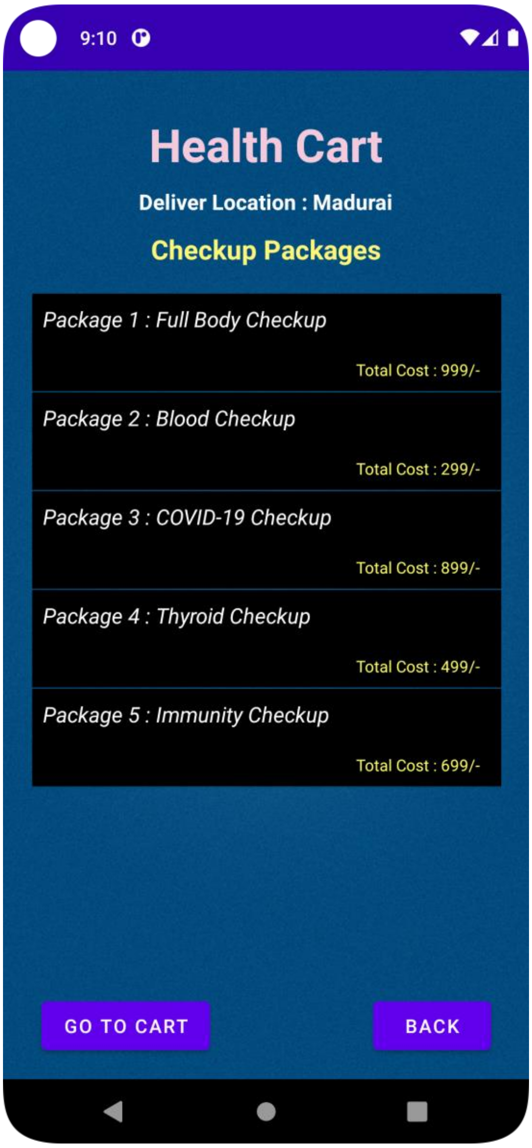


3. Home Page:



4. Lab Test page:

i. Packages List:



Health Cart

Deliver Location : Madurai

Checkup Packages

Package 1 : Full Body Checkup

Total Cost : 999/-

Package 2 : Blood Checkup

Total Cost : 299/-

Package 3 : COVID-19 Checkup

Total Cost : 899/-

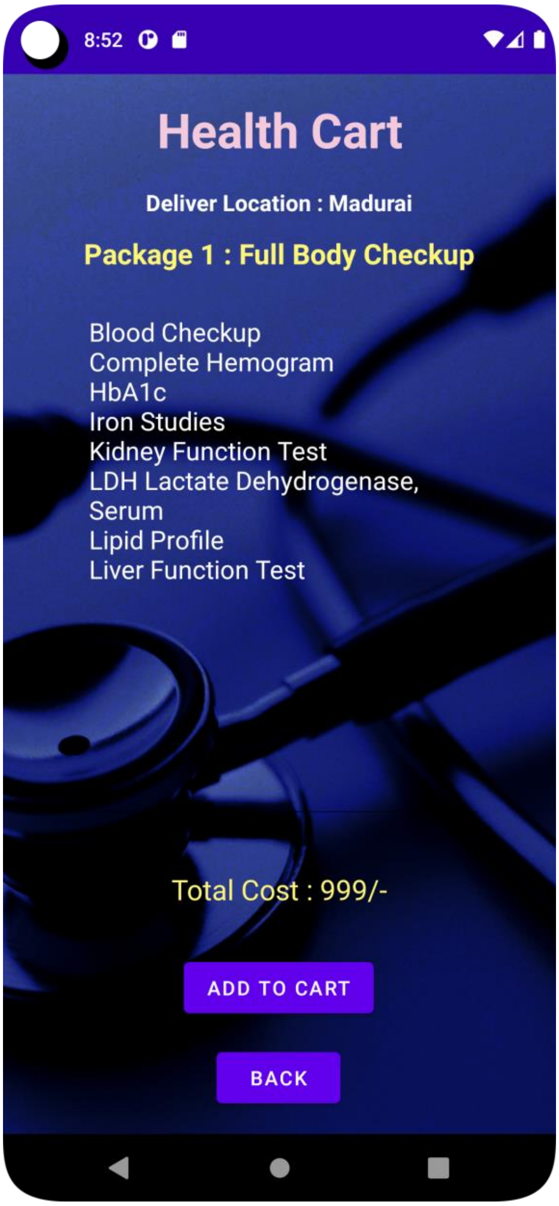
Package 4 : Thyroid Checkup

Total Cost : 499/-

Package 5 : Immunity Checkup

Total Cost : 699/-

ii. Package Details:



8:52

Health Cart

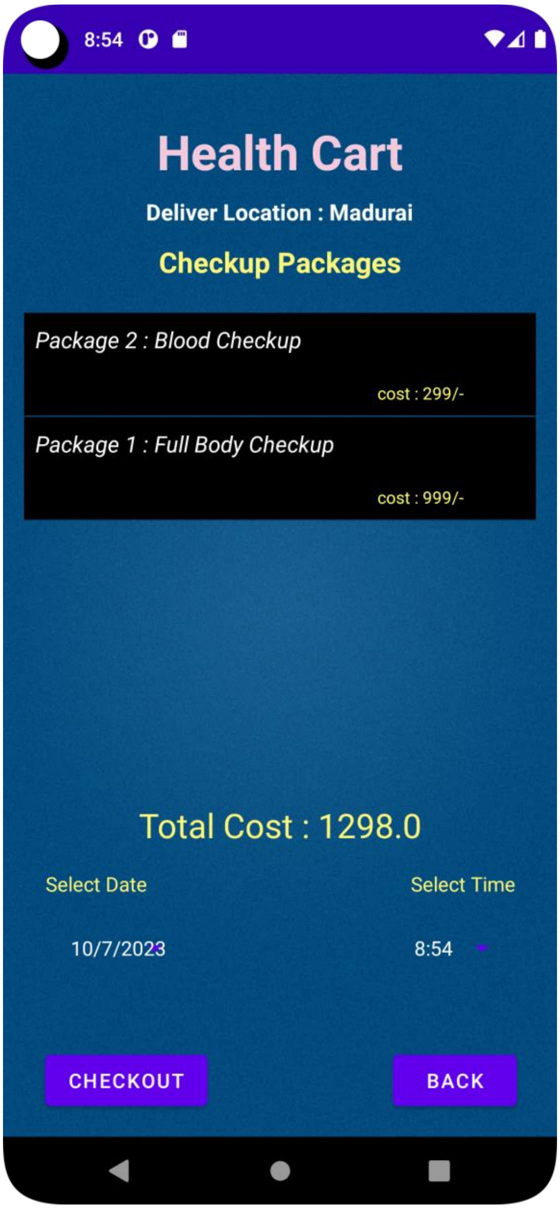
Deliver Location : Madurai

Package 1 : Full Body Checkup

Blood Checkup
Complete Hemogram
HbA1c
Iron Studies
Kidney Function Test
LDH Lactate Dehydrogenase, Serum
Lipid Profile
Liver Function Test

Total Cost : 999/-

iii. Go To Cart Button:



8:54

Health Cart

Deliver Location : Madurai

Checkup Packages

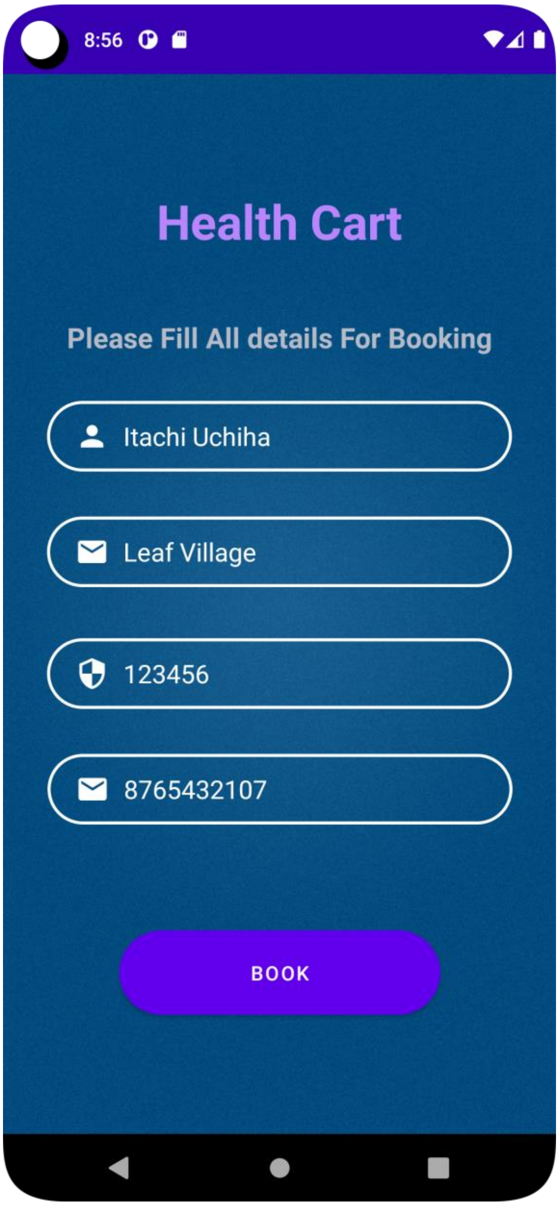
Package 2 : Blood Checkup	cost : 299/-
Package 1 : Full Body Checkup	cost : 999/-

Total Cost : 1298.0

Select Date: 10/7/2023

Select Time: 8:54

iv. Checkout Button Page:



Health Cart

Please Fill All details For Booking

Itachi Uchiha

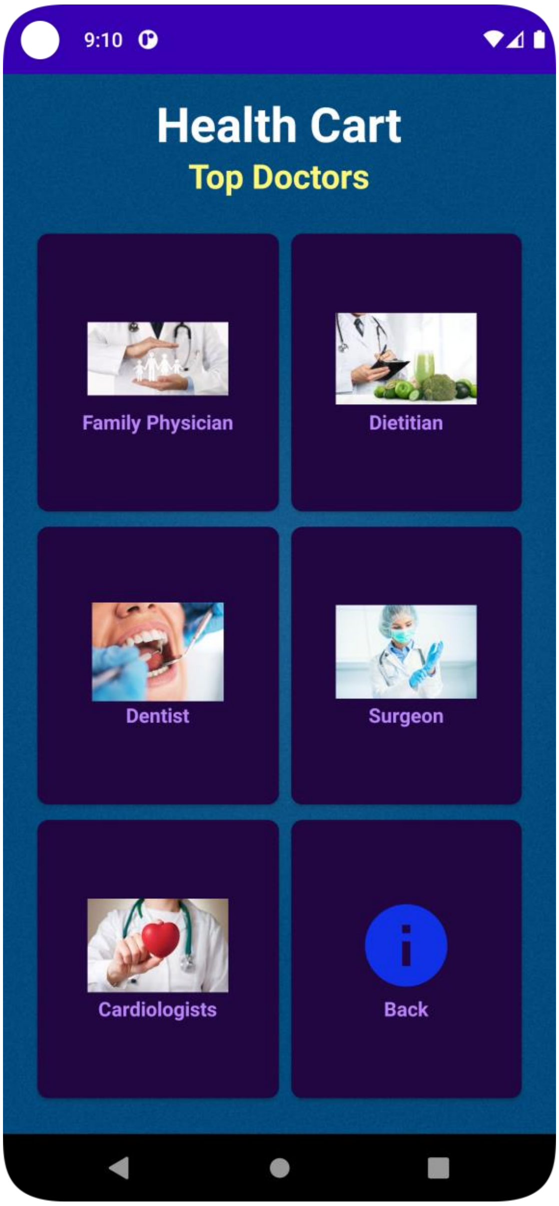
Leaf Village

123456

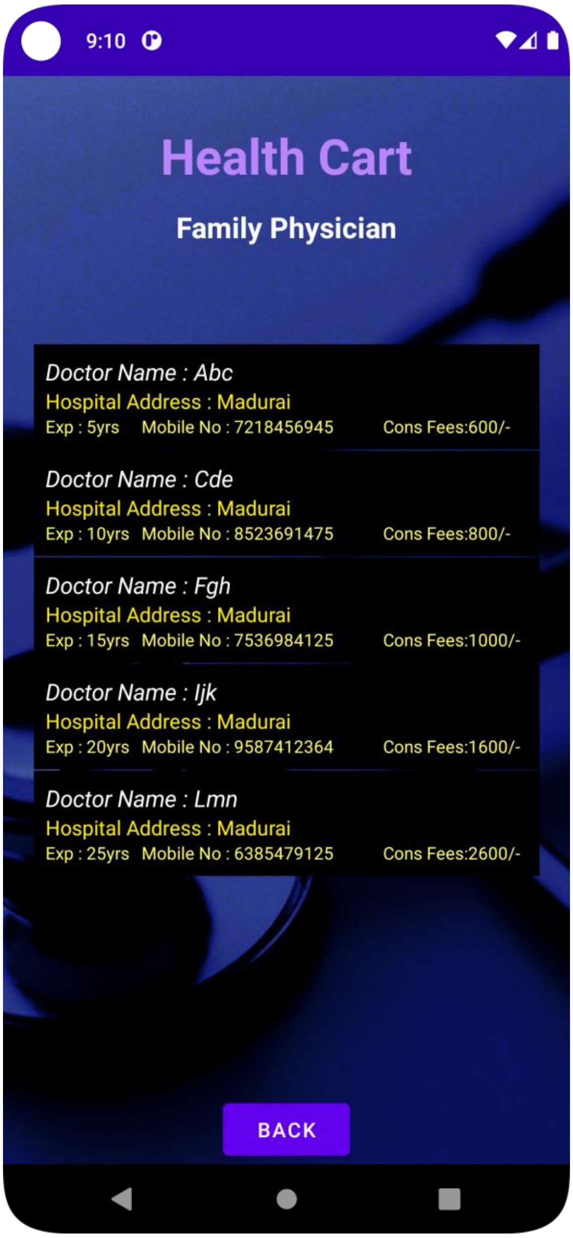
8765432107

5. Find Doctors Page:

i. Doctor’s Specialty:



ii. Doctor’s List:



Doctor Name : Abc
Hospital Address : Madurai
Exp : 5yrs Mobile No : 7218456945 Cons Fees:600/-

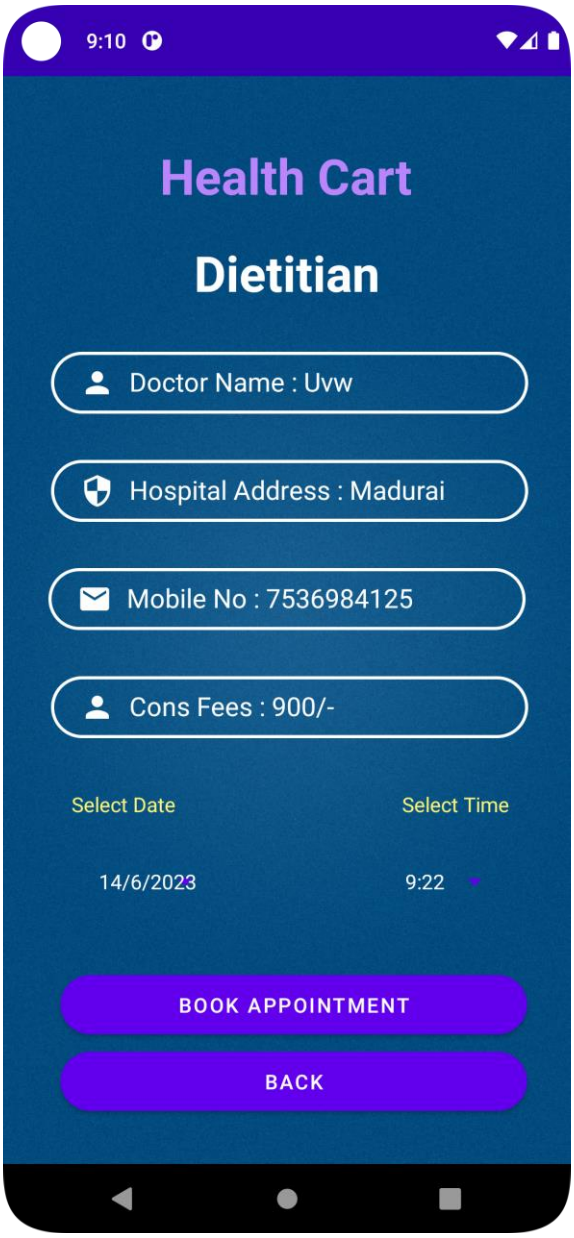
Doctor Name : Cde
Hospital Address : Madurai
Exp : 10yrs Mobile No : 8523691475 Cons Fees:800/-

Doctor Name : Fgh
Hospital Address : Madurai
Exp : 15yrs Mobile No : 7536984125 Cons Fees:1000/-

Doctor Name : Ijk
Hospital Address : Madurai
Exp : 20yrs Mobile No : 9587412364 Cons Fees:1600/-

Doctor Name : Lmn
Hospital Address : Madurai
Exp : 25yrs Mobile No : 6385479125 Cons Fees:2600/-

iii. Doctor Description:



Doctor Name : Uvw

Hospital Address : Madurai

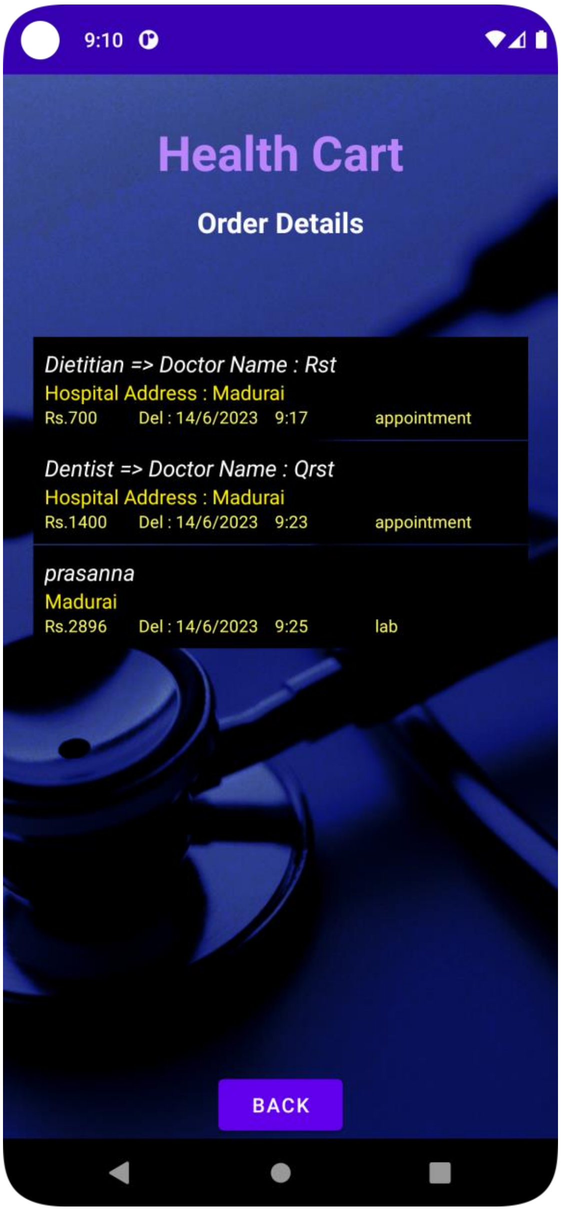
Mobile No : 7536984125

Cons Fees : 900/-

Select Date: 14/6/2023

Select Time: 9:22

6. Order Details Page:



9:10

Health Cart

Order Details

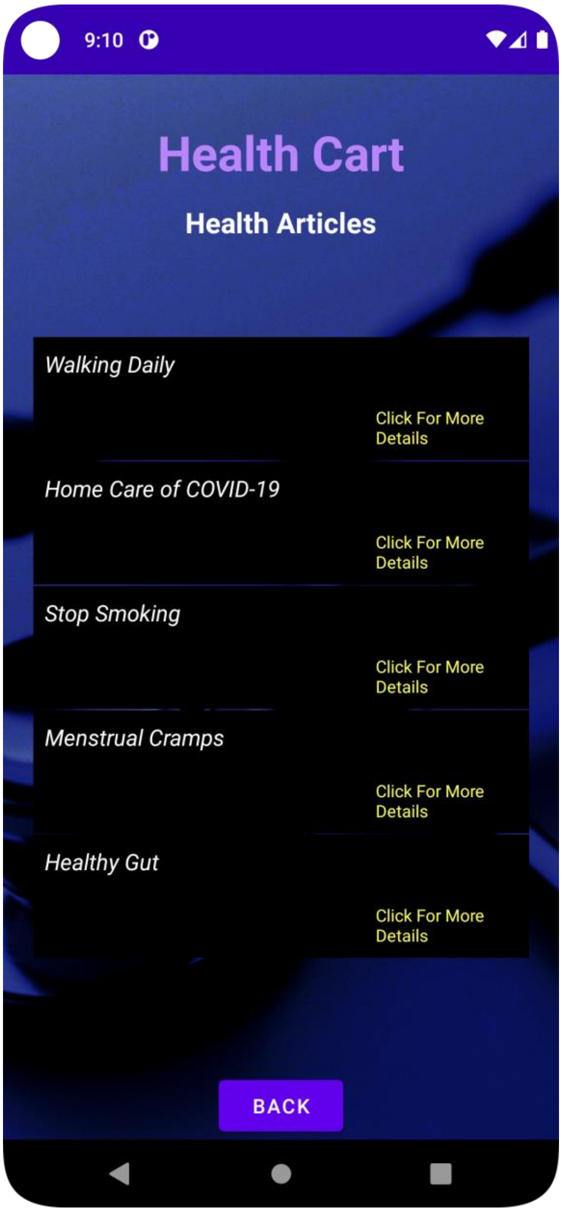
Dietitian => Doctor Name : Rst
Hospital Address : Madurai
Rs.700 Del : 14/6/2023 9:17 appointment

Dentist => Doctor Name : Qrst
Hospital Address : Madurai
Rs.1400 Del : 14/6/2023 9:23 appointment

prasanna
Madurai
Rs.2896 Del : 14/6/2023 9:25 lab

7. Health Articles Page:

i. Article's List:



9:10

Health Cart

Health Articles

Walking Daily
Click For More Details

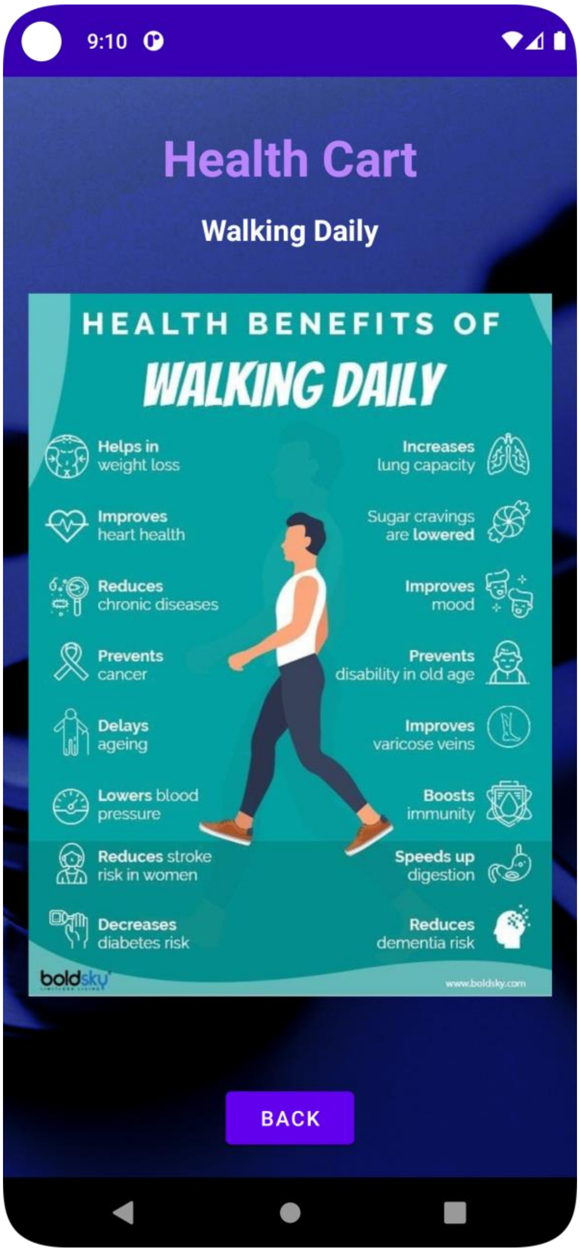
Home Care of COVID-19
Click For More Details

Stop Smoking
Click For More Details

Menstrual Cramps
Click For More Details

Healthy Gut
Click For More Details

ii. Articles Content :



Health Cart

Walking Daily

HEALTH BENEFITS OF

WALKING DAILY

Helps in weight loss

Improves heart health

Reduces chronic diseases

Prevents cancer

Delays ageing

Lowers blood pressure

Reduces stroke risk in women

Decreases diabetes risk

Increases lung capacity

Sugar cravings are lowered

Improves mood

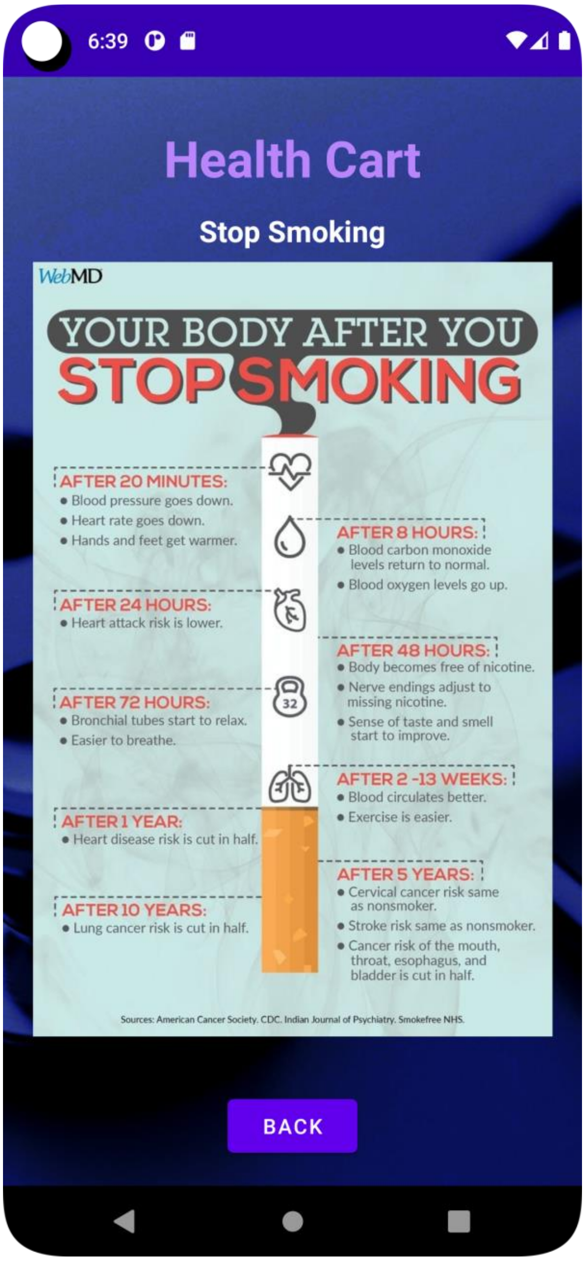
Prevents disability in old age

Improves varicose veins

Boosts immunity

Speeds up digestion

Reduces dementia risk



Health Cart

Stop Smoking

YOUR BODY AFTER YOU STOP SMOKING

AFTER 20 MINUTES:
- Blood pressure goes down.
- Heart rate goes down.
- Hands and feet get warmer.

AFTER 8 HOURS:
- Blood carbon monoxide levels return to normal.
- Blood oxygen levels go up.

AFTER 24 HOURS:
- Heart attack risk is lower.

AFTER 48 HOURS:
- Body becomes free of nicotine.
- Nerve endings adjust to missing nicotine.
- Sense of taste and smell start to improve.

AFTER 72 HOURS:
- Bronchial tubes start to relax.
- Easier to breathe.

AFTER 2 -13 WEEKS:
- Blood circulates better.
- Exercise is easier.

AFTER 1 YEAR:
- Heart disease risk is cut in half.

AFTER 5 YEARS:
- Cervical cancer risk same as nonsmoker.
- Stroke risk same as nonsmoker.
- Cancer risk of the mouth, throat, esophagus, and bladder is cut in half.

AFTER 10 YEARS:
- Lung cancer risk is cut in half.

Sources: American Cancer Society. CDC. Indian Journal of Psychiatry. Smokefree NHS.