SCHOOL OF COMPUTING SCIENCES

DEPARTMENT OF COMPUTER APPLICATIONS

LEARNING APPLICATION

A Mini Project

Submitted for the Partial fulfilment for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS (HONS)

BY

KIRAN VIGNESH RAJ .J

19116114

Under the guidance of

Dr. K. KALAISELVI

Professor

May 2022

INSTITUTE OF SCIENCE, TECHNOLOGY & ADVANCED STUDIES (VISTAS)
(Deemed to be University under section 3 of UGC Act, 1956)
NAAC ACCREDITED WITH 'A' GRADE

BONAFIDE CERTIFICATE

This is to certify that the Capstone Project entitled **LEARNING APPLICATION** is the originalrecord work done by **KIRAN VIGNESH RAJ .J, bearing the register number 19116114.**

Under my guidance and supervision for the partial fulfilment of award of degree of BACHELOR **OF COMPUTER APPLICATIONS (HONS)**, as per syllabus prescribed by the VISTAS.

GUIDE **HEAD OF THE DEPARTMENT**

Submitted for the Viva-Voce examination held on_____at

VISTAS, Pallavaram, Chennai.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I deeply wish to express my sincere thanks to **Dr.ISHARI K.GANESH** M.Com,B.L..,Ph.D.,Chancellor, and Dr. S. Sriman Narayanan the Vice-Chancellor, VISTAS for providing me necessary facilities.

I wish to extend my heartfelt and sincere thanks to **Dr.P.SARAVANAN,** Registrar **Dr.A.UDHAYA KUMAR**, Controller of Examinations, VISTAS

I wish to extend my deep sense of gratitude and sincere thanks to the HEAD OF THE DEPARTMENT, **Dr.S.PRASANNA** MCA.,M.Phil.,Ph.D.

I would like to express my special thanks to my project guide **Dr.K. KALAISELVI,** M.Sc., M.Phil., Ph.D. for her guidance in completing this project

I am using this opportunity to express my sincere gratitude to **Mr.S.GANESH** and **Ms.SANDHYASRI,NIIT.** I express my thanks to the staff members of the department of Computer Applications for their encouragement and co-operation during the completion of my project work.

It is not only my responsibilities, but a deep wish to express my gratitude to my parents and the ALMIGHTY in making this project a grand success. I wish to remember forever the help rendered by my friends for their constant encouragement during the study.

DECLARATION

I affirm that the mini project work titled **"LEARNING APPLICATION"** being submitted in the partial fulfilment for the award of Bachelor of Computer Applications (Hons) is the original work carried out by me. It has not formed the part of any project work submitted for award of any degree or diploma, either in this or any other University.

(Signature of the candidate)
KIRAN VIGNESH RAJ .J

S.NO	CONTENT	PAGE NO
	ABSTRACT	
1	**OBJECTIVE OF THE PROJECT**	**7**
2	**PROBLEM STATEMENT**	**7**
3	**SYSTEM SPECIFICATION** 3.1 HARDWARE REQUIREMENT 3.2 SOFTWARE REQUIREMENT	9
4	**SOFTWARE DESCRIPTION** 4.1 FRONT END 4.2 BACKEND	10
5	**PROJECT DESIGN** 5.1 DATA FLOW DIAGRAM 5.2 NAVIGATION DIAGRAM	13
6	**SYSTEM REQUIREMENT ANALYSIS**	15
7	**SYSTEM TESTING**	16
8	**SYSTEM IMPLEMENTION**	21
9	**APPENDIX** 9.1 SOURCE CODE 9.2 OUTPUT SCREEN SHOT	24
10	**CONCLUSION**	42
11	**WEB REFERENCES**	43

ABSTRACT

The project “learning application” delivers a modern, interactive, and personalized digital storefront for students. Therefore, all brands that wish to have an online presence are to consider a mobile- friendly design or fully responsive applications. The homepage has minimal content yet provides quick navigation to other essential pages using the sticky menu. In particular, a visitor can easily access the application using the e-book, maps and quiz. The logorepresents the brand so this application ensures that it is clearly visible at the center of the page layout.

A learning application includes different functions and functionality which helps the student to find the best e-book with ease. a student can effortlessly study and alter his or her frame of mind to a highly functional and active, interactive standards. The quiz part provides a fun learning environment where a student can learn from his mistakes and it is the more efficient way to understand high end concepts through simple multiple choice questions, especially when social distancing is still in place. This function allows a user to take full advantage of the growing trend. Since this is a standalone application, once installed in a mobile device there is no requirement for the internet and all the features present in the application can be accessed without the function of internet. The map feature makes it much easier to detect the institutions at ease and provides a easier path of direction to reach the certain institution effectively.

The learning application is designed from a user point of view. The user-friendly design helpsthe user in accomplishing their task with ease. Attempts have been made to keep the design simpleand understandable. The total line of code written for this application is JAVA & XML.

1. OBJECTIVE OF THE LEARNING APPLICATION PROJECT

- This application provides transparency, accuracy, and clarity. Good UI is considered to be of first and foremost importance.
- It has a well-structured application with compelling content and awesome features.
- A student can learn a lot from the e-books section and the data is represented in a simple and easy to read format.
- A Quiz section to make learning more efficient and entertaining.

2. PROBLEM STATEMENT

- **NO SOCIAL INTERACTION**

There is no actual interaction between teachers and students, since it is considered as a self-learning application there is no student to student or teacher to student interaction.

That makes the lack of social interaction between people is considered as a major drawback of this application. Social skills are great for teaching kids important social skills. Apps that teach listening, friendship, and even making eye contact are effective options. When you use apps to teach social skills, you make learning more enjoyable experience.

❖ DESKTOP IMCOMPATIBLE

The application is solely made on android studio which makes it incompatible for bigger and wider screens like desktop. The application is not responsive, which means the contents present in the application cannot adapt to the screen size when it varies but it can change according to few mobile devices and API. Software incompatibility is a characteristic of software components or systems which cannot operate satisfactorily together on the same computer, or on different computers linked by a computer network. They may be components or systems which are intended to operate cooperatively or independently. They may be components or systems which are intended to operate cooperatively or independently.

.

❖ STATIC APPLICATION

The application is considered to be static since there is no network added and the pages are not created during the runtime. Every user who registers and logins to this application will have access to the same data and has the same functions and limitations.

If a application has a e-book name, history and local recognition, then that is what will make them stand out froma wide selection of other franchised books. As such, sharing content that sets your company apart from the restis vital.

❖ SOFTWARE ISSUES

A software is an application that runs on a device according to the instructions embedded in the software at the time of coding. Even though it seems like the life of software is smooth, but there are other externa factors that hinder its smooth life span. These external factors are changing trends in the field of IT. Software compatibility issues, not upgrading to a new version, regular system crashes, etc. are some of the issues that hinder the working of the software, thereby interrupting your smooth mobile learning experience.

3. SYSTEM SPECIFICATION

3.1. PC REQUIREMENTS FOR ANDROID STUDIO

Processor: 2nd generation Intel Core or newer, or AMD CPU.

Operating system: 64-bit Microsoft Windows 8/10/11.

RAM: 8 GB or more.

Android studio: Bumblebee (2021.1.1).

Disk space: 8 GB minimum (IDE + Android SDK + Android Emulator).

3.2. MOBILE REQUIREMENTS

Version: Android 8.1 (oreo) or above.

API level: 27 or more.

Resolution: 720 x 1250

4. SOFTWARE DESCRIPTION

Android studio:

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

Android Studio supports all the same programming languages of Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

4.1 FRONT END

Front End Tools : JAVA and XML

JAVA:

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

XML :

Extensible Markup Language (**XML**) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications with all of them free open standards.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures.

The main purpose of XML is serialization, i.e. storing, transmitting, and reconstructing arbitrary data. For two disparate systems to exchange information, they need to agree upon a file format. As a markup language, XML labels, categorizes, and structurally organizes information XML tags represent the data structure and contain metadata. What's within the tags is data, encoded in the way the XML standard specifies. An additional XML schema (XSD) defines the necessary metadata for interpreting and validating XML

4.2 BACK END

Back End Tools: Firebase

FIREBASE:

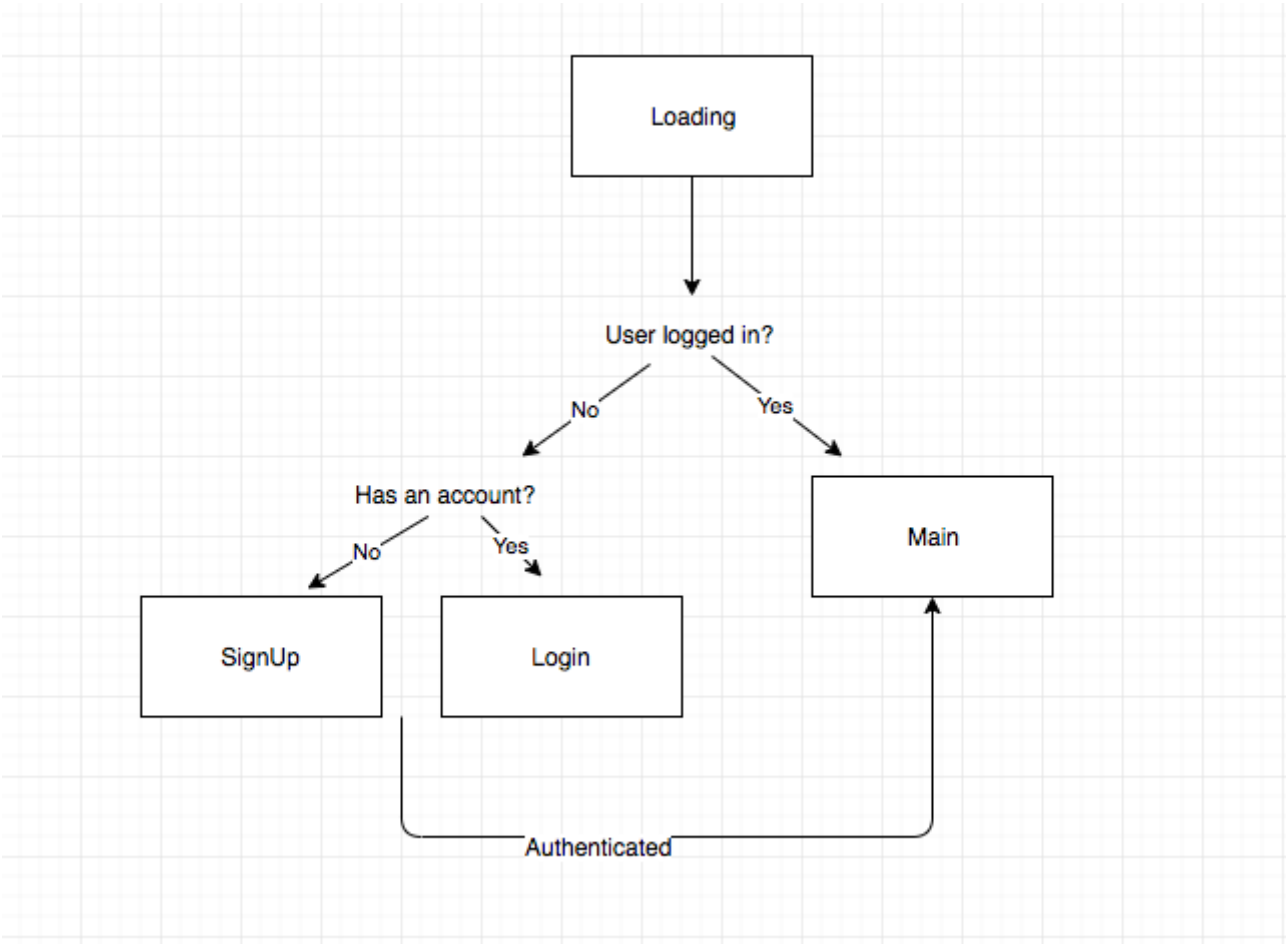
Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app developments of most widely used programming languages,

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in 2011 and it launched to the public in April 2012.

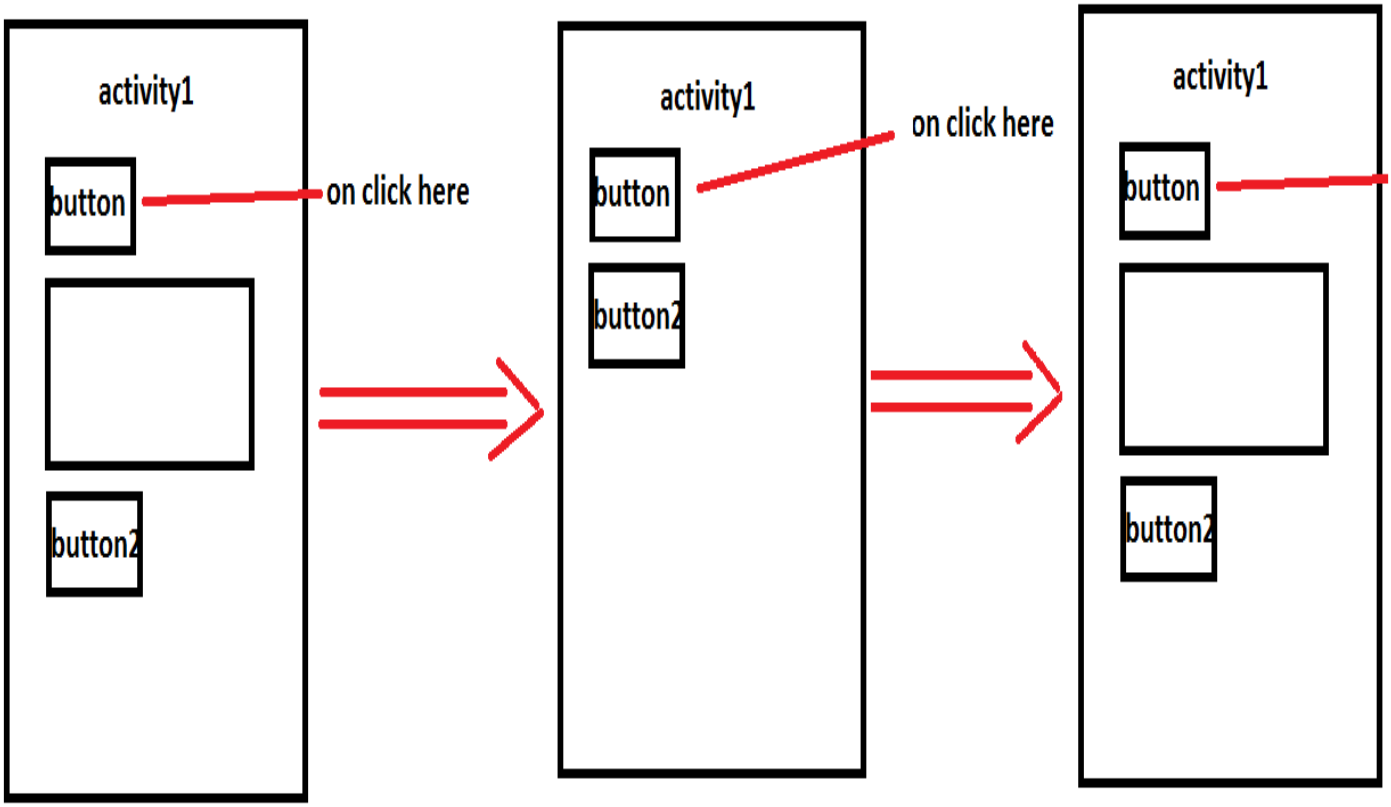
Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 NAVIGATION DIAGRAM



6. SYSTEM REQUIREMENT ANALYSIS

Technical Feasibility –

In Technical Feasibility current resources both hardware software along with required technology are analysed/assessed to develop the project. This technical feasibility study reports whether there exists correct required resources and technologies which will be used for project development. Along with this, the feasibility study also analyses technical skills and capabilities of the technical team, whether existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc.

Operational Feasibility –

In Operational Feasibility the degree of providing service to requirements is analyzed along with how easy the product will be to operate and maintain after deployment. Along with this other operational scopes are determining usability of product, Determining suggested solution by software development team is acceptable or not etc.

Economic Feasibility –

In the Economic Feasibility study the cost and benefit of the project is analyzed. Means under this feasibility study a detailed analysis is carried out of what will be the cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on. After that it is analyzed whether the project will be beneficial in terms of finance for the organization or not.

Legal Feasibility –

The Legal Feasibility study project is analyzed from a legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall it can be said that a Legal Feasibility Study is to know if proposed projects conform to legal and ethical requirements.

Schedule Feasibility –

In Schedule Feasibility Study mainly timelines/deadlines is analyzed for proposed project which includes how many times teams will take to complete final project which has a great impact on the organization as purpose of project may fail if it can't be completed on time

7. SYSTEM TESTING

System Testing is an important stage in any system development life cycle. Testing is a process of executing a program with the intention of finding errors. The importance of software testing and its implications with respect to software quality cannot be over Emphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. A good test case is one that has a high probability of finding a yet undiscovered error.

Testing is the set of activities that can be planned in advance and conducted systematically. Different test conditions should be thoroughly checked and the bugs detected should be fixed. The testing strategies formed by the user are performed to prove that the software is free and clear from errors. To do this, there are many ways of testing the system's reliability, completeness and maintainability.

7.1. LEVELS OF TESTING:

The different types of testing are as follows:

Unit Testing:

- In the unit testing the analyst tests the program making up a system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system, many modules on different levels are needed.

- Unit testing can be performed from the bottom up starting with the smallest and lowest level modules and proceeding one at time. For each module in a bottom-up testing, a short program executes the module and provides the needed data.

Integration Testing:

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. Objectives are used to take unit test modules and build program structure that has been directed by design. The integration testing is performed for this Hospital Management System when all the modules were to make it a complete system. After integration the project works successfully.

Black Box Testing:

This method treats the coded module as a black box. The module runs with inputs that are likely to cause errors. Then the output is checked to see if any error occurred. This method cannot be used to test all errors, because some errors may depend on the code or algorithm used to implement the module.

White Box Testing

White box testing, sometimes called glass-box testing, is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

Guarantee that all independent paths within a module have been exercised at least once. Exercise all logical decisions on their true and false side

- Execute all loops at their boundaries and within their operational bounds ➢

Exercise internal data structure to assure their validity.

For example in this project white box testing is performed in the patient module. Without entering text if we apply it displays the message “First add record then save it” else it should be saved

Verification testing

Testing the system with the intent of confirming readiness of the product and customer acceptance.

Test	Test Area	Verification Result
Verification testing	Interaction of users with the system	The user should be able to use the system with ease.

Validation testing

Validation testing can be defined in many ways, but a simple definition is that can be reasonably expected by the customer. After the validation test has been conducted, one of two possible conditions exists.

- The functions or performance characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered and a deficiency list is created.
- Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

For example, in this project validation testing is performed against inpatient search module. This module is tested with the following valid and invalid inputs for the field patient name.

User Acceptance Testing

Test	Test area	Expected results
Module testing	Adding new record	System should be able to add new records to the database
	Delete records	The system should be able to delete unwanted records
	Update records	The system should update records and save changes made.
	Search records	The system should be able to retrieve relevant records required by the user.

7.2.Why System Testing?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in three types of problems:

1. The time lag between the cause and the appearance of the problem. 2. The effect of system errors on the files and records within the system. 3. Another reason for system testing is its utility as a user-oriented vehicle before implementation.

Activity Network for System Testing

1. Prepare a test plan.

2. Specify conditions for user acceptance testing.

3. Prepare test data for program testing.

4. Prepare test data for transaction path testing.

5. Plan user training.

6. Compile/assemble programs.

7. Prepare job performance aids.

8. Prepare operational documents.

Prepare Test

A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

• Outputs expected from the system.

• Criteria for evaluating outputs.

• Procedure for using test data.

• Personnel and training requirements.

Specify Conditions for User Acceptance Testing

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test

.

Prepare Test Data for Program Testing

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

Prepare Test Data for Transaction Path Testing

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

8. SYSTEM IMPLEMENTATION

Implementation of a new computer system to replace an existing one. This is usually difficult conversion. If not properly planned, there can be many problems. So large computer systems may take as long as a year to convert.

Implementation of a modified application to replace the existing one using the same computer. This type of conversion is relatively easy to handle, usually there are no major changes in the file.

The process of implementing software is much more difficult as compared to the task of creating the project. First we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale. Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements.

8.1. HARDWARE EVALUATION FACTORS:

When we evaluate computer hardware, we should first investigate specific physical and performance characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These hardware evaluation factors questions are summarized in the below figure.

.

HARDWARE EVALUATION FACTORS:

- Performance
- Cost
- Reliability
- Availability
- Compatibility
- Modularity
- Technology
- Ergonomics
- Connectivity
- Environmental requirements
- Software
- Support

8.2. SYSTEM MAINTENANCE

Security

Every member of staff of the hospital requires a username and password to log on to the system. The administrator of this system registers each member staff allotting username and password to each and he/she can also revoke access if it is deemed fit for any reason. The data of data-base are protected through multiple layers of security which includes but not limited to pass-words which are encrypted (should the hospital decide to take the software online with net-working available but since this is a stand-alone software, the password is not

encrypted) but each member is required to protect their password and change on the first logging when created by the administrator.

Performance requirement

For any software developed in this modern time, one of the most important things to do regularly is to update and upgrade and fix whatever bugs are found. The following are the list of maintenance required for this software: ➢ Database archiving

- ➢ Password encryption
- ➢ Anti-virus protection.
- ➢ Password update every 72 days

8.3. Error handling

The system also has error and debugging code within the software to prevent system collapse and krypton 2 professional encryption programs are embedded within the software security layer to prevent hacking it when installed over an internet network. Also, within the code itself, SQL inject susceptible characters had been cleaned up.

Installation

After developing and testing the software, the next thing is to deploy the software and run it. For the purpose of this thesis, the installation and deployment of the software will be done from a compact disc (CD) from which the software has been copied to and will be run on a window platform.

9. APPENDIX

Code Efficiency:

Reviewing of Code efficiency for a module is carried out after the module is successfully compiled and all the syntax errors eliminated. Code efficiency review is an extremely cost-effective strategy for reduction in coding errors in order to produce high quality code. Normally, two types of efficiency are carried out on the code of a module - code optimization and code inspection. The procedure and final objective of these two efficiency techniques are very different as discussed below.

Optimization of Code:

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated. Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in an optimization meeting where the coder of the module is also present.

9.1. SOURCE CODE:

MainActivity.java

```
package com.example.c9mk5;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.view.View;
import android.widget.ImageView;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.view.WindowManager;

import com.example.c9mk5.HelperClasses.HomeAdapter.FeaturedAdapter;
import com.example.c9mk5.HelperClasses.HomeAdapter.FeaturedHelperClass;
import com.example.location.MapsActivity;

import java.util.ArrayList;

public class DashTheFlash extends AppCompatActivity {

    RecyclerView featuredRecycler;
    RecyclerView.Adapter adapter;
    ImageView imageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG
_FULLSCREEN);
        setContentView(R.layout.activity_dash_the_flash);

        //hooks
        featuredRecycler = findViewById(R.id.featured_recycler);

        featuredRecycler();


    }

    private void featuredRecycler(){
        featuredRecycler.setHasFixedSize(true);
        featuredRecycler.setLayoutManager(new
LinearLayoutManager(this,LinearLayoutManager.HORIZONTAL,false));

        ArrayList<FeaturedHelperClass> featuredcourses = new ArrayList<>();
        featuredcourses.add(new FeaturedHelperClass(R.drawable.java1,"JAVA", "Java programming
language is official language for mobile application development"));
        featuredcourses.add(new FeaturedHelperClass(R.drawable.php,"PHP", "PHP programming
```

```
language is official language for mobile application development"));
        featuredcourses.add(new FeaturedHelperClass(R.drawable.android,"Android", "Android
programming language is official language for mobile application development"));

        adapter = new FeaturedAdapter(featuredcourses);
        featuredRecycler.setAdapter(adapter);

        GradientDrawable gradient1 = new
GradientDrawable(GradientDrawable.Orientation.LEFT_RIGHT, new int[]{0xffeff400, 0xffaff600});

        imageView = findViewById(R.id.map_btn);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(DashTheFlash.this, MapsActivity.class);
                startActivity(intent);
            }

    });

        imageView = findViewById(R.id.ho_btn);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(DashTheFlash.this, DashTheFlash.class);
                startActivity(intent);
            }

        });

        imageView = findViewById(R.id.ebay_btn);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(DashTheFlash.this, bkbeg1.class);
                startActivity(intent);
            }

        });

        imageView = findViewById(R.id.quiz_btn);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(DashTheFlash.this, DashTheFlash.class);
                startActivity(intent);
            }

        });

        imageView = findViewById(R.id.call_btn);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(DashTheFlash.this, DashTheFlash.class);
                startActivity(intent);
                });
```

MainActivity.java (quiz fragment)

```
package com.learnoset.offlinequizapp;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;

import java.util.List;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

    // total quiz time in minutes
    private int totalTimeInMins = 1;

    // Timer class object for countdown timer
    private Timer quizTimer;
    private int seconds = 0; // current countdown seconds

    // questions array list
    private List<QuestionsList> questionsLists;

    // Current questions index position from  questionsLists ArrayList.
    private int currentQuestionPosition = 0;

    // Options
    private AppCompatButton option1, option2, option3, option4;

    // next button
    private AppCompatButton nextBtn;

    // Total questions and main question TextView
    private TextView question;
    private TextView questions;

    // selectedOption's Value. if user not selected any option yet then it is empty by default
    private String selectedOptionByUser = "";

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

        // initialize widgets from activity_main.xml file
        final ImageView backBtn = findViewById(R.id.backBtn);
        final TextView topicName = findViewById(R.id.topicName);
        final TextView timer = findViewById(R.id.timer);
```

```
        question = findViewById(R.id.question);
        questions = findViewById(R.id.questions);
        option1 = findViewById(R.id.option1);
        option2 = findViewById(R.id.option2);
        option3 = findViewById(R.id.option3);
        option4 = findViewById(R.id.option4);
        nextBtn = findViewById(R.id.nextButton);

        // get Topic Name and User Name from StartActivity via Intent
        final String getTopicName = getIntent().getStringExtra("selectedTopic");

        // set Topic Name to TextView
        topicName.setText(getTopicName);

        // start quiz countdown timer
        startTimer(timer);

        // get questions from QuestionsBank class according to selectedTopicName and assign to
questionsLists ArrayList
        questionsLists = QuestionsBank.getQuestions(getTopicName);

        // set current question to TextView along with options from questionsLists ArrayList
        questions.setText((currentQuestionPosition + 1) + "/" + questionsLists.size());
        question.setText(questionsLists.get(currentQuestionPosition).getQuestion());
        option1.setText(questionsLists.get(currentQuestionPosition).getOption1());
        option2.setText(questionsLists.get(currentQuestionPosition).getOption2());
        option3.setText(questionsLists.get(currentQuestionPosition).getOption3());
        option4.setText(questionsLists.get(currentQuestionPosition).getOption4());

        option1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // check if user has not attempted this question yet
                if (selectedOptionByUser.isEmpty()) {

                    selectedOptionByUser = option1.getText().toString();

                    // change selected AppCompatButton background color and text color
                    option1.setBackgroundResource(R.drawable.round_back_red10);
                    option1.setTextColor(Color.WHITE);

                    // reveal answer
                    revealAnswer();

                    // assign user answer value to userSelectedOption in QuestionsList class

questionsLists.get(currentQuestionPosition).setUserSelectedOption(selectedOptionByUser);
                }
            }
        });

        option2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // check if user has not attempted this question yet
                if (selectedOptionByUser.isEmpty()) {
```

```
                    selectedOptionByUser = option2.getText().toString();

                    // change selected AppCompatButton background color and text color
                    option2.setBackgroundResource(R.drawable.round_back_red10);
                    option2.setTextColor(Color.WHITE);

                    // reveal answer
                    revealAnswer();

                    // assign user answer value to userSelectedOption in QuestionsList class

questionsLists.get(currentQuestionPosition).setUserSelectedOption(selectedOptionByUser);
                }
            }
        });

        option3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // check if user has not attempted this question yet
                if (selectedOptionByUser.isEmpty()) {

                    selectedOptionByUser = option3.getText().toString();

                    // change selected AppCompatButton background color and text color
                    option3.setBackgroundResource(R.drawable.round_back_red10);
                    option3.setTextColor(Color.WHITE);

                    // reveal answer
                    revealAnswer();

                    // assign user answer value to userSelectedOption in QuestionsList class

questionsLists.get(currentQuestionPosition).setUserSelectedOption(selectedOptionByUser);
                }
            }
        });

        option4.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // check if user has not attempted this question yet
                if (selectedOptionByUser.isEmpty()) {

                    selectedOptionByUser = option4.getText().toString();

                    // change selected AppCompatButton background color and text color
                    option4.setBackgroundResource(R.drawable.round_back_red10);
                    option4.setTextColor(Color.WHITE);

                    // reveal answer
                    revealAnswer();

                    // assign user answer value to userSelectedOption in QuestionsList class
```

```
questionsLists.get(currentQuestionPosition).setUserSelectedOption(selectedOptionByUser);
                }
            }
        });

        backBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // cancel timer
                quizTimer.purge();
                quizTimer.cancel();

                // open StartActivity.java
                startActivity(new Intent(MainActivity.this, StartActivity.class));
                finish(); // finish(destroy) this activity
            }
        });

        nextBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // check if user has not selected any option yet
                if (selectedOptionByUser.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please select an option",
Toast.LENGTH_SHORT).show();
                } else {
                    changeNextQuestion();
                }
            }
        });
    }

    private void startTimer(TextView timerTextView) {

        quizTimer = new Timer();
        quizTimer.schedule(new TimerTask() {
            @Override
            public void run() {

                if (seconds == 0) {
                    totalTimeInMins--;
                    seconds = 59;
                } else if (seconds == 0 && totalTimeInMins == 0) {
                    // cancel timer
                    quizTimer.purge();
                    quizTimer.cancel();
                    Toast.makeText(MainActivity.this, "Timer Over", Toast.LENGTH_SHORT).show();

                    // Open result activity along with correct and incorrect answers
                    Intent intent = new Intent(MainActivity.this, QuizResults.class);
                    intent.putExtra("correct", getCorrectAnswers());
                    intent.putExtra("incorrect", getIncorrectAnswers());
                    startActivity(intent);

                    // finish(destroy) this activity
                    finish();
```

```
                } else {
                    seconds--;
                }

                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {

                        String finalMinutes = String.valueOf(totalTimeInMins);
                        String finalSeconds = String.valueOf(seconds);

                        // check if minutes has only one digit(Ex. 9) then attach 0 before the
digit to make it 09
                        if (finalMinutes.length() == 1) {
                            finalMinutes = "0" + finalMinutes;
                        }

                        // check if seconds has only one digit(Ex. 9) then attach 0 before the
digit to make it 09
                        if (finalSeconds.length() == 1) {
                            finalSeconds = "0" + finalSeconds;
                        }

                        timerTextView.setText(finalMinutes + ":" + finalSeconds);
                    }
                });
            }
        }, 1000, 1000);
    }

    private void revealAnswer() {

        // get answer of current question
        final String getAnswer = questionsLists.get(currentQuestionPosition).getAnswer();

        // change background color and text color of option which match with answer
        if (option1.getText().toString().equals(getAnswer)) {
            option1.setBackgroundResource(R.drawable.round_back_green10);
            option1.setTextColor(Color.WHITE);
        } else if (option2.getText().toString().equals(getAnswer)) {
            option2.setBackgroundResource(R.drawable.round_back_green10);
            option2.setTextColor(Color.WHITE);
        } else if (option3.getText().toString().equals(getAnswer)) {
            option3.setBackgroundResource(R.drawable.round_back_green10);
            option3.setTextColor(Color.WHITE);
        } else if (option4.getText().toString().equals(getAnswer)) {
            option4.setBackgroundResource(R.drawable.round_back_green10);
            option4.setTextColor(Color.WHITE);
        }
    }

    private void changeNextQuestion() {

        // increment currentQuestionPosition by 1 for next question
        currentQuestionPosition++;

        // change next button text to submit if it is last question
        if ((currentQuestionPosition + 1) == questionsLists.size()) {
```

```
            nextBtn.setText("Submit Quiz");
        }

        // check if next question is available. else quiz completed
        if (currentQuestionPosition < questionsLists.size()) {

            // make selectedOptionByUser empty to hold next question's answer
            selectedOptionByUser = "";

            // set normal background color and text color to options
            option1.setBackgroundResource(R.drawable.round_back_white_stroke2_10);
            option1.setTextColor(Color.parseColor("#1F6BB8"));
            option2.setBackgroundResource(R.drawable.round_back_white_stroke2_10);
            option2.setTextColor(Color.parseColor("#1F6BB8"));
            option3.setBackgroundResource(R.drawable.round_back_white_stroke2_10);
            option3.setTextColor(Color.parseColor("#1F6BB8"));
            option4.setBackgroundResource(R.drawable.round_back_white_stroke2_10);
            option4.setTextColor(Color.parseColor("#1F6BB8"));

            // set current question to TextView along with options from questionsLists
ArrayList
            questions.setText((currentQuestionPosition + 1) + "/" + questionsLists.size());
            question.setText(questionsLists.get(currentQuestionPosition).getQuestion());
            option1.setText(questionsLists.get(currentQuestionPosition).getOption1());
            option2.setText(questionsLists.get(currentQuestionPosition).getOption2());
            option3.setText(questionsLists.get(currentQuestionPosition).getOption3());
            option4.setText(questionsLists.get(currentQuestionPosition).getOption4());
        } else {

            // Open result activity along with correct and incorrect answers
            Intent intent = new Intent(MainActivity.this, QuizResults.class);
            intent.putExtra("correct", getCorrectAnswers());
            intent.putExtra("incorrect", getIncorrectAnswers());
            startActivity(intent);

            // finish(destroy) this activity
            finish();
        }
    }

    private int getCorrectAnswers() {

        int correctAnswers = 0;

        for (int i = 0; i < questionsLists.size(); i++) {
            final String getUserSelectedOption = questionsLists.get(i).getUserSelectedOption();
            final String getAnswer = questionsLists.get(i).getAnswer();

            // compare user selected option with original answer
            if (getUserSelectedOption.equals(getAnswer)) {
                correctAnswers++;
            }
        }
        return correctAnswers;
    }

    private int getIncorrectAnswers() {
```

```
    int incorrectAnswers = 0;

    for (int i = 0; i < questionsLists.size(); i++) {
        final String getUserSelectedOption = questionsLists.get(i).getUserSelectedOption();
        final String getAnswer = questionsLists.get(i).getAnswer();

        // compare user selected option with original answer
        if (!getUserSelectedOption.equals(getAnswer)) {
            incorrectAnswers++;
        }
    }
    return incorrectAnswers;
}

@Override
public void onBackPressed() {
    // cancel timer
    quizTimer.purge();
    quizTimer.cancel();

    // open StartActivity.java
    startActivity(new Intent(MainActivity.this, StartActivity.class));
    finish(); // finish(destroy) this activity
}
```

LoginActiivity.java

```
package com.example.c9mk5;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap;

public class LoginActivity extends AppCompatActivity {

    private EditText email, password, name;
    private Button mlogin;
    private TextView newdnewaccount, reocverpass;
    FirebaseUser currentUser;
    private ProgressDialog loadingBar;
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        ActionBar actionBar = getSupportActionBar();
        actionBar.setTitle("Create Account");
        actionBar.setDisplayShowHomeEnabled(true);
        actionBar.setDisplayHomeAsUpEnabled(true);

        // initialising the layout items
        email = findViewById(R.id.login_email);
        password = findViewById(R.id.login_password);
```

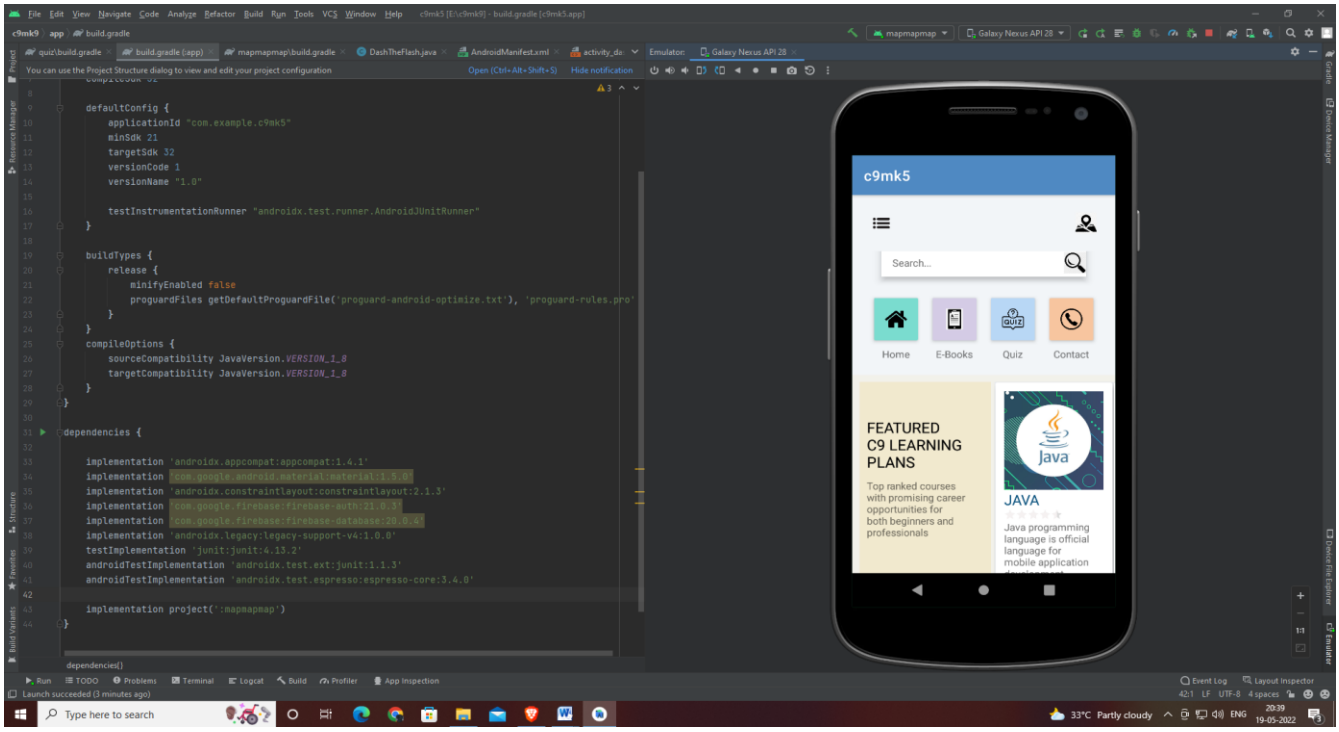
```
    newdnewaccount = findViewById(R.id.needs_new_account);
    reocverpass = findViewById(R.id.forgetp);
    mAuth = FirebaseAuth.getInstance();
    mlogin = findViewById(R.id.login_button);
    loadingBar = new ProgressDialog(this);
    mAuth = FirebaseAuth.getInstance();

    // checking if user is null or not
    if (mAuth != null) {
        currentUser = mAuth.getCurrentUser();
    }

    mlogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String emaill = email.getText().toString().trim();
            String pass = password.getText().toString().trim();

            // if format of email doesn't matches return null
            if (!Patterns.EMAIL_ADDRESS.matcher(emaill).matches()) {
                email.setError("Invalid Email");
                email.setFocusable(true);

            } else {
                loginUser(emaill, pass);
            }
        }
    });

    // If new account then move to Registration Activity
    newdnewaccount.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(LoginActivity.this, RegistrationActivity.class));
        }
    });

    // Recover Your Password using email
    reocverpass.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showRecoverPasswordDialog();
        }
    });
}

private void showRecoverPasswordDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Recover Password");
    LinearLayout linearLayout = new LinearLayout(this);
    final EditText emailet = new EditText(this);//write your registered email
    emailet.setText("Email");
    emailet.setMinEms(16);
    emailet.setInputType(InputType.TYPE_TEXT_VARIATION_EMAIL_ADDRESS);
    linearLayout.addView(emailet);
    linearLayout.setPadding(10, 10, 10, 10);
    builder.setView(linearLayout);
```

```
        builder.setPositiveButton("Recover", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                String emaill = emailet.getText().toString().trim();
                beginRecovery(emaill);//send a mail on the mail to recover password
            }
        });
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });
        builder.create().show();
    }

    private void beginRecovery(String emaill) {
        loadingBar.setMessage("Sending Email....");
        loadingBar.setCanceledOnTouchOutside(false);
        loadingBar.show();

        // send reset password email
        mAuth.sendPasswordResetEmail(emaill).addOnCompleteListener(new
OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                loadingBar.dismiss();
                if (task.isSuccessful()) {
                    Toast.makeText(LoginActivity.this, "Done sent", Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(LoginActivity.this, "Error Occured",
Toast.LENGTH_LONG).show();
                }
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                loadingBar.dismiss();
                Toast.makeText(LoginActivity.this, "Error Failed", Toast.LENGTH_LONG).show();
            }
        });
    }

    private void loginUser(String emaill, String pass) {
        loadingBar.setMessage("Logging In....");
        loadingBar.show();

        // sign in with email and password after authenticating
        mAuth.signInWithEmailAndPassword(emaill, pass).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if (task.isSuccessful()) {
```

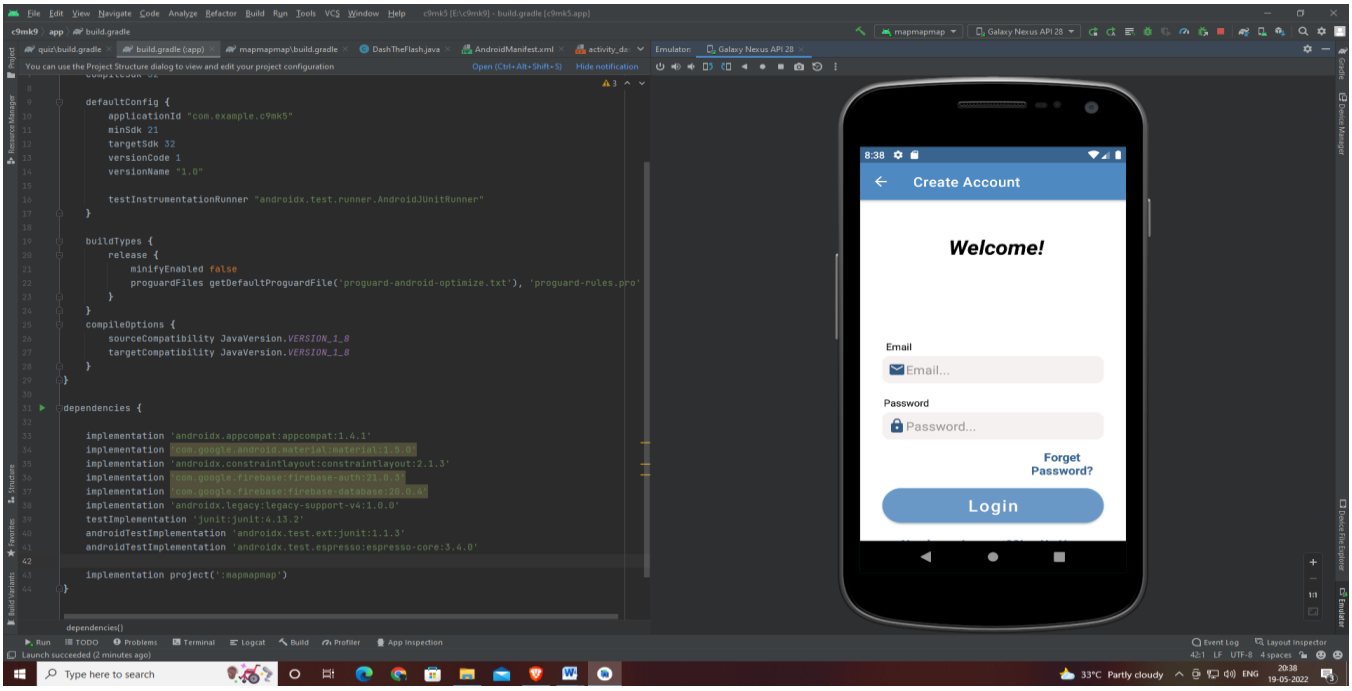
```
                    loadingBar.dismiss();
                    FirebaseUser user = mAuth.getCurrentUser();

                    if (task.getResult().getAdditionalUserInfo().isNewUser()) {
                        String email = user.getEmail();
                        String uid = user.getUid();
                        HashMap<Object, String> hashMap = new HashMap<>();
                        hashMap.put("email", email);
                        hashMap.put("uid", uid);
                        hashMap.put("name", "");
                        hashMap.put("onlineStatus", "online");
                        hashMap.put("typingTo", "noOne");
                        hashMap.put("phone", "");
                        hashMap.put("image", "");
                        hashMap.put("cover", "");
                        FirebaseDatabase database = FirebaseDatabase.getInstance();

                        // store the value in Database in "Users" Node
                        DatabaseReference reference = database.getReference("Users");

                        // storing the value in Firebase
                        reference.child(uid).setValue(hashMap);
                    }
                    Toast.makeText(LoginActivity.this, "Registered User " + user.getEmail(),
Toast.LENGTH_LONG).show();
                    Intent mainIntent = new Intent(LoginActivity.this, DashTheFlash.class);
                    mainIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(mainIntent);
                    finish();
                } else {
                    loadingBar.dismiss();
                    Toast.makeText(LoginActivity.this, "Login Failed",
Toast.LENGTH_LONG).show();
                }
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                loadingBar.dismiss();
                Toast.makeText(LoginActivity.this, "Error Occured", Toast.LENGTH_LONG).show();
            }
        });
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }
}
```

Screenshots:

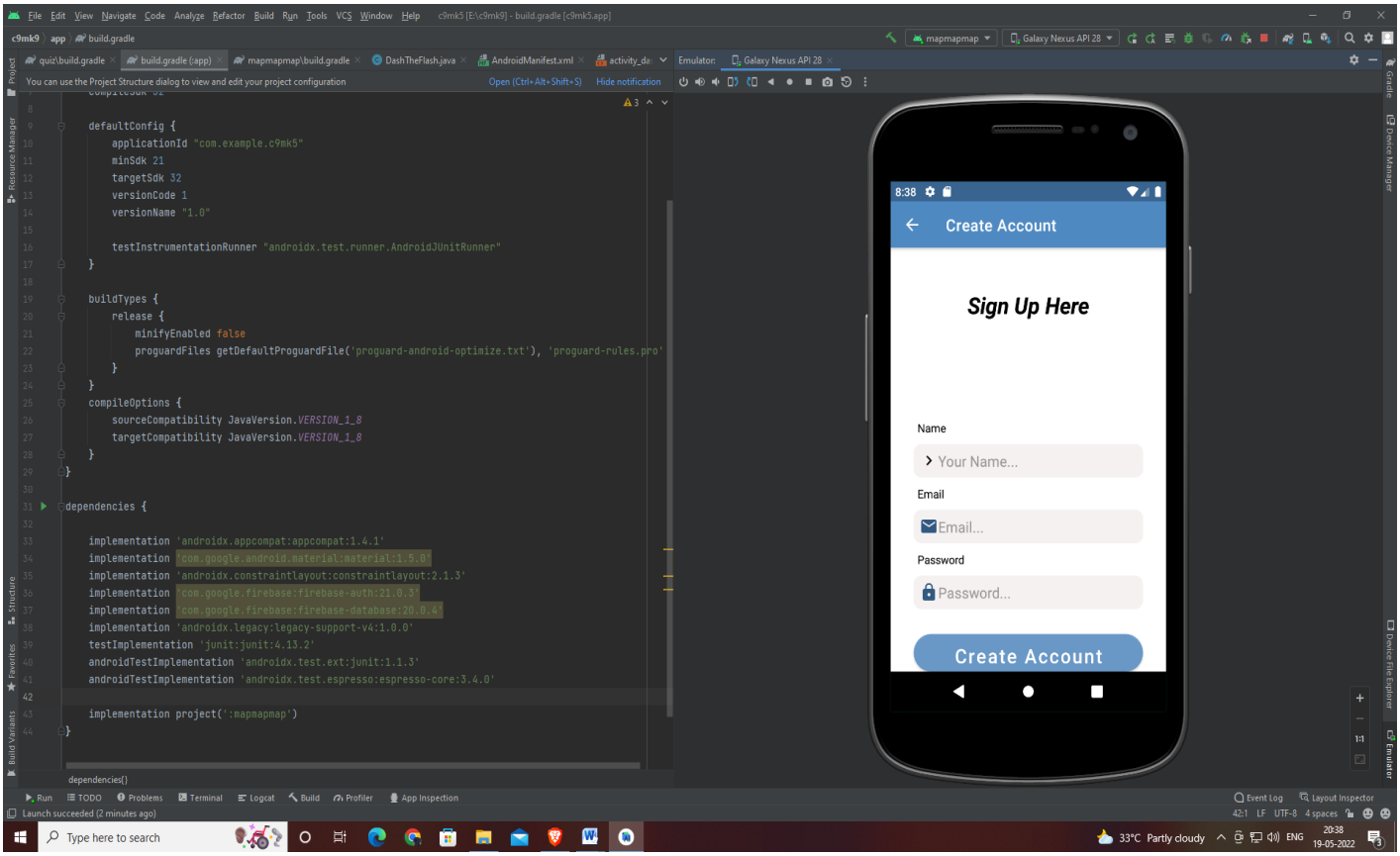
Home page



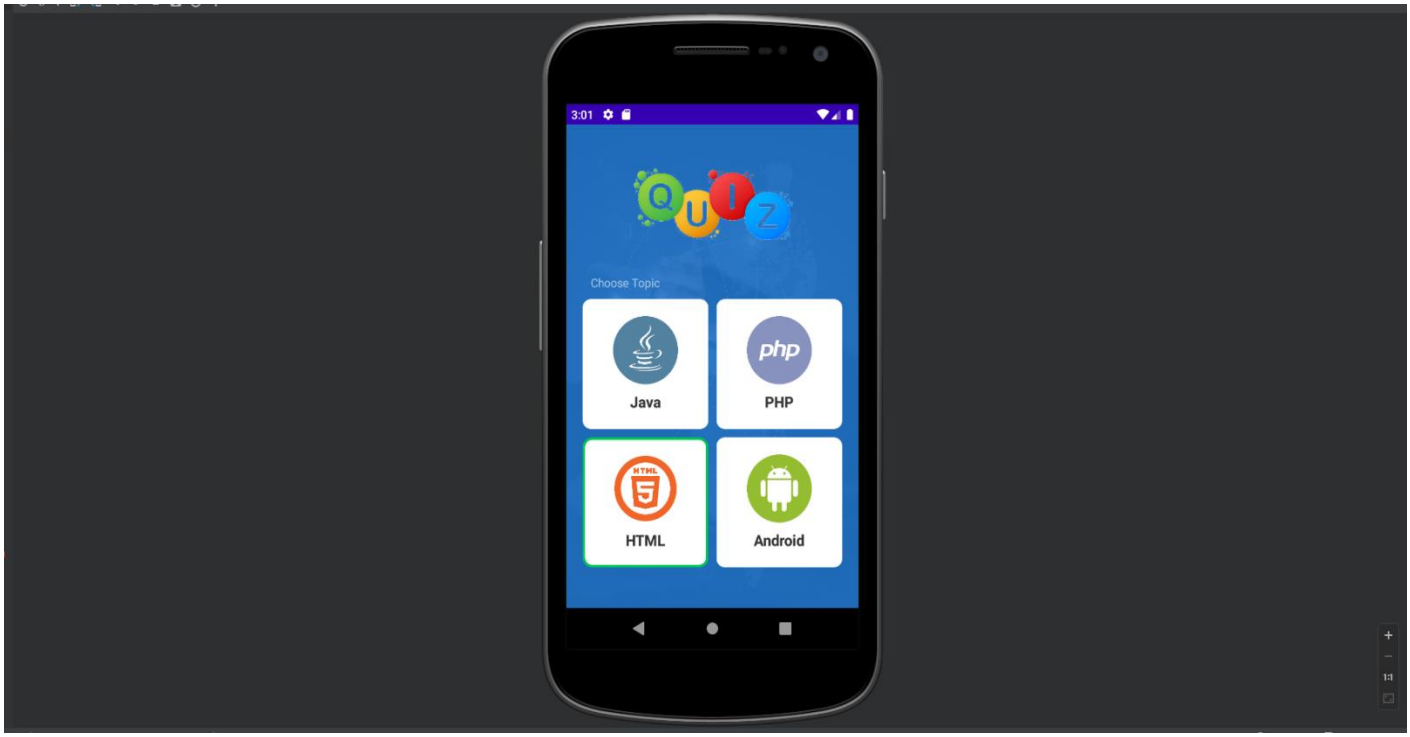
Login page:



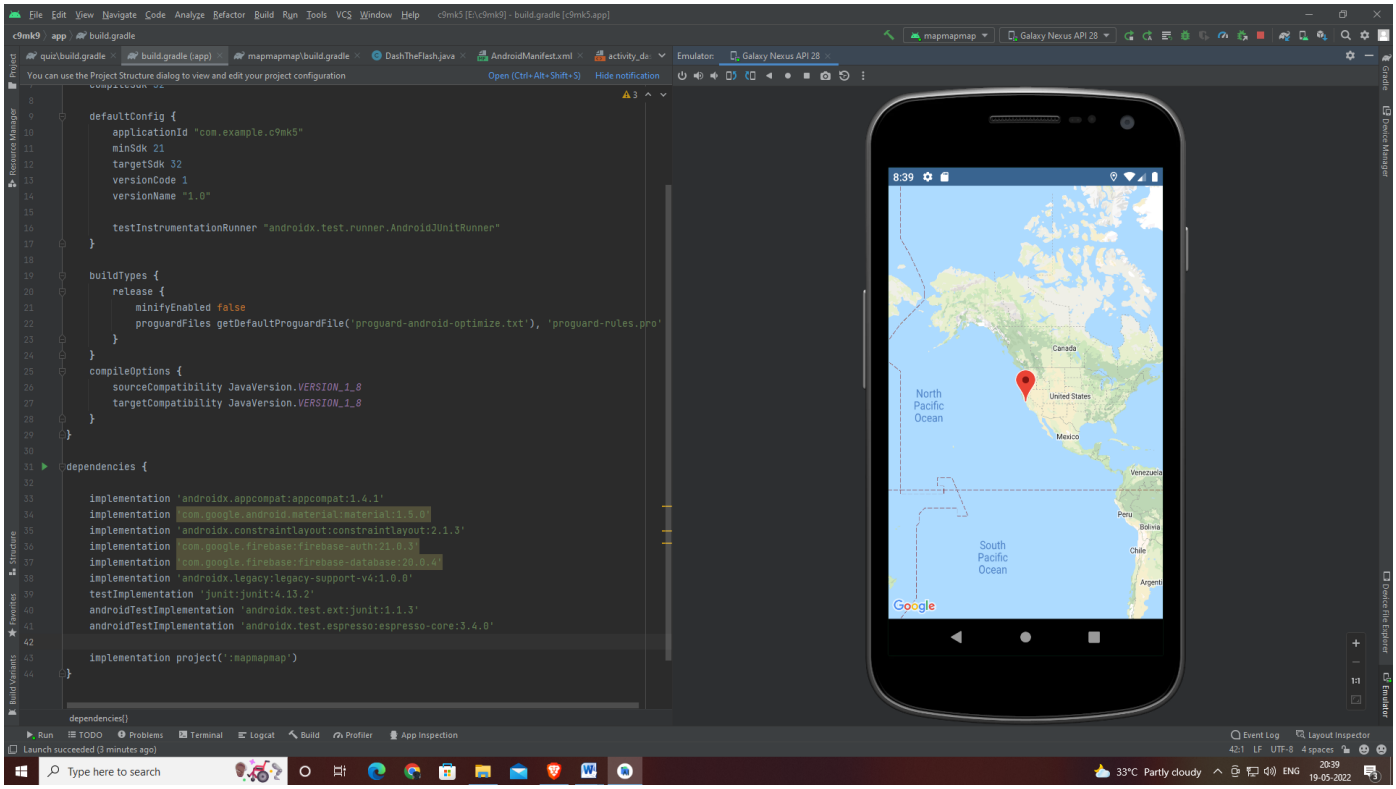
Registration page:



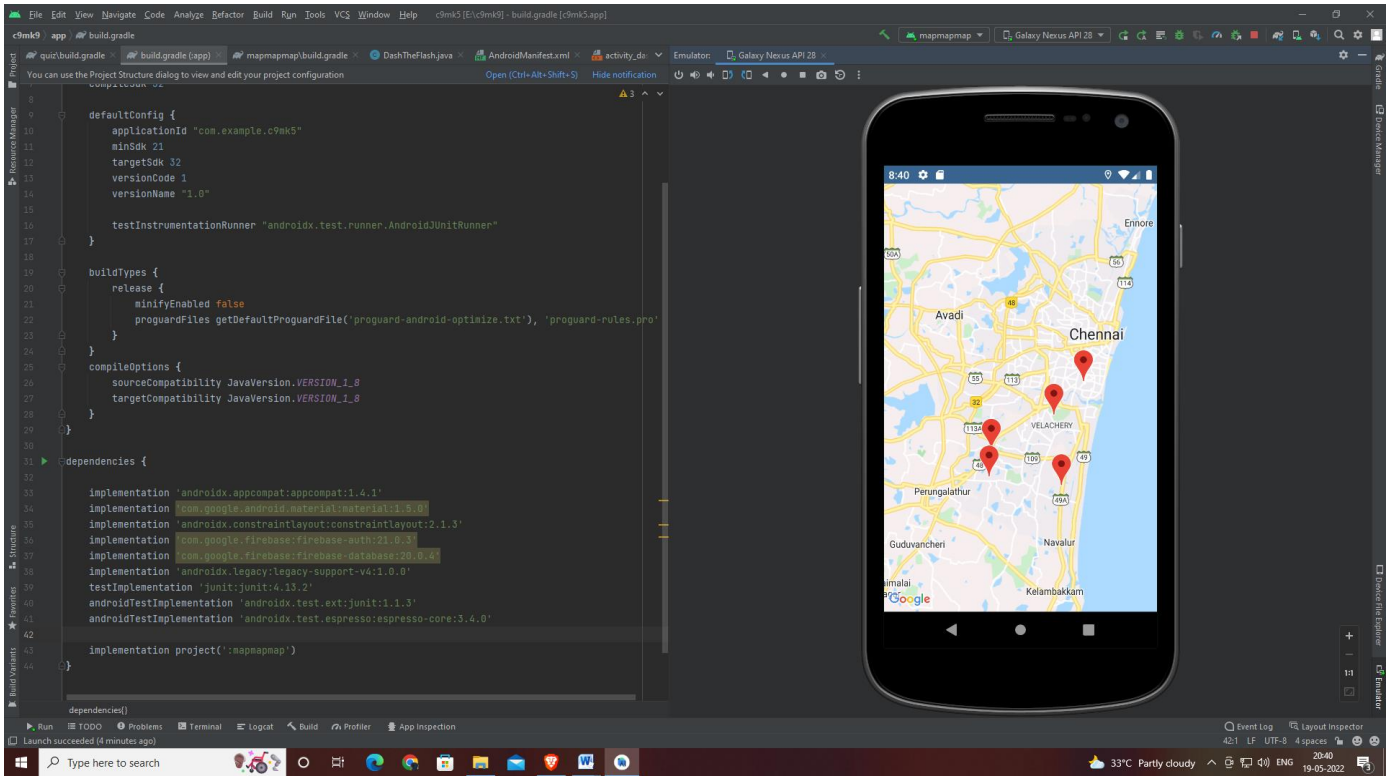
Quiz page:



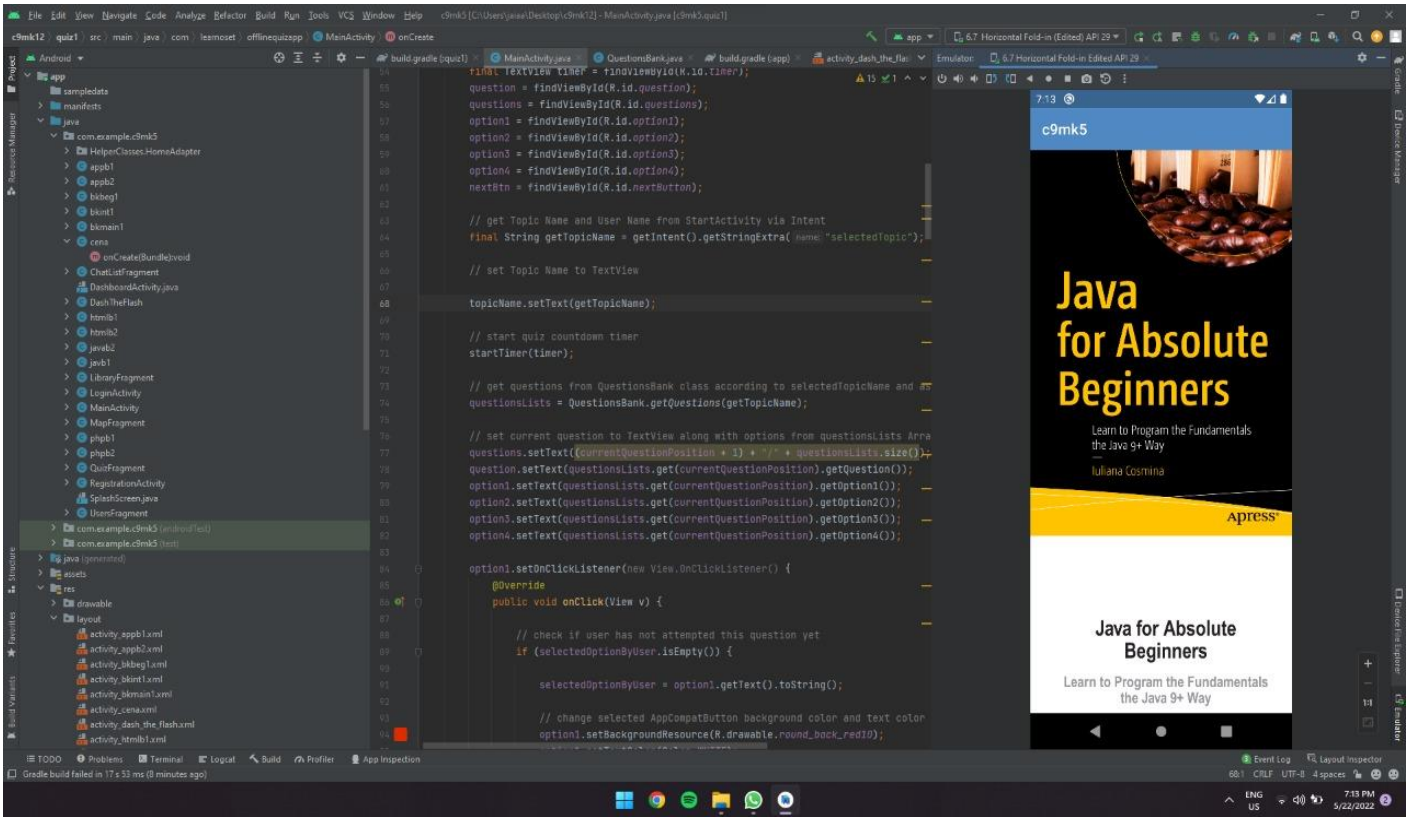
Map Fragment:



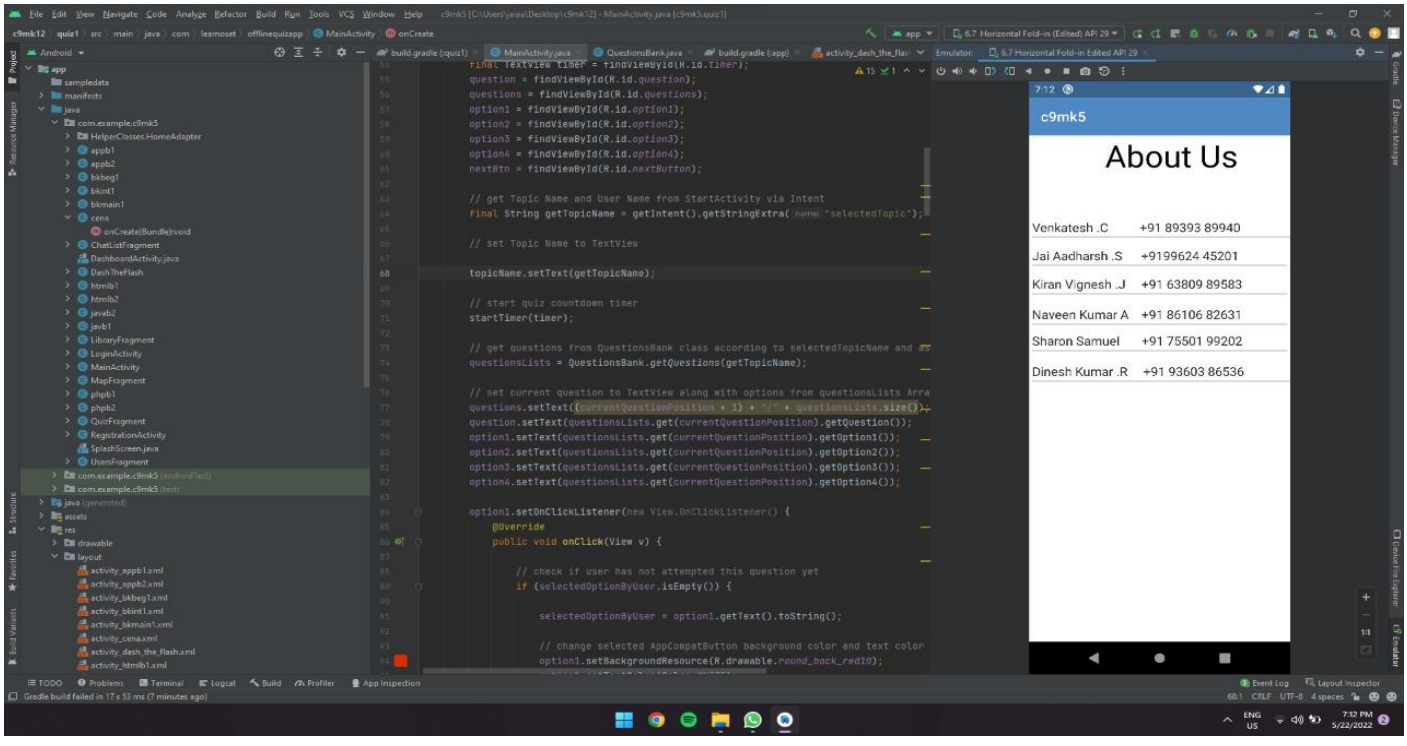
Zoomed Map Fragment:



E-book Fragment:



Contact us:



10. CONCLUSION

This project has been a rewarding experience in more than one way. The entireproject work has enlightened us in the following areas.

- The project “Learning Application” delivers a modern, interactive, and personalized digital environment for innovative learners.
- Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.
- scheduling a project and adhering to that schedule creates a strong sense of time management.
- Sense of teamwork has developed and confidence of handling real life project hasincreased to a great extent.
- Initially, there were problem with the validation but with discussions, we were to implement validations.

11. WEB REFERENCES

- **www.google.com**
- **www.developer.android.com**
- **www.w3schools.com**
- **www.youtube.com**
- **www.javapoint.com**
- **www.stackoverflow.com**
- **www.github.com**