**LIST OF FIGURES**

**FIGURE PAGE NO**

Navigation Drawer 10

Full Screen Activity 11

Dashboard Activity 12

Basic Activity 13

Use Case Diagram 18

Sequence Diagram 19

Job search Activity 42

Job Notifications 42

Resume Upload 43

Aptitude Page 43

**LIST OF ABBREVATIONS & NOMENCLATURE**

ABBREVATION MEANING

UI User Interface

T&P Training and Placement

CV Curriculum vitae

FB FireBase

RDB Real Time Database

ALU  arithmetic logic unit

AM Amplitude Modulation

API Application Programming Interface

DA Destination address

DB Database

DBA Database Administrator

DBMS Database management system

**LIST OF SYMBOLS AND NOMENCLATURE**

**SYMBOL NOMENCLATURE**

‘\*’ Dereferencing operator

‘+’,’-‘ Unary plus and minus operator

‘++’ Increment operator

‘—‘ Decrement operator

‘!’ Logical negation operator

‘~’ Bitwise complement operator

‘<,>,<=,=>’ Relational Operators

‘==’,’!=’ Equality Operators

‘.’,’- >’ Selection Operators

‘[ ]’ Array subscript operator

‘( )’ Parentheses operators

‘{ }’ Braces

‘=’ Equal sign

‘,’ Comma operator

‘;’ Semicolon

‘:’ Colon

‘\*’ Asterisk

‘” “’ Quotes

‘(type)’ TypeCast

‘…..’ Ellipsis

‘//’ Floor Division

in Evaluates to true if it finds a variable

in the specified sequence and false

otherwise

**ABSTRACT**

Android app for the students who are facing struggle to face the Interview during their final year phase ,we came across to build a app for those students to overcome their problems . Here the student can learn the aptitude which helps him to get on all the first rounds conducted by the companies during interviews and also for government exams such as bank, railways .The student can learn formulae regarding to solve the concern problem ,that linked to popular websites in our app. Students can learn tips and tricks to get a job through our app.

The main feature of our application is students can search for the job nearer to their locations . Student can upload their resume through our portal and we will provide the notifications regarding to their resumes .

During our survey we came around students are lack in soft skills , so we are provided the quiz to improve the student knowledge and test their skills that he can clear the aptitude round . The quiz questions will be changed once he completed the level one .

**CHAPTER 1**

* 1. **PURPOSE**

In this modern era search for employment is one heck of a job to do, rather than just submitting your resume. Job interviews play a crucial part in shaping our life, so here we are with "INTERVIEW APP" a great App to help you succeed in your interview.   
  
 We provide you with all the support you need to perform the best during the interview. Our "Interview Guide" App provides you with detailed guidelines for Resume writing, Personal Interview, Group discussion, Phone Etiquette and many more. There is no fixed platform that focus on the news of the changing trends. So, this project will help them as a guide in choosing the technologies.

* 1. **OBJECTIVES**

The app will helps the students to learn the aptitude for their first round during the interview ,it also provide tips and smart tricks to clear the aptitude round .The main feature of our app will helps the students to find the interviews nearer to their locations and also get notifications regarding to their streams. Our app helps the students to prepare company wise regarding .

* 1. **TECHNOLOGIES**

Technologies that are used to make pocket code android app are Android studio,

Firebase, other supporting libraries. Kotlin is used to develop the project. Kotlin is the official language of android and released in 2018.

**CHAPTER 2**

**LITERATURE REVIEW**

Once upon a time getting a job is quite easy because of less competition . Now in a revolutionized world getting a job is very difficult ,students are facing many difficulties to crack the interview due to lack of aptitude and verbal knowledge . According to our literature survey students don’t know how to prepare for aptitude and other competitive examinations . This app allows each and every student how to prepare and crack the competitive examinations in an easy way . Searching is done manually based on the company criteria training and placement head will identify the eligible student by looking the excel sheet.TP head has to see each and every student marks and their eligibility. No searching method is provided. The student will get notify through traditional notice board and emails . In various colleges, training and placement organizer have to manage the CV‟s and documents of students for their training and placement manually. T&P organizer have to collect the information of various companies who want to recruit students and notify students time to time about them. If any modification or updating are required in CV of any student, it has to search and to be done it manually.

So there is a need to develop an Android App that can solve the mentioned problem. This App will reduce the paperwork and accuracy in result. Manage the man and machine resources efficiently. Secured check in, check out & updates. Allow easy navigation about T&P

Information . You decide where, when, and how quickly you’ll learn, and you can do it anytime, anywhere, on any device. Small-sized lessons and fun practice sessions grab your attention and keep you focused, for guaranteed best results

**CHAPTER 3**

**HARDWARE REQUIREMENT**

**Client side:**

|  |  |
| --- | --- |
| **RAM** | 1 GB |
| **Android API Level** | 20 + |
| **Processor** | 1.0 GHz |

**Server side:**

|  |  |
| --- | --- |
| **RAM** | 1 GB |
| **Harddisk** | 20 GB |
| **Processor** | 2.0 GHz |

**SOFTWARE REQUIREMENT**

**Client side:**

|  |  |
| --- | --- |
| Android | Sdk |
| Compiler | Compiler app |

**Server side:**

|  |  |
| --- | --- |
| Language | Kotlin |
| Database Server | Firebase |
| Operating System | Windows or any equivelent OS |

**Front End :** Java, Xml

**Back End :** Firebase.

**CHAPTER 4**

**4.1 UI DESIGN**

Android is a widely used OS made for smart phones and tablets. It is an open source project led by Google and it is released under Apache License. This permissive license helped this OS to be widely adopted and allows the manufacturers to freely modify and customize it. As matter of fact, despite Android being designed for smartphones and tablets, it is also used in TVs, cameras and so on. Moreover, Android has a very large community that extend its features and creates apps that cover almost all aspects.

All android applications, called apps, are built on Android UI framework. App interface is the first thing a user sees and interacts with. From the user perspective, this framework keeps the overall experience consistent for every app installed in our smartphone or tablets. At the same time, from the developer perspective, this framework provides some basic blocks that can be used to build complex and consistent user interface (API).

Android UI interface is divided in three different areas:

• Home screen

• All apps

• Recent screen

The home screen is the “landing” area when we power our phone on. This interface is highly customizable and themed. Using widgets we can create and personalize our “home” screen. All apps is the interface where the app installed are displayed, while recent screens are the list of last used apps. Since its born, Android has changed a lot in terms of its features and its interfaces. The growth of the smartphone power made possible creating ever more appealing apps. At the beginning, apps in Android did not have a consistent interface and well defined rules so every app had a different approach, navigation structure and buttons position. This caused user confusion and it was one of the most important missing features compared to the iOS

•View:- It is the base class for all visual components (control and widgets). All the controls present in an android app are derived from this class. A View is an object that draws something on a smartphone screen and enables an user to interact with it.

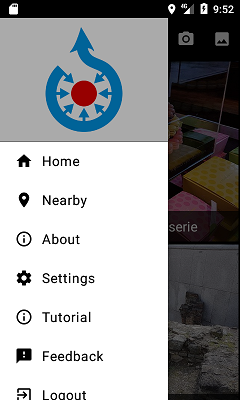
• Viewgroup :- A ViewGroup can contain one or more Views and defines how these Views are placed in the user interface (these are used along with Android Layout managers.)

•Fragments :- Introduced from API level 11, this component encapsulates a single piece of UI interface. They are very usefulwhen we have to create and optimize our app user interface for multiple devices or multiple screen size.

•Activities :- Usually an Android app consists of several activities that exchange data and information.

**4.1.1 NAVIGATION DRAWER**

* The navigation drawer is a panel that displays the app’s main navigation options on the left edge of the screen. It is hidden most of the time, but is revealed when the user swipes a finger from the left edge of the screen or, while at the top level of the app, the user touches the app icon in the action bar. The main content view must be the first child in the Drawer Layout because the XML order implies z-ordering and the drawer must be on top of the content.
* The main content view is set to match the parent view's width and height, because it represents the entire UI when the navigation drawer is hidden.
* The drawer view must specify its horizontal gravity with the android:layout\_gravity attribute. To support right-to-left (RTL) languages, specify the value with "start" instead of "left" (so the drawer appears on the right when the layout is RTL).
* The drawer view width fits the content and the height matches the parent view. The drawer width should be no more than 320dp so the user can always see a portion of the main content.



**4.1.2 FULL SCREEN ACTIVITY**

The geometry of a view is that of a rectangle. A view has a location, expressed as a pair of *left* and *top* coordinates, and two dimensions, expressed as a width and a height. The unit for location and dimensions is the pixel.

It is possible to retrieve the location of a view by invoking the methods [getLeft()](https://developer.android.com/reference/android/view/View.html#getLeft()) and [getTop()](https://developer.android.com/reference/android/view/View.html#getTop()). The former returns the left, or X, coordinate of the rectangle representing the view. The latter returns the top, or Y, coordinate of the rectangle representing the view. These methods both return the location of the view relative to its parent. For instance, when getLeft() returns 20, that means the view is located 20 pixels to the right of the left edge of its direct parent.

In addition, several convenience methods are offered to avoid unnecessary computations, namely [getRight()](https://developer.android.com/reference/android/view/View.html#getRight()) and [getBottom()](https://developer.android.com/reference/android/view/View.html#getBottom()). These methods return the coordinates of the right and bottom edges of the rectangle representing the view. For instance, calling [getRight()](https://developer.android.com/reference/android/view/View.html#getRight()) is similar to the following computation: getLeft() + getWidth() (see [Size](https://developer.android.com/reference/android/view/View.html#SizePaddingMargins) for more information about the width.)

****

**4.1.3 BASIC ACTIVITY**

Android Studio provides code templates that follow the Android design and development best practices to get you on the right track to creating beautiful, functional apps. You can use templates to create new app modules, individual activities, or other specific Android project components.

Some templates provide starter code for common usage contexts, such as navigation drawers or login screens. You can choose from these app module and activity templates when you first [create your project](https://developer.android.com/studio/projects/create-project.html), when you [add a new app module within an existing project](https://developer.android.com/studio/projects/add-app-module.html), or when you add a new activity within an app module.

In addition to activities, you can also add other Android project components to an existing app using templates. These templates include both code components, such as services and fragments, and non-code components, such as folders and XML files.

This page discusses how to add Android project components like activities to your project and describes the commonly used activity templates available in Android Studio. Note that most templates depend on the [Android Support Library](https://developer.android.com/tools/support-library/features.html) to include user interface principles based on [material design](https://developer.android.com/design/material/index.html). The list of templates provided in Android Studio is constantly growing. Android Studio groups templates by the type of component that they add, such as an **Activity** or an **XML** file, as shown in figure 1.

To add an Android project component using a template, use the **Project** window. Right-click on the folder in which you want to add the new component, and select **New**. Based on what components can be added to the folder you clicked on, you then see a list of template types like those shown in figure 1.

When you select the template you want to add, a corresponding wizard window appears and asks for the component's configuration information, such as its name. After you enter the configuration information, Android Studio creates and opens the files for your new component. It also runs a Gradle build to sync your project.

**4.2 DASHBOARD**

The Activity dashboard gives you an overview of all the activity on your GitHub Enterprise instance.

The Activity dashboard provides weekly, monthly, and yearly graphs of the number of

* new pull requests
* merged pull requests
* new issues
* closed issues
* new issue comments
* new repositories
* new user accounts
* new organizations
* new teams

As an app/game developer (I'm doing Andorid apps), why do we have an option to put those links to the other app stores? I understand links to the Youtube.  
I've put my first ever developed free game to itch.io, also to Amazon and Google play. So far, believe me or not, itch.io users generate most of my downloads. Here comes my question, does (itch.io dashboard: Activity Summary) actually counts clicks on links that leads to another app store?

the Dashboard was converting from miles to kilometres, but a quick conversion shows this not to be the case. I haved walked the same footpaths each day for some time so have a good idea of the distances involved. As far as I can tell the distances recorded in the Activities log are as accurate as they have ever been in 18 months of use

****

**4.3 QUIZ ACTIVITY**

An exam or quiz is a form of student assessment that measures knowledge, skills, and abilities. Generally, an exam is a culminating assessment that assesses a student over a large period of time and over a range of material. A quiz is generally a frequent and short assessment that can gauge a student’s retention and comprehension of a small amount of information.   A quiz can function throughout a course as an informative feedback device allowing both the instructor and the students to see where they are excelling or need more focus.  In order to effectively create exams and quizzes, it is important to establish and understand the learning objectives that are being measured



**4.3.1 LANGUAGE EDITORS**

Source code editors have features specifically designed to simplify and speed up input of source code, such as [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [indentation](https://en.wikipedia.org/wiki/Indent_style), [autocomplete](https://en.wikipedia.org/wiki/Autocomplete) and [brace matching](https://en.wikipedia.org/wiki/Brace_matching) functionality. These editors also provide a convenient way to run a [compiler](https://en.wikipedia.org/wiki/Compiler), [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)), [debugger](https://en.wikipedia.org/wiki/Debugger), or other program relevant for the [software development process](https://en.wikipedia.org/wiki/Software_development_process). So, while many text editors can be used to edit source code, if they don't enhance, automate or ease the editing of code, they are not *source code editors*, but simply *text editors that can also be used to edit source code*.

[Structure editors](https://en.wikipedia.org/wiki/Structure_editor) are a different form of source code editor, where instead of editing raw text, one manipulates the code's structure, generally the [abstract syntax tree](https://en.wikipedia.org/wiki/Abstract_syntax_tree). In this case features such as syntax highlighting, validation, and code formatting are easily and efficiently implemented from the [concrete syntax tree](https://en.wikipedia.org/wiki/Concrete_syntax_tree) or abstract syntax tree, but editing is often more rigid than free-form text. Structure editors also require extensive support for each language, and thus are harder to extend to new languages than text editors, where basic support only requires supporting syntax highlighting or indentation. For this reason, strict structure editors are not popular for source code editing, though some IDEs provide similar functionality.

A source code editor can check syntax while code is being entered and immediately warn of syntax problems. A few source code editors compress source code, typically converting common keywords into single-byte tokens, removing unnecessary whitespace, and converting numbers to a binary form. Such tokenizing editors later uncompress the source code when viewing it, possibly [prettyprinting](https://en.wikipedia.org/wiki/Prettyprint) it with consistent capitalizing and spacing. A few source code editors do both.

A compiler is likely to perform many or all of the following operations: [preprocessing](https://en.wikipedia.org/wiki/Preprocessor), [lexical analysis](https://en.wikipedia.org/wiki/Lexical_analysis), [parsing](https://en.wikipedia.org/wiki/Parsing), [semantic analysis](https://en.wikipedia.org/wiki/Semantic_analysis_(compilers)) ([syntax-directed translation](https://en.wikipedia.org/wiki/Syntax-directed_translation)), conversion of input programs to an [intermediate representation](https://en.wikipedia.org/wiki/Intermediate_representation), [code optimization](https://en.wikipedia.org/wiki/Code_optimization) and [code generation](https://en.wikipedia.org/wiki/Code_generation_(compiler)). Compilers implement these operations in phases that promote efficient design and correct transformations of source input to target output. Program faults caused by incorrect compiler behavior can be very difficult to track down and work around; therefore, compiler implementers invest significant effort to ensure [compiler correctness](https://en.wikipedia.org/wiki/Compiler_correctness).[[2]](https://en.wikipedia.org/wiki/Compiler#cite_note-Sun2016-2)

Compilers are not the only translators used to transform source programs. An [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) is computer software that transforms and then executes the indicated operations. The translation process influences the design of computer languages which leads to a preference of compilation or interpretation. In practice, an interpreter can be implemented for compiled languages and compilers can be implemented for interpreted languages.

However, there are many different types of compilers. If the compiled program can run on a computer whose [CPU](https://en.wikipedia.org/wiki/Central_processing_unit) or [operating system](https://en.wikipedia.org/wiki/Operating_system) is different from the one on which the compiler runs, the compiler is a [cross-compiler](https://en.wikipedia.org/wiki/Cross-compiler). A [bootstrap compiler](https://en.wikipedia.org/wiki/Bootstrap_compiler) is written in the language that is compiled. A program that translates from a [low-level language](https://en.wikipedia.org/wiki/Low-level_language) to a higher level one is a [decompiler](https://en.wikipedia.org/wiki/Decompiler). A program that translates between high-level languages is usually called a [source-to-source compiler](https://en.wikipedia.org/wiki/Source-to-source_compiler) or transpiler. A language [rewriter](https://en.wikipedia.org/wiki/Rewriting) is usually a program that translates the form of expressions without a change of language. The term [compiler-compiler](https://en.wikipedia.org/wiki/Compiler-compiler) refers to tools used to create parsers that perform syntax analysis.

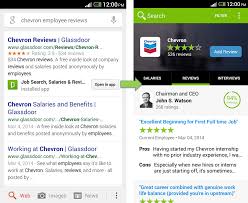
**4.4 UPLOAD RESUME & JOB NOTIFICATIONS ACTIVITY**

In a world where all your candidates have gone mobile, SmartRecruiters is the hiring platform used by thousands of businesses worldwide to create and advertise jobs, source and manage candidates, and make the best hire. Loved by customers for its ease of use, SmartRecruiters is the easiest way to collaborate and manage mobile recruiting for your business.   
The student can upload resume through the resume activity . The student must upload the resume with .pdf, and .word document .If He uploaded the resume successfully it shows the toast message that the file is successfully uploaded .

Notifications activity shows the notifications regarding the job s to the student .Students get notifications after he uploading the resume with in an one day that we will scan their resume and we will upload the push job notifications.



**4.4.1 JOB SEARCH ACTIVITY**

****

.

See the latest jobs with Glassdoor's fast and easy-to-use job search. We search millions of jobs from thousands of websites. You're sure to find great job opportunities. Save interesting jobs so you'll never miss out on a great opportunity. When you're ready to apply on your computer, all of your saved jobs will be waiting for you. Stay up-to-date on the latest company reviews, salary reports and interview questions - shared by those who know the company the best - the employees. Looking for a new job at your office can be tough and can even land you in sudden unpleasant situations. To escape from the unimaginable trauma, it is best to take advantage of technology by downloading the best job search apps of India. You can access the recently updated positions from your phone anytime and anywhere without the fear of getting caught. Plus you can respond to your job mails quickly and easily. Here are five job search apps which must be present in your phone if you are hunting for a job .

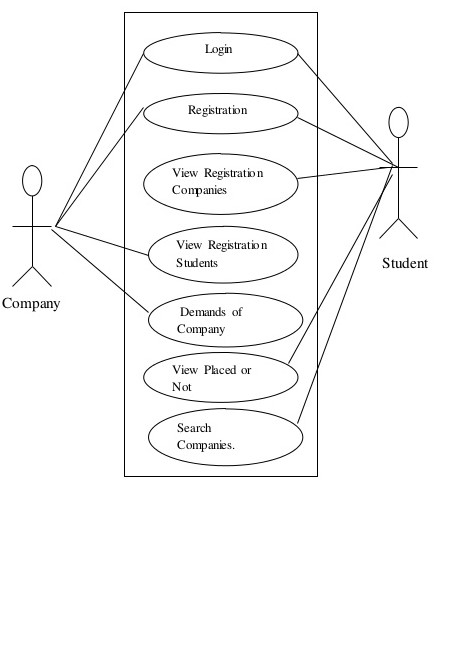
**4.5 TUTORIALS AND EXERCISES**

A **tutorial** is a method of transferring knowledge and may be used as a part of a [learning](https://en.wikipedia.org/wiki/Learning) process. More [interactive](https://en.wikipedia.org/wiki/Interactivity) and specific than a [book](https://en.wikipedia.org/wiki/Book) or a [lecture](https://en.wikipedia.org/wiki/Lecture), a tutorial seeks to teach by example and supply the information to complete a certain task.

Online tutoring is the process of tutoring in an online, virtual environment or networked environment in which teachers and learners are separated by time and space. Online tutoring, as a reflection of the diversity of the wider Internet, is practiced using many different approaches and is addressed to distinct sets of users. The distinctions are in online content and interface, as well as in tutoring and tutor-training methodologies. Definitions associated with online tutoring vary widely, reflecting the ongoing evolution of the technology, the refinement and variation in online learning methodology, and the interactions of the organizations that deliver online tutoring services with the institutions, individuals, and learners that employ the services. This form of Internet service is a classical micropublishing situation.

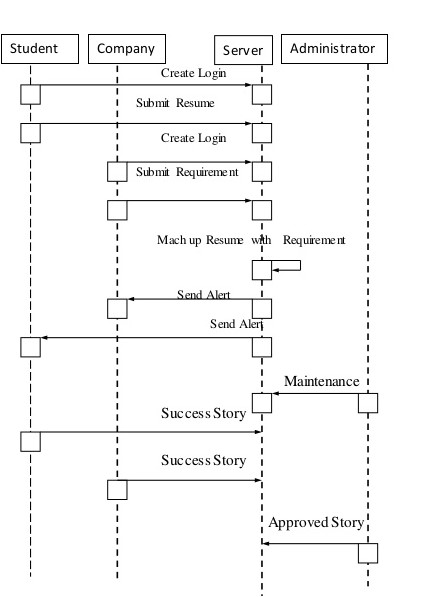
**CHAPTER 5**

**5.1 USE CASE DIAGRAM :**

****

A **use case diagram** is a dynamic or behavior **diagram in UML**. **Use case diagrams** model the functionality of a system using actors and **use cases**. **Use cases** are a set of actions, services, and functions that the system needs to perform.

**5.2 SEQUENCE DIAGRAM :**

****

A **sequence diagram** shows object interactions arranged in time **sequence**. It depicts the objects and classes involved in the scenario and the **sequence** of messages exchanged between the objects needed to carry out the functionality of the scenario.  **Sequence diagrams** are sometimes called event **diagrams** or event scenarios

**CHAPTER 6**

**CODING :**

**MainActivity.java**

**package com.sree.sreekanthreddy.interview;**

**import android.content.Intent;**

**import android.support.design.widget.NavigationView;**

**import android.support.v4.view.GravityCompat;**

**import android.support.v4.widget.DrawerLayout;**

**import android.support.v7.app.ActionBarDrawerToggle;**

**import android.support.v7.app.AppCompatActivity;**

**import android.os.Bundle;**

**import android.view.Menu;**

**import android.view.MenuItem;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.widget.AdapterView;**

**import android.widget.BaseAdapter;**

**import android.widget.ImageView;**

**import android.widget.ListView;**

**import android.widget.TextView;**

**public class MainActivity<navigation> extends AppCompatActivity {**

**private DrawerLayout mDrawerLayout;**

**NavigationView navigationView;**

**private ActionBarDrawerToggle mToggle;**

**ListView mylistview;**

**int[] images={R.drawable.aptitude,R.drawable.interview,R.drawable.quiz,R.drawable.formula,R.drawable.tips,};**

**String[] Names={"Aptitude","Interview","Quiz","Formula","Tips and Tricks"};**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**mDrawerLayout = (DrawerLayout) findViewById(R.id.drawerLayout);**

**mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, R.string.open, R.string.close);**

**mDrawerLayout.addDrawerListener(mToggle);**

**mToggle.syncState();**

**getSupportActionBar().setDisplayHomeAsUpEnabled(true);**

**mylistview = (ListView) findViewById(R.id.listview);**

**CustomAdapter customAdapter = new CustomAdapter();**

**mylistview.setAdapter(customAdapter);**

**navigationView=(NavigationView)findViewById(R.id.navigate);**

**navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {**

**@Override**

**public boolean onNavigationItemSelected(MenuItem item) {**

**switch (item.getItemId()) {**

**case R.id.resumeupload:**

**Intent anIntent = new Intent(getApplicationContext(), ResumeActivity.class);**

**startActivity(anIntent);**

**mDrawerLayout.closeDrawers();**

**break;**

**case R.id.notifycations:**

**anIntent = new Intent(getApplicationContext(), JobActivity.class);**

**startActivity(anIntent);**

**mDrawerLayout.closeDrawers();**

**break;**

**case R.id.jobsearch:**

**anIntent=new Intent(getApplicationContext(),JobSearch.class);**

**startActivity(anIntent);**

**mDrawerLayout.closeDrawers();**

**}**

**return false;**

**}**

**});**

**mylistview.setOnItemClickListener(new AdapterView.OnItemClickListener() {**

**@Override**

**public void onItemClick(AdapterView<?> parent, View view, int position, long id) {**

**if (position == 0) {**

**Intent myIntent = new Intent(view.getContext(), AptitudeActivity.class);**

**startActivityForResult(myIntent, 0);**

**}**

**if (position == 1) {**

**Intent myIntent = new Intent(view.getContext(), Interview.class);**

**startActivityForResult(myIntent, 0);**

**}**

**if (position == 2) {**

**Intent myIntent = new Intent(view.getContext(), QuizActivity.class);**

**startActivityForResult(myIntent, 0);**

**}**

**if (position == 3) {**

**Intent myIntent = new Intent(view.getContext(), Formula.class);**

**startActivityForResult(myIntent, 0);**

**}**

**if (position == 4) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","tip");**

**bundle.putString("domain","tips");**

**Intent in = new Intent(getApplicationContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**}**

**});**

**}**

**class CustomAdapter extends BaseAdapter implements com.sree.sreekanthreddy.interview.CustomAdapter {**

**@Override**

**public int getCount() {**

**return images.length;**

**}**

**@Override**

**public Object getItem(int position) {**

**return null;**

**}**

**@Override**

**public long getItemId(int position) {**

**return 0;**

**}**

**@Override**

**public View getView(int position, View convertView, ViewGroup parent) {**

**View view=getLayoutInflater().inflate(R.layout.customlayout,null);**

**ImageView myImageView=(ImageView)view.findViewById(R.id.imageView);**

**TextView myTextView=(TextView) view.findViewById(R.id.textview);**

**myImageView.setImageResource(images[position]);**

**myTextView.setText(Names[position]);**

**return view;**

**}**

**}**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**if(mToggle.onOptionsItemSelected(item))**

**{**

**return true;**

**}**

**return super.onOptionsItemSelected(item);**

**}**

**Quiz Activity.java :**

**package com.sree.sreekanthreddy.interview;**

**import android.support.v7.app.AppCompatActivity;**

**import android.os.Bundle;**

**import android.util.Log;**

**import android.view.KeyEvent;**

**import android.view.View;**

**import android.widget.Button;**

**import android.widget.EditText;**

**import android.widget.ProgressBar;**

**import android.widget.TextView;**

**import java.util.ArrayList;**

**import cn.pedant.SweetAlert.SweetAlertDialog;**

**public class QuizActivity extends AppCompatActivity {**

**TextView questionLabel, questionCountLabel, scoreLabel;**

**EditText answerEdt;**

**Button submitButton;**

**ProgressBar progressBar;**

**ArrayList<QuestionModel> questionModelArraylist;**

**int currentPosition = 0;**

**int numberOfCorrectAnswer = 0;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_quiz);**

**questionCountLabel = findViewById(R.id.noQuestion);**

**questionLabel = findViewById(R.id.question);**

**scoreLabel = findViewById(R.id.score);**

**answerEdt = findViewById(R.id.answer);**

**submitButton = findViewById(R.id.submit);**

**progressBar = findViewById(R.id.progress);**

**questionModelArraylist = new ArrayList<>();**

**setUpQuestion();**

**setData();**

**submitButton.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**checkAnswer();**

**}**

**});**

**answerEdt.setOnKeyListener(new View.OnKeyListener() {**

**public boolean onKey(View v, int keyCode, KeyEvent event) {**

**// If the event is a key-down event on the "enter" button**

**Log.e("event.getAction()",event.getAction()+"");**

**Log.e("event.keyCode()",keyCode+"");**

**if ((event.getAction() == KeyEvent.ACTION\_DOWN) &&**

**(keyCode == KeyEvent.KEYCODE\_ENTER)) {**

**checkAnswer();**

**return true;**

**}**

**return false;**

**}**

**});**

**}**

**public void checkAnswer(){**

**String answerString = answerEdt.getText().toString().trim();**

**if(answerString.equalsIgnoreCase(questionModelArraylist.get(currentPosition).getAnswer())){**

**numberOfCorrectAnswer ++;**

**new SweetAlertDialog(QuizActivity.this, SweetAlertDialog.SUCCESS\_TYPE)**

**.setTitleText("Good job!")**

**.setContentText("Right Answer")**

**.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {**

**@Override**

**public void onClick(SweetAlertDialog sweetAlertDialog) {**

**currentPosition ++;**

**setData();**

**answerEdt.setText("");**

**sweetAlertDialog.dismiss();**

**}**

**})**

**.show();**

**}else {**

**new SweetAlertDialog(QuizActivity.this, SweetAlertDialog.ERROR\_TYPE)**

**.setTitleText("Wrong Answer")**

**.setContentText("The right answer is : "+questionModelArraylist.get(currentPosition).getAnswer())**

**.setConfirmText("OK")**

**.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {**

**@Override**

**public void onClick(SweetAlertDialog sDialog) {**

**sDialog.dismiss();**

**currentPosition ++;**

**setData();**

**answerEdt.setText("");**

**}**

**})**

**.show();**

**}**

**int x = ((currentPosition+1) \* 100) / questionModelArraylist.size()**

**progressBar.setProgress(x);**

**}**

**public void setUpQuestion(){**

**questionModelArraylist.add(new QuestionModel("A train running at the speed of 60 km/hr crosses a pole in 9 seconds. What is the length of the train? ","150"));**

**questionModelArraylist.add(new QuestionModel("The length of the bridge, which a train 130 metres long and travelling at 45 km/hr can cross in 30 seconds, is","245"));**

**questionModelArraylist.add(new QuestionModel(" A train 240 m long passes a pole in 24 seconds. How long will it take to pass a platform 650 m long?","89"));**

**questionModelArraylist.add(new QuestionModel("A sum of money at simple interest amounts to Rs. 815 in 3 years and to Rs. 854 in 4 years. The sum is","698"));**

**questionModelArraylist.add(new QuestionModel("A sum fetched a total simple interest of Rs. 4016.25 at the rate of 9 p.c.p.a. in 5 years. What is the sum","8925"));**

**questionModelArraylist.add(new QuestionModel("It was Sunday on Jan 1, 2006. What was the day of the week Jan 1, 2010?","Friday"));**

**questionModelArraylist.add(new QuestionModel("A sum of money amounts to Rs. 9800 after 5 years and Rs. 12005 after 8 years at the same rate of simple interest. The rate of interest per annum is:","12"));**

**questionModelArraylist.add(new QuestionModel("The cost price of 20 articles is the same as the selling price of x articles. If the profit is 25%, then the value of x is","16"));**

**questionModelArraylist.add(new QuestionModel("In the first 10 overs of a cricket game, the run rate was only 3.2. What should be the run rate in the remaining 40 overs to reach the target of 282 runs?","6.25"));**

**questionModelArraylist.add(new QuestionModel("What was the day of the week on 17th June, 1998?","Wednesday"));**

**questionModelArraylist.add(new QuestionModel("A man's speed with the current is 15 km/hr and the speed of the current is 2.5 km/hr. The man's speed against the current is:","10km/hr"));**

**questionModelArraylist.add(new QuestionModel("A person crosses a 600 m long street in 5 minutes. What is his speed in km per hour?","7.2"));**

**questionModelArraylist.add(new QuestionModel("A can do a work in 15 days and B in 20 days. If they work on it together for 4 days, then the fraction of the work that is left is","8/15"));**

**questionModelArraylist.add(new QuestionModel("Two numbers are respectively 20% and 50% more than a third number. The ratio of the two numbers is:","4:5"));**

**questionModelArraylist.add(new QuestionModel("Tickets numbered 1 to 20 are mixed up and then a ticket is drawn at random. What is the probability that the ticket drawn has a number which is a multiple of 3 or 5?","9/20"));**

**}**

**public void setData(){**

**if(questionModelArraylist.size()>currentPosition) {**

**questionLabel.setText(questionModelArraylist.get(currentPosition).getQuestionString());**

**scoreLabel.setText("Score :" + numberOfCorrectAnswer + "/" + questionModelArraylist.size());**

**questionCountLabel.setText("Question No : " + (currentPosition + 1));**

**}else{**

**new SweetAlertDialog(QuizActivity.this, SweetAlertDialog.SUCCESS\_TYPE)**

**.setTitleText("You have successfully completed the quiz")**

**.setContentText("Your score is : "+ numberOfCorrectAnswer + "/" + questionModelArraylist.size())**

**.setConfirmText("Restart")**

**.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {**

**@Override**

**public void onClick(SweetAlertDialog sDialog) {**

**sDialog.dismissWithAnimation();**

**currentPosition = 0;**

**numberOfCorrectAnswer = 0;**

**progressBar.setProgress(0);**

**setData();**

**}**

**})**

**.setCancelText("Close")**

**.setCancelClickListener(new SweetAlertDialog.OnSweetClickListener() {**

**@Override**

**public void onClick(SweetAlertDialog sDialog) {**

**sDialog.dismissWithAnimation();**

**finish();**

**}**

**})**

**.show();**

**}**

**}**

**}**

**JOB NOTIFICATIONS AND RESUME ACTIVITY :**

**package com.sree.sreekanthreddy.interview;**

**import android.Manifest;**

**import android.app.ProgressDialog;**

**import android.content.Intent;**

**import android.content.pm.PackageManager;**

**import android.net.Uri;**

**import android.support.annotation.NonNull;**

**import android.support.v4.app.ActivityCompat;**

**import android.support.v4.content.ContextCompat;**

**import android.support.v7.app.AppCompatActivity;**

**import android.os.Bundle;**

**import android.view.View;**

**import android.widget.Button;**

**import android.widget.TextView;**

**import android.widget.Toast;**

**import com.google.android.gms.tasks.OnCompleteListener;**

**import com.google.android.gms.tasks.OnFailureListener;**

**import com.google.android.gms.tasks.OnSuccessListener;**

**import com.google.android.gms.tasks.Task;**

**import com.google.firebase.database.DatabaseReference;**

**import com.google.firebase.database.FirebaseDatabase;**

**import com.google.firebase.storage.FirebaseStorage;**

**import com.google.firebase.storage.OnProgressListener;**

**import com.google.firebase.storage.StorageReference;**

**import com.google.firebase.storage.UploadTask;**

**public class ResumeActivity extends AppCompatActivity {**

**Button selectFile,upload;**

**TextView notifications;**

**Uri pdfUri;**

**FirebaseStorage storage;**

**FirebaseDatabase database;**

**ProgressDialog progressDialog;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_resume);**

**storage=FirebaseStorage.getInstance();**

**database=FirebaseDatabase.getInstance();**

**selectFile=findViewById(R.id.selectFile);**

**upload=findViewById(R.id.upload);**

**notifications=findViewById(R.id.notifications);**

**selectFile.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**if(ContextCompat.checkSelfPermission(ResumeActivity.this, Manifest.permission.READ\_EXTERNAL\_STORAGE)== PackageManager.PERMISSION\_GRANTED)**

**{**

**selectPdf();**

**}**

**else**

**ActivityCompat.requestPermissions(ResumeActivity.this,new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE},9);**

**}**

**});**

**upload.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**if(pdfUri!=null)**

**uploadFile(pdfUri);**

**else**

**Toast.makeText(ResumeActivity.this,"Select a File",Toast.LENGTH\_SHORT).show();**

**}**

**});**

**}**

**private void uploadFile(Uri pdfUri) {**

**progressDialog =new ProgressDialog(this);**

**progressDialog.setProgressStyle(ProgressDialog.STYLE\_HORIZONTAL);**

**progressDialog.setTitle("Uploading File...");**

**progressDialog.setProgress(0);**

**progressDialog.show();**

**final String fileName=System.currentTimeMillis()+"";**

**StorageReference storageReference=storage.getReference();**

**storageReference.child("ResumeUploads").child(fileName).putFile(pdfUri)**

**.addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {**

**@Override**

**public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {**

**String url=taskSnapshot.getUploadSessionUri().toString();**

**DatabaseReference reference=database.getReference();**

**reference.child(fileName).setValue(url).addOnCompleteListener(new OnCompleteListener<Void>() {**

**@Override**

**public void onComplete(@NonNull Task<Void> task) {**

**if(task.isSuccessful())**

**Toast.makeText(ResumeActivity.this,"File Successfully uploaded",Toast.LENGTH\_SHORT).show();**

**else**

**Toast.makeText(ResumeActivity.this,"File Successfully uploaded",Toast.LENGTH\_SHORT).show();**

**}**

**});**

**}**

**}).addOnFailureListener(new OnFailureListener() {**

**@Override**

**public void onFailure(@NonNull Exception e) {**

**Toast.makeText(ResumeActivity.this,"File Successfully uploaded",Toast.LENGTH\_SHORT).show();**

**}**

**}).addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {**

**@Override**

**public void onProgress(UploadTask.TaskSnapshot taskSnapshot) {**

**int currentProgress=(int) (100\*taskSnapshot.getBytesTransferred()/taskSnapshot.getTotalByteCount());**

**progressDialog.setProgress(currentProgress);**

**}**

**});**

**}**

**@Override**

**public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {**

**if(requestCode==9&& grantResults[0]==PackageManager.PERMISSION\_GRANTED)**

**{**

**selectPdf();**

**}**

**else**

**Toast.makeText(ResumeActivity.this, "Please give Permission..", Toast.LENGTH\_SHORT).show();**

**}**

**private void selectPdf() {**

**Intent intent=new Intent();**

**intent.setType("application/pdf");**

**intent.setAction(Intent.ACTION\_GET\_CONTENT);**

**startActivityForResult(intent,80);**

**}**

**@Override**

**protected void onActivityResult(int requestCode, int resultCode, Intent data) {**

**if(requestCode==80&&resultCode==RESULT\_OK&&data!=null)**

**{**

**pdfUri=data.getData();**

**notifications.setText("A file is selected:"+data.getData().getLastPathSegment());**

**}**

**else {**

**Toast.makeText(ResumeActivity.this,"Please select a File",Toast.LENGTH\_SHORT).show();**

**}**

**}**

**}**

**Aptitude Activity .java**

**package com.sree.sreekanthreddy.interview;**

**import android.content.Context;**

**import android.content.Intent;**

**import android.net.Uri;**

**import android.os.Bundle;**

**import android.support.annotation.Nullable;**

**import android.support.v4.app.Fragment;**

**import android.view.LayoutInflater;**

**import android.view.View;**

**import android.view.ViewGroup;**

**import android.widget.AdapterView;**

**import android.widget.BaseAdapter;**

**import android.widget.ImageView;**

**import android.widget.ListView;**

**import android.widget.TextView;**

**public class Tab3 extends Fragment {**

**ListView mylistview;**

**String url;**

**int[] images = {R.drawable.antonym, R.drawable.comprehension, R.drawable.orderingwords, R.drawable.selectword, R.drawable.spottheerror, R.drawable.synonyms, R.drawable.verbalanalogies};**

**String[] Names = { "Antonyms", "Comprehension", "Ordering Words", "Select Word", "Spot The Error", "Synonyms", "Verbal Analogies"};**

**private static final String TAG = "Tab1Fragment";**

**@Nullable**

**@Override**

**public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {**

**View view = inflater.inflate(R.layout.fragment\_tab1, container, false);**

**super.onCreate(savedInstanceState);**

**mylistview = (ListView) view.findViewById(R.id.listview);**

**Tab3.CustomAdapter customAdapter = new Tab3.CustomAdapter();**

**mylistview.setAdapter(customAdapter);**

**mylistview.setOnItemClickListener(new AdapterView.OnItemClickListener() {**

**@Override**

**public void onItemClick(AdapterView<?> parent, View view, int position, long id) {**

**if (position == 0) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","antonym");**

**bundle.putString("domain","antonyms");**

**Intent in = new Intent(getContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**if (position == 1) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","comprehension");**

**bundle.putString("domain","comprehensions");**

**Intent in = new Intent(getContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**if (position == 2) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","ordering");**

**bundle.putString("domain","orderings");**

**Intent in = new Intent(getContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**if (position == 3) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","select");**

**bundle.putString("domain","selects");**

**Intent in = new Intent(getContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**if (position == 4) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","spot");**

**bundle.putString("domain","spots");**

**Intent in = new Intent(getContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**if (position == 5) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","synonym");**

**bundle.putString("domain","synonyms");**

**Intent in = new Intent(getContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**if (position == 6) {**

**Bundle bundle = new Bundle();**

**bundle.putInt("position",position);**

**bundle.putString("name","verbal");**

**bundle.putString("domain","verbals");**

**Intent in = new Intent(getContext(), Webactivity.class);**

**in.putExtra("bundle",bundle);**

**startActivity(in);**

**}**

**}**

**});**

**return view;**

**}**

**class CustomAdapter extends BaseAdapter {**

**@Override**

**public int getCount() {**

**return images.length;**

**}**

**@Override**

**public Object getItem(int position) {**

**return null;**

**}**

**@Override**

**public long getItemId(int position) {**

**return 0;**

**}**

**@Override**

**public View getView(int position, View convertView, ViewGroup parent) {**

**View view = getLayoutInflater().inflate(R.layout.customlayout1, null);**

**ImageView myImageView = (ImageView) view.findViewById(R.id.imageView);**

**TextView myTextView = (TextView) view.findViewById(R.id.textview);**

**myImageView.setImageResource(images[position]);**

**myTextView.setText(Names[position]);**

**return view;**

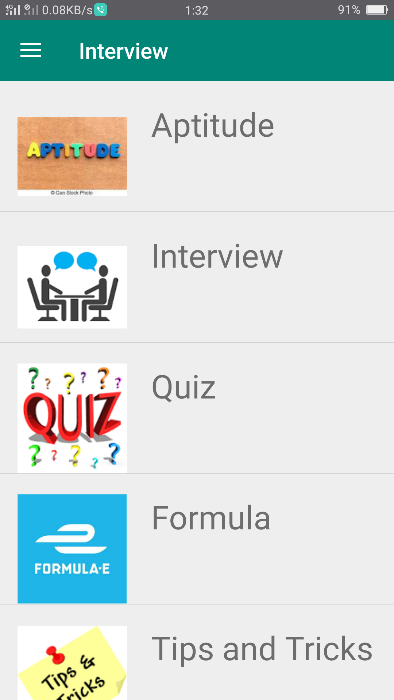
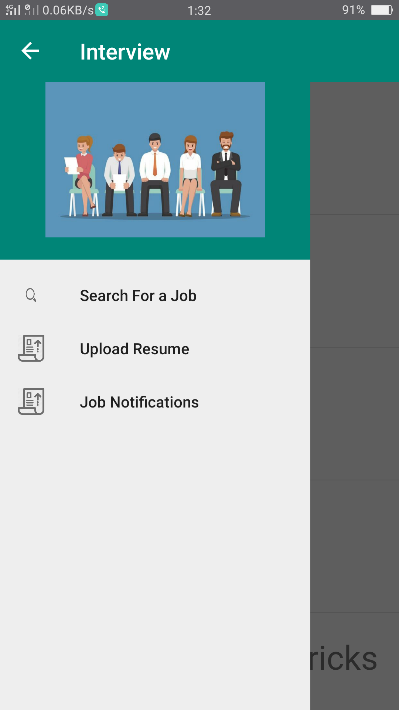
**}**

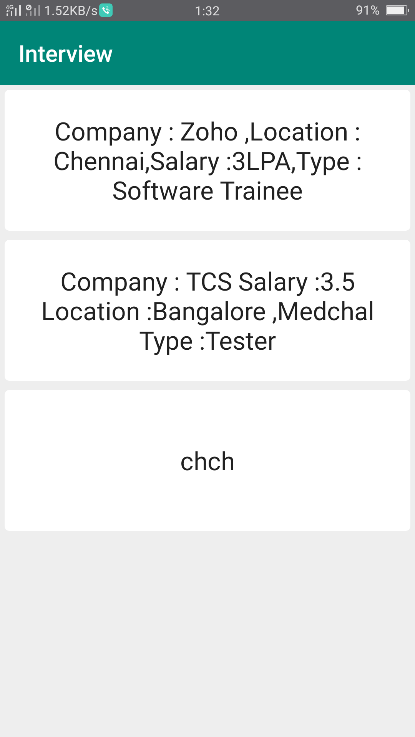
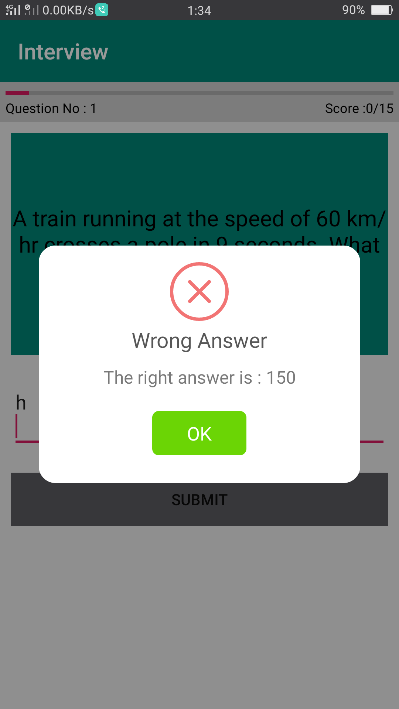
**}**

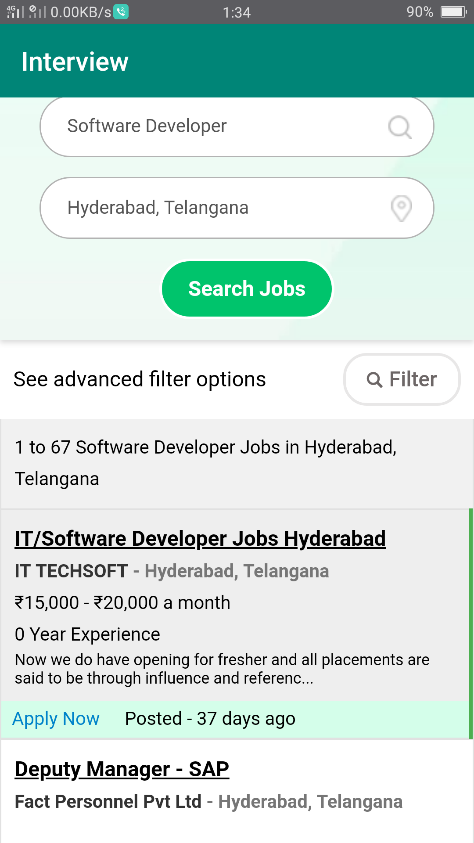
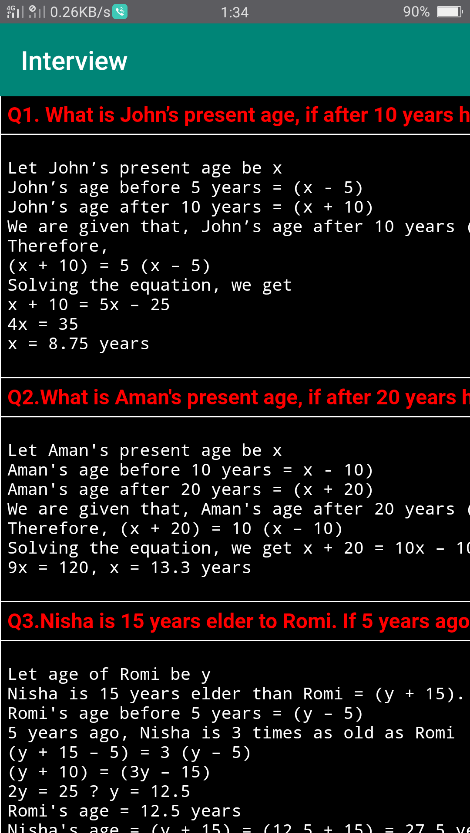
**}**

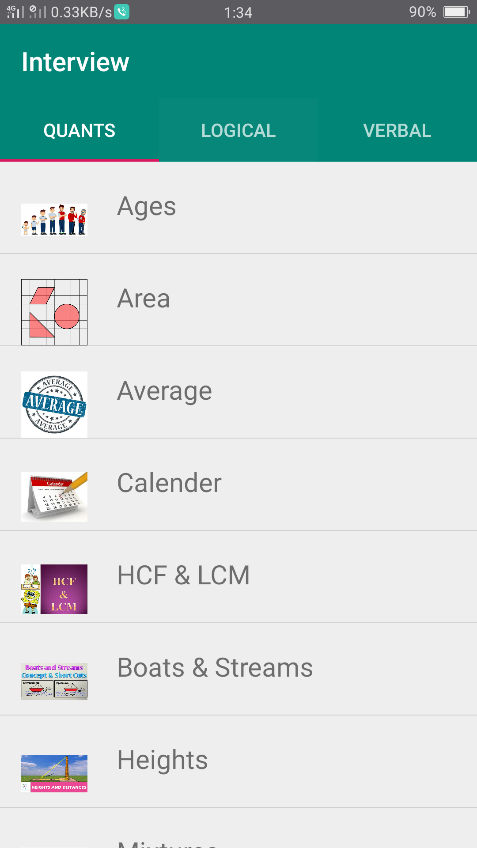
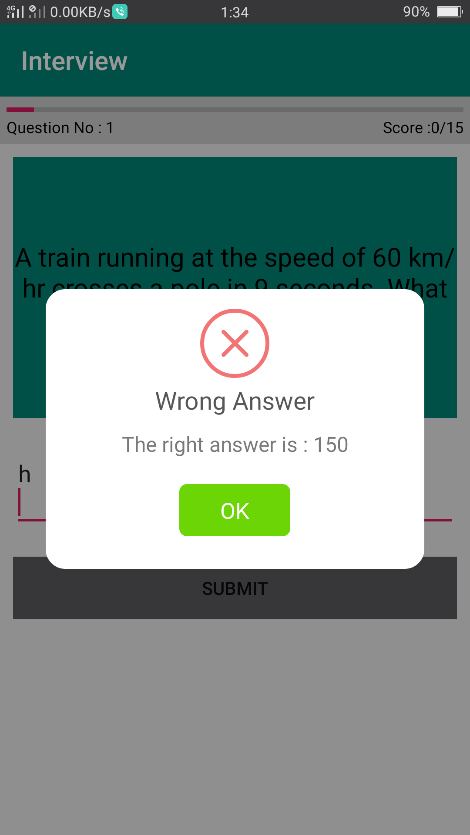
**CHAPTER 7**

**OUTPUT SCREENSHOTS :**

**** 

**** 

** **

** **

**CHAPTER 7**

**CONCLUSION AND BENEFITS**

**7.1 CONCLUSION**

Interview Project is worth applicable to the student community where knowledge plays a vital role in the students carrier. This project will also helps in the form of guide where the news will tells the students and It will serves as a good newsfeed for the students relating the recent technologies and trends.

The App will be monitored by us each and every week and if any bugs are reported and any issues in the design or UI we fix it regularly. Updates are also will be issued when ever there is a need for changing the app or flow of project.

**References:**

* An already existing app called “**solo learn**” where we find popular languages tutorials

<https://play.google.com/store/apps/details?id=com.sololearn&hl=en>

* AndroidDevelopers.com

<https://developer.android.com/index.html>

* Android-arsenal.com

<https://android-arsenal.com/>

* W3schools.com

https://www.w3schools.in/category/android-tutorial/

* Android Developers site.

[www.mkyong.com/tutorials/android-tutorial/](http://www.mkyong.com/tutorials/android-tutorial/)

* Udemy Course on Android Development by   
  Rob Percival

<https://www.udemy.com/user/robpercival/>

* Android Tutorials in you tube channel