**A Project report on**

**QUIZ-TUT LEARNING**

Submitted to

Ajay Kumar Garg Engineering College, Ghaziabad

****

As a partial fulfillment of Master of Computer Applications

(MCA V Sem, 2016-17)

**Submitted By:-**

Pravin Yadav (1402714030)

Nalin Sharma (1402714025)

Krishankant Duvey(1402714020)

**Uttar Pradesh Technical University, Lucknow**

**ACKNOWLEDGEMENT**

We express our sincere gratitude to all those who initiated and helped us in successful completion of our project. For the successful completion of our project we express our gratitude to **Mr. S.L. Kapoor (HOD).**

We are thankful to **Ms. Pratima Singh (Assistant Professor)**, our mentor for providing us with valuable and useful information while bringing out this project. Her inspiring suggestions and timely guidance enabled us to perceive the various aspects of the project in a new light.

The effort could not be successfully completed without timely guidance and suggestions by the staff members of **Ajay Kumar Garg Engineering College.**

Finally, we express our gratitude to all those who encouraged us directly or indirectly throughout the completion of this project.

**ABSTRACT**

**Introduction:**

This Application is an Education Portal App forUsers or for fresher’s who wants to make their career in IT industry. Basically This App Is designedfor those who are Not Able to Do Coaching or for Training. This App Helping those in Understanding Modern Programming Technologies Like Java, Hadoopetc. This Application Also Help those to Increase The Capacity Of Answering The Question By Online Test. It also Provide Some Functionality like user can talk to the other user And Suggest Better Answer To them..!!

**Proposed Work:**

* First of all,we will try to overcome the limitations discussed previously.
* We will complete the incomplete modules in the existing project.
* We will introduce the concept of regaining the access in case of Forget Password.
* We will design a strong database using MySQL.
* By developing this software it is easy to generate the report automatically at the end of the session or in the between of the session.

**Software Required:**

**Operating system**: Android Version 4.1.1 or more

(Front End: Android Back End: SQLite)

**Framework**: Android

**Hardware Required:**

Android Phone (with 1GB RAM)

**Members:**

Pravin Yadav,

Krishankant Dubey,

Nalin Sharma

**Table of Contents**

1. Introduction…………………………………………………………...7
2. Problem statement and description…………………………………...7
3. Investigation and data collection…………………………….……….7
4. SRS………………………………………………………….………..8

4.1. Introduction…………………………………………………8

4.1.1. Purpose………………………………………………8

4.1.2. Scope………………………………………...............8

4.1.3. Definitions, Acronyms and Abbreviations………….9

4.1.4. References…………………………………………...9

4.1.5. Overview…….………………………………………9

4. 2. General Description………………………………………...10

4.2.1. Product Perspective…………………………………10

4.2.2. Product Functions…………………………………...11

4.2.3. User Characteristics…………………………………11

4.2.4. General Constraints…………………………………11

4.2.5. Assumptions and Dependencies…………………….11

4.3. Specific Requirements…………………………………........12

4.3.1. Design Constraints………………………..................13

4.3.2. Logical Database Requirements…………………….14

1. System design (DFD) ……………………………………….……….15-16
2. SDD…………………………………………………………………..17-20
3. Database description ……………………………………….………...21-24
4. Coding……………………………………………………….………..25-47
5. Snapshots of forms…………………………………………….…..….47-59

**List of Table Names**

1. login…………………………………………………………..22
2. Quiz……………………………………………......................22

**List of Activities**

1. LoginActivity.java………………………………………………………...26
2. MainActivity.java…………………………………………………………28
3. QuizActivity.java………………………………………………………….30
4. CoursesActivity.java………………………………………………..……..32
5. InterviewQuesions.java……………………………………………………34

**List of Abbreviations**

1. QT Quiz tutorial Application

2. DFD Data Flow Diagram

3. ER Entity Relationship

4. ID Identity

5. SQL Standard Query Language

6. SRS Software Requirement Specification

**INTRODUCTION**

This Application is an Education Portal App forUsers or for fresher’s who wants to make their career in IT industry.Basically This App Is designedfor those who are Not Able to Do Coaching or for Training.This App Helping Us in Understanding Modern Programming Technologies Like java.net.This Application Also Help Us to Increase the Capacity of Answering the Question by Online Test.It also Provide Some Functionality like user can talk to the other user And Suggest Better Answer To them..!!

**PROBLEM STATEMENT AND DISCRIPTION**

There are times, when people like tolearn about the technology and current question for interview and tests then this application are able to make them perfect, apart from staying in touch with books, which we carry . Taking this thought into consideration, we are planning to develop an application which will behave as follows:

This App Is designedfor those who are Not Able to Do Coaching or for Training.This App Helping Us in Understanding Modern Programming Technologies Like java.net.This Application Also Help Us to Increase the Capacity of Answering the Question by Online Test.

**PROPOSED SYSTEM**

This app basically made for the learning purpose where we are trying to provide as per possible content and trying to well design and understable app. We are proposing this system as learner material on hand and from anywhere you can access it.

This app also provides necessary first-aid measures that should be taken at the time of emergency situations. Features:

• Let the Students and friends Learn very large content in on hand

• Provides a Test facility to check there ability

• Let your family and friends know your path to the destination.

**Investigation and data collection**

1. W3schools.com
2. Internet stuffs.
3. Stack Overflow
4. Developer.Android.blogspot.in

**Software Requirement Specification**

**INTRODUCTION**

This Application is an Education Portal App forUsers or for fresher’s who wants to make their career in IT industry.Basically This App Is designedfor those who are Not Able to Do Coaching or for Training.This App Helping Us in Understanding Modern Programming Technologies Like java.net.This Application Also Help Us to Increase the Capacity of Answering the Question by Online Test.It also Provide Some Functionality like user can talk to the other user And Suggest Better Answer To them..!!

1.1 **Purpose**

There are times, when people like to know the current location of his/her friend/(s) or colleague/(s), apart from staying in touch, which we do in social networking applications. Taking this thought into consideration, we are planning to develop an application which will behave as follows:

This application will provide user with his/her friend’s location using GPS (Global Positioning System). It will provide global position of that device itself the user is holding, and through satellite. The premises of application are not premised up to this extend only. It also facilitates user to make new friends that are using that particular application and are internet.

**1.2** **Scope**

* Be able to handle user's login from the android using username and password.
* Be able to recognize user sessions after the login.
* Allow to check if a username is already being used.
* Allow the creation of new users.
* Allow the clients to get the logged in user's information.
* Allow the clients to change their own information.
* Allow an administrator to get any user's positions.
* Allow the user to send his own comment.
* Allow the clients to search for other users in the system.
* Allow the administrator to get all the users in the system with their information.

## 1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirement Specification

TUT: Tutorials

BD: Big Data

## 1.4 References

* Revision of IEEE Std.830-1993)
* IEEE Std.828-1998, IEEE Standard for Software Configuration ManagementPlans.
* IEEE Std. 610.12-1990, IEEE Standard Glossary of Software EngineeringTerminology.
* IEEE Std. 730-1998, IEEE Standard for Software Quality Assurance Plans.

**1.5 Overview**

The application offers an ability to work with location sensitive information. It will allow the user to login/register to the system. The user can press the alert button which will send the text to the listed contacts. He can select particular friend from his friend list and can trace his/her current location, provided that he owes Android GPS based mobile phone and his GPS facility should be activated. Application gives surety that user’s personal and location based information is never shared without users permission. For accessing this application, user has to be connected through internet.

**2. The General Description**

This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

## 2.1 Product Perspective

The Quiz tut is an independent stand–alone system. It is totally self-contained.

2.1.1 **Hardware Interfaces**

The quiz tut shall run on Android Based System.

### 2.1.2 Software Interfaces

Interfacing with several module in order to perform various operation. it require database connectivity (SQL), server interfacing.

- Communication Interfaces

- Memory Constraints

- Operations

- Site Adaptation Requirements

**2.2Product Functions**

Adding information by Administrate

1 add new User information

2 add new question by administrate

**Update information**

1. Update existing information

2. Update question information

3. Update result

**Generate Reports**

1. Date wise and subject wise/section wise.
2. Result details with given range on the basis of percentage.
3. Four consecutive wrong ans warning message and report
4. Result marks

**2.3 User Characteristics**

Educational level of QTL computer software – Low

Experience of QTL software – None

Technical Expertise – Little

**2.4 General constraints**

The Quiz tut shall be a stand-alone system running in aAndroid. The system shall be developed using Java and MySQL or Oracle database.

**2.5 Assumptions and Dependencies**

The system is not required to save generated reports.

**3. Specific Requirements**

This section contains all the software requirements at a level of detail, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

### 3.1 Logical Database Requirements

The logical database requirements include the retention of the following data elements. This list is not a complete list and is designed as a starting point for development.

**Tutorial Information**

Subject name

**User ID**

Subject teach in which Section and course

**User Information**

User Name

User ID

User’s Password

Courses

### 3.2 Design Constraints

The shallQuiz learning be a stand-alone system running in an Android environment. The system shall be developed using Android studio.

We use the following constraints:

**NOT NULL**: Indicates that a column cannot store null value

**PRIMARY KEY**: A combination of a not null and unique. It helps to find the record in a table more easily and quickly.

**FOREIGN KEY**: Ensures the referential integrity of the data in one table to match values in another table.

**CHECK**: Ensure that the values in a column meets a specific condition.

**DEFAULT:** Specifies the default value when specified none for this column .

## 3.3 External Interfaces

The Quiz Tut Learning will use the standard input/output devices for phones &system.

### 3.3.1User Interfaces

### Main Screen: All user actions start from here.

**Login Screen:** User must login before getting any access to the database updating.

**Course Information Screen:** new Course information is entered by administrator and saved to database.

**Result Update Screen:** used to enter or update Result.

### 3.3.2 Software Interfaces

**Operating System** : Android 4.1- later

### 3.3.3 Hardware Interfaces

**Processor** : Intel Pentium 4 1.2GHZ

**Hard Disk** : 60GB

**Memory (Ram )** : 512MB

**System Type**  : 32-bit Operating System

### 3.3.4 Communication Interfaces

There are no special communication interfaces requirements. The system will verify the Username and password, and the user will be considered “Logged-in”. If any new user then registered with some basic required details & then user can “sign in”.

## 3.4 Functional Requirements

Functional requirements define the fundamental actions that system must perform

The functional requirements for the system are divided into three main categories,

Add information, update information and report generation

Adding information by Administrate.

1.Add new User information

2.add new question by administrator

**Update information:**

1. Update existing User information

2. Update course information

3. Update result

**Generate Reports:**

1. Date wise and subject wise/section wise
2. Descending/Ascending Result

3.New test message and report

4.Marks

**3.5 Non-functional Requirements**

Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

### 3.5.1 Performance Requirements

Performance requirements define acceptable response times for system functionality.

* The load time for user interface screens shall take no longer than two seconds.
* The log in information shall be verified within five seconds.
* Queries shall return results within five seconds.

### 3.5.4 Standards Compliance

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.

### 3.5.5 Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery.

### 3.5.6 Availability

The system shall be available during timing.

### 3.5.7 Security

### The Administrator or User can view the reports. And Administrator and User can access and update information but only after passing through login process and only administrator can add new user.

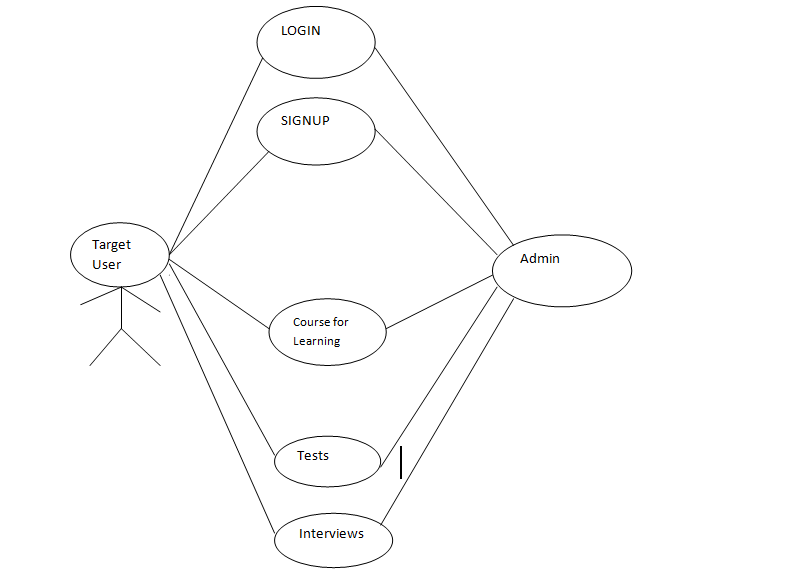
### 3.5.8 Maintainability

The Quiz tut is being developed in Android. Java is an object oriented programming language and shall be easy to maintain.

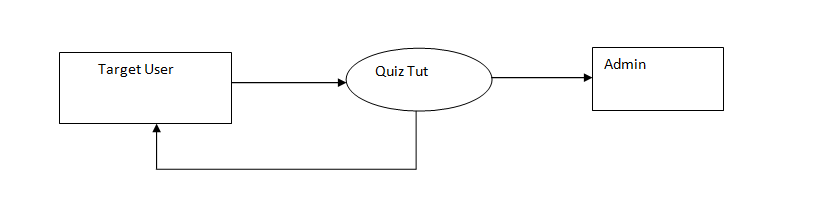
### 3.5.9 Portability

The Quiz tut Learning shall run in any Microsoft Windows environment that contains Android Runtime and the MY SQL.

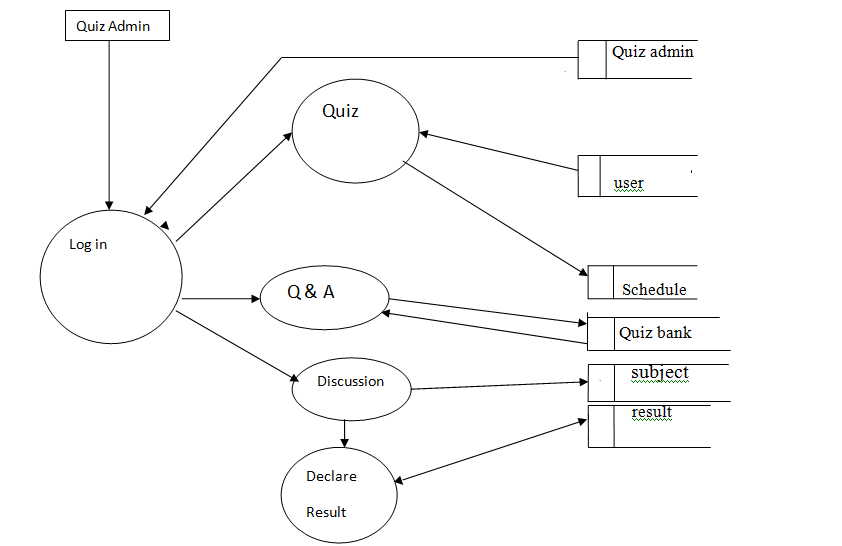
**SYSTEM DESIGN (DFD)**



CONTEXT OR 0 LEVEL DFD :



1 LEVEL DFD:



**SDD**

**SYSTEM DOCUMENT DESIGN**

# INTRODUCTION

This Application is an Education Portal App forUsers or for fresher’s who wants to make their career in IT industry.Basically This App Is designedfor those who are Not Able to Do Coaching or for Training.This App Helping Us in Understanding Modern Programming Technologies Like java.net.This Application Also Help Us to Increase the Capacity of Answering the Question by Online Test.It also Provide Some Functionality like user can talk to the other user And Suggest Better Answer To them..!!

**1.1 Purpose**

There are times, when people like to know the current location of his/her friend/(s) or colleague/(s), apart from staying in touch, which we do in social networking applications. Taking this thought into consideration, we are planning to develop an application which will behave as follows:

This application will provide user with his/her friend’s location using GPS (Global Positioning System). It will provide global position of that device itself the user is holding, and through satellite. The premises of application are not premised up to this extend only. It also facilitates user to make new friends that are using that particular application and are internet.

**1.2 Scope**

* Be able to handle user's login from the android using username and password.
* Be able to recognize user sessions after the login.
* Allow to check if a username is already being used.
* Allow the creation of new users.
* Allow the clients to get the logged in user's information.
* Allow the clients to change their own information.
* Allow an administrator to get any user's positions.
* Allow the user to send his own comment.
* Allow the clients to search for other users in the system.
* Allow the administrator to get all the users in the system with their information.

## 1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirement Specification

GPS: Global Positioning System

GPRS: General Packet Radio Service

## 1.4 References

* Revision of IEEE Std.830-1993)
* IEEE Std.828-1998, IEEE Standard for Software Configuration ManagementPlans.
* IEEE Std. 610.12-1990, IEEE Standard Glossary of Software EngineeringTerminology.
* IEEE Std. 730-1998, IEEE Standard for Software Quality Assurance Plans.

**1.5 Overview**

The application offers an ability to work with location sensitive information. It will allow the user to login/register to the system. The user can press the alert button which will send the text to the listed contacts. He can select particular friend from his friend list and can trace his/her current location, provided that he owes Android GPS based mobile phone and his GPS facility should be activated. Application gives surety that user’s personal and location based information is never shared without users permission. For accessing this application, user has to be connected through internet.

# 2. SYSTEM OVERVIEW

The application offers an ability to work with location sensitive information. It will allow the user to login/register to the system. The user can press the alert button which will send the text to the listed contacts. He can select particular friend from his friend list and can trace his/her current location, provided that he owes Android GPS based mobile phone and his GPS facility should be activated. Application gives surety that user’s personal and location based information is never shared without users permission. For accessing this application, user has to be connected through internet.

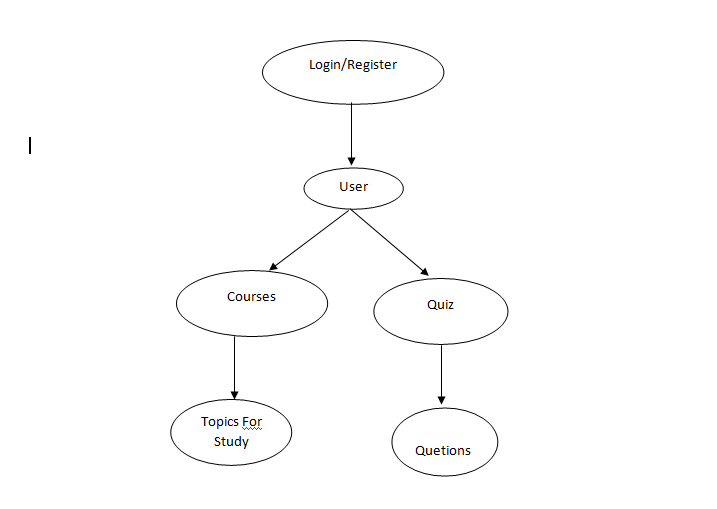
# SYSTEM ARCHITECTURE

**3.1 Architectural Design**

To establish the communication between the components we decided to use the model-view controller. This methodology divides the system into the following blocks:

* Model: The model manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).
* View: The view renders the model into a form suitable for interaction, a user interface element or the way to communicate a component with the others (API). Multiple views can exist for a single model for different purposes.
* Controller: The controller receives input and initiates a response by making calls on model objects. A controller accepts input from the user and instructs the model and view port to perform actions based on that input.

## 3.2 Decomposition Description



**3.3 Design Rationale**

We wanted to build a system such that available from everywhere. Therefore, we designed a Client-Server system. Moreover, we don’t want to restrict the user to his/her device storage size. So, we designed system such that data is kept in a public database. We chose Android for mobile platform because approximately 50% of smartphone in market uses Android for operating system. We wanted to use a RESTful web service because a REST service is:

* Platform-independent (you don't care if the server is Unix , the client is a Mac, or anything else),
* Language-independent.
* Can easily be used in the presence of firewalls.

**Coupling:**

* Data coupling is used between emergency button and send alert text module , add people and interaction module, search location and view location, search and share location, view to location message and share to location message.
* Control coupling is used between search to view and search to share and view to location message and share to location message.

# DATA DESIGN

# 4.1 Database Description:

**List of table:**

* 1. **User Login Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Field** | **Datatype** | **Primary key** |
|  |  |  |  |
| 1 | Username | Varchar(50) | Yes |
| 2 | Password | Varchar(50) | Yes |

* 1. **User registration Table (Personal info):**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Field** | **Datatype** | **Primary key** |
|  |  |  |  |
| 1 | Name | Varchar(50) | Yes |
| 2 | Password | Varchar(50) | Yes |
| 3 | E-mail/Username | Varchar(60) | Yes |

* **LOGIN /REGISTER MODULE:** In this module user has to register first only then he\she is allowed to login and use website.

**CLASSES**:

* + MainActivity (public class ,designing for front page )
  + SignUp

**FUNCTIONS:**

* + Validate(required,custom[integer],maxSize[10],minSize[10])

(used for validating the user name and password)

* **QUIZ MODULE**: In this module user has to press the alert button to send alert message to the listed contact.

**CLASSES:**

* + QuizActivity.java (private class ,designing for main page after login )
  + Question.java

**FUNCTIONS:**

* + Reslult()

(used for Give the Results as per answer reply)

* **COURSE MODULE:** In this module user can view other user location.

**CLASSES:**

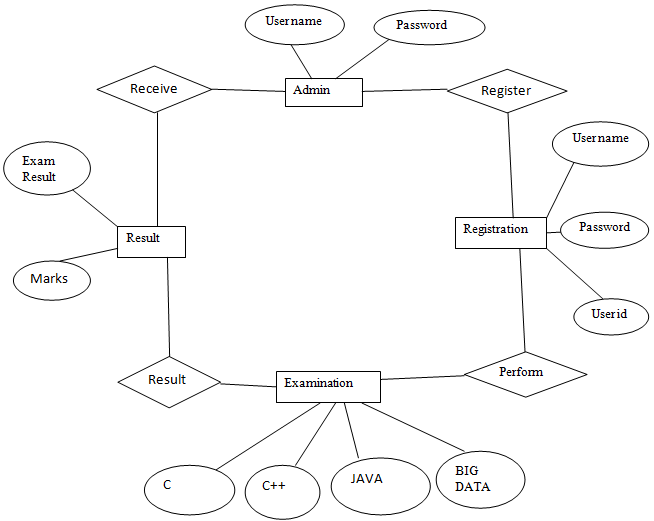
* + Course.java (private class ,designing for search location )

**FUNCTIONS:**

* + function GetApply()

(used to apply for particular course)

## 4.2 ER Diagram



# COMPONENT DESIGN

* All the modules are dependent on login/register module for the functioning.
* Contact of the user is to be used in each module.
* Chat module is functioning for all communication with the other user.
* Travelled distance calculated from search location module.

# HUMAN INTERFACE DESIGN

## 6.1 Overview of User Interface

* **UserInterfaces**

This facilitates the user to interact with the system through a mobile phone. For that the user must have installed the mobile application which is a part of the system. Every detail such as username, passwords, error messages and other notifications are received or initialized through this interface.

* **Hardware interfaces**

|  |  |
| --- | --- |
| **Hardware** | **Specification** |
| CPU | Intel Pentium IV |
| SPEED | 1.5GHz |
| RAM | 1GB |
| HARD DISK | 1GB |
| KEYBOARD | 105 Keys |

* **Software interfaces**

Software is a set of program, documents, and procedure, routines associated with computer system. Software is an essential complement to hardware. It is the computer programs which when “Quiz-Tut Learning” has been developed using the following tools:

* + Computer system
  + Windows
  + Android Studio
  + SQLite database
  + Java and XML files

**Coding**

**Splash screen. Java**

package com.example.android.learningportal;  
public class Splashscreen extends Activity {  
public void onAttachedToWindow() {  
super.onAttachedToWindow();  
 Window window = getWindow();  
 window.setFormat(PixelFormat.*RGBA\_8888*);  
 }  
*/\*\* Called when the activity is first created. \*/*Thread splashTread;  
@Override  
public void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_splashscreen*);  
 StartAnimations();  
 }  
private void StartAnimations() {  
 Animation anim = AnimationUtils.*loadAnimation*(this, R.anim.*alpha*);  
 anim.reset();  
 iv.clearAnimation();  
 iv.startAnimation(anim);  
  
splashTread = new Thread() {  
@Override  
public void run() {  
try {  
int waited = 0;  
*// Splash screen pause time*while (waited <3500) {  
*sleep*(100);  
 waited += 100;  
 }  
 Intent intent = new Intent(Splashscreen.this,Main2Activity.class);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_NO\_ANIMATION*);  
 startActivity(intent);  
 Splashscreen.this.finish();  
 } catch (InterruptedException e) {  
*// do nothing*} finally {  
 Splashscreen.this.finish();  
 }  
  
 }  
 }

**DatabaseHelper.java**

packagecom.example.android.learningportal;  
  
*/\*\*  
 \* Created by Krishna on 10/2/2016.  
 \*/*public class DatabaseHelper extends SQLiteOpenHelper {  
  
public DatabaseHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version){  
super(context,name, factory, version);  
 }  
  
@Override  
public void onCreate(SQLiteDatabase db) {  
  
 db.execSQL(LoginDatabaseAdapter.*TABLE\_CREATE\_QUERY*);  
 }  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
 String query ="DROP TABLE IF EXISTS " + "TEMPLATE";  
 db.execSQL(query);  
this.onCreate(db);  
 }  
  
  
  
}

**LoginDatabaseAdapter.java**

packagecom.example.android.learningportal;  
  
*/\*\*  
 \* Created by Krishna on 10/2/2016.  
 \*/*public class LoginDatabaseAdapter {  
  
static final int *DATABASE\_VERSION* = 114;  
static final String *DATABASE\_NAME* ="contact.db";  
static final String *TABLE\_NAME* = "loginTb";  
static final String *COLUMN\_ID* = "\_id";  
private static final String *COLUMN\_FNAME*="fname";  
private static final String *COLUMN\_LNAME*="lname";  
private static final String *COLUMN\_EMAIL*="email";  
private static final String *COLUMN\_DOB*="dob";  
private static final String *COLUMN\_PWD*="pwd";  
private static final String *COLUMN\_CPWD*="cpwd";  
  
private static final String *COLUMN\_PHNO*="phno";  
static final String *TABLE\_CREATE\_QUERY* = "create table "+*TABLE\_NAME*+" (" +*COLUMN\_ID*+ " INTEGER PRIMARY KEY, "  
+*COLUMN\_FNAME*+ " TEXT NOT NULL," +*COLUMN\_LNAME*+ " TEXT NOT NULL," +*COLUMN\_EMAIL*+ " TEXT NOT NULL,"  
+*COLUMN\_DOB*+ " TEXT NOT NULL," +*COLUMN\_PWD*+ " TEXT NOT NULL," +*COLUMN\_CPWD*+ " TEXT NOT NULL,"  
+*COLUMN\_PHNO*+ " TEXT NOT NULL)";  
  
public SQLiteDatabase db;  
private final Context context;  
  
private DatabaseHelper dbHelper;  
public LoginDatabaseAdapter(Context \_context)  
 {  
this.context = \_context;  
dbHelper = new DatabaseHelper(context,*DATABASE\_NAME*,null,*DATABASE\_VERSION*);  
 }  
public LoginDatabaseAdapter open() throws SQLException  
 {  
db = dbHelper.getWritableDatabase();  
return this;  
 }  
public void close()  
 {  
db.close();  
 }  
  
public SQLiteDatabase getDatabaseInstance()  
 {  
return db;  
 }  
  
public void insertEntry(Compact c)  
 {  
 ContentValues contentValues = new ContentValues();  
  
 contentValues.put(*COLUMN\_FNAME*,c.getFname());  
 contentValues.put(*COLUMN\_LNAME*,c.getlname());  
 contentValues.put(*COLUMN\_EMAIL*,c.getemail());  
 contentValues.put(*COLUMN\_DOB*,c.getdob());  
 contentValues.put(*COLUMN\_PWD*,c.getpwd());  
 contentValues.put(*COLUMN\_CPWD*,c.getcpwd());  
 contentValues.put(*COLUMN\_PHNO*,c.getphno());  
  
db.insert(*TABLE\_NAME*, null, contentValues);  
 }  
  
public String getSingleEntry(String userName)  
 {  
 Cursor cursor=db.query(*TABLE\_NAME*, null, " email=?", new String[]{userName}, null, null, null);  
if(cursor.getCount()<1) *// UserName Not Exist*{  
 cursor.close();  
return "NOT EXIST";  
 }  
 cursor.moveToFirst();  
 String password= cursor.getString(cursor.getColumnIndex(*COLUMN\_PWD*));  
 cursor.close();  
return password;  
 }

**Compact.java**

packagecom.example.android.learningportal;  
  
import static android.R.attr.*name*;  
  
public class Compact {  
int id;  
 String fname,lname,pwd,email,dob,phno,cpwd;  
  
public void setId(int id)  
 {  
this.id=id;  
  
 }  
public int getId()  
 {  
return this.id;  
  
 }  
public void setfname(String fname)  
 {  
  
this.fname=fname;  
 }  
public String getFname()  
 {  
  
return this.fname;  
 }  
  
  
  
public void setlname(String lname)  
 {  
  
this.lname=lname;  
 }  
public String getlname()  
 {  
  
return this.lname;  
 }  
  
  
public void setemail(String email)  
 {  
  
this.email=email;  
 }  
public String getemail()  
 {  
  
return this.email;  
 }  
  
public void setpwd(String pass)  
 {  
  
this.pwd=pass;  
 }  
public String getpwd()  
 {  
  
return this.pwd;  
 }  
  
public void setcpwd(String cpass)  
 {  
  
this.cpwd=cpass;  
 }  
public String getcpwd()  
 {  
  
return this.cpwd;  
 }  
public void setdob(String dob)  
 {  
  
this.dob=dob;  
 }  
public String getdob()  
 {  
  
return this.dob;  
 }  
  
  
  
public void setphno(String phno)  
 {  
  
this.phno=phno;  
 }  
public String getphno()  
 {  
  
return this.phno;  
 }  
  
  
  
}

**DbHelper.java**

package com.example.android.learningportal;  
public class DbHelper extends SQLiteOpenHelper {  
private static final int *DATABASE\_VERSION* = 240;  
*// Database Name*private static final String *DATABASE\_NAME* = "Qbqr";  
*// tasks table name*private static final String *TABLE\_QUEST* = "qqqr";  
*// tasks Table Columns names*private static final String *KEY\_ID* = "id1";  
private static final String *KEY\_QUES* = "question";  
private static final String *KEY\_ANSWER* = "answer"; *//correct option*private static final String *KEY\_OPTA*= "opta"; *//option a*private static final String *KEY\_OPTB*= "optb"; *//option b*private static final String *KEY\_OPTC*= "optc"; *//option c*private static final String *KEY\_OPTD*= "optd";  
private SQLiteDatabase dbase;  
public DbHelper(Context context) {  
super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
@Override  
public void onCreate(SQLiteDatabase db) {  
dbase=db;  
 String sql = "CREATE TABLE IF NOT EXISTS " + *TABLE\_QUEST* + " ( "  
+ *KEY\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, " + *KEY\_QUES*+ " TEXT, "+*KEY\_OPTA* +" TEXT, "  
+*KEY\_OPTB* +" TEXT, "+*KEY\_OPTC*+" TEXT,"+*KEY\_OPTD*+" TEXT, " + *KEY\_ANSWER*+ " TEXT)";  
 db.execSQL(sql);  
 addQuestions();  
*//db.close();*}  
private void addQuestions()  
 {  
 Question q1=new Question("\n" +  
"IBM and \_\_\_\_\_\_\_\_ have announced a major initiative to use Hadoop to support university courses in distributed computer programming.\n",  
"Google Latitude", "Android (operating system)", "Google Variations", "Google","C");  
this.addQuestion(q1);  
 Question q2=new Question("Point out the correct statement : an operating system?", "Hadoop is an ideal environment for extracting and transforming small volumes of data",  
"Hadoop stores data in HDFS and supports data compression/decompression",  
" The Giraph framework is less useful than a MapReduce job to solve graph and machine learning",  
"None of the mentioned","B");  
this.addQuestion(q2);  
 Question q3=new Question("What license is Hadoop distributed under ?","Apache License 2.0", "Mozilla Public License","Shareware","Commercial","A");  
this.addQuestion(q3);  
 Question q4=new Question("Sun also has the Hadoop Live CD \_\_\_\_\_\_\_\_ project, which allows running a fully functional Hadoop cluster using a live CD.",  
"OpenOffice.org", "OpenSolaris", "GNU","Linux","B");  
this.addQuestion(q4);  
 Question q5=new Question("Which of the following genres does Hadoop produce ?","Distributed file system","JAX-RS","Java Message Service","Relational Database Management System","A");  
this.addQuestion(q5);  
 }  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldV, int newV) {  
*// Drop older table if existed*db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_QUEST*);  
*// Create tables again*onCreate(db);  
 }  
*// Adding new question*public void addQuestion(Question quest) {  
*//SQLiteDatabase db = this.getWritableDatabase();*ContentValues values = new ContentValues();  
 values.put(*KEY\_ID*,quest.getID());  
 values.put(*KEY\_QUES*, quest.getQUESTION());  
 values.put(*KEY\_OPTA*, quest.getOPTA());  
 values.put(*KEY\_OPTB*, quest.getOPTB());  
 values.put(*KEY\_OPTC*, quest.getOPTC());  
 values.put(*KEY\_OPTD*, quest.getOPTD());  
 values.put(*KEY\_ANSWER*, quest.getANSWER());  
  
*// Inserting Row*dbase.insert(*TABLE\_QUEST*, null, values);  
 }  
public List<Question> getAllQuestions() {  
 List<Question> quesList = new ArrayList<Question>();  
*// Select All Query*String selectQuery = "SELECT \* FROM " + *TABLE\_QUEST*;  
dbase=this.getReadableDatabase();  
 Cursor cursor = dbase.rawQuery(selectQuery, null);  
*// looping through all rows and adding to list*if (cursor.moveToFirst()) {  
do {  
 Question quest = new Question();  
 quest.setID(cursor.getInt(0));  
 quest.setQUESTION(cursor.getString(1));  
 quest.setOPTA(cursor.getString(2));  
 quest.setOPTB(cursor.getString(3));  
 quest.setOPTC(cursor.getString(4));  
 quest.setOPTD(cursor.getString(5));  
 quest.setANSWER(cursor.getString(6));  
 quesList.add(quest);  
 } while (cursor.moveToNext());  
 }  
*// return quest list*return quesList;  
 }  
public int rowcount()  
 {  
int row=0;  
 String selectQuery = "SELECT \* FROM " + *TABLE\_QUEST*;  
 SQLiteDatabase db = this.getWritableDatabase();  
 Cursor cursor = db.rawQuery(selectQuery, null);  
 row=cursor.getCount();  
return row;  
 }  
}

**Main2Activity.java**

package com.example.android.learningportal;  
  
  
public class Main2Activity extends AppCompatActivity {  
  
 LoginDatabaseAdapter loginDatabaseAdapter;  
 EditText userEdit, passEdit;  
  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main2*);  
  
loginDatabaseAdapter = new LoginDatabaseAdapter(this);  
loginDatabaseAdapter = loginDatabaseAdapter.open();  
  
 }  
  
public void Clicked(View v)  
 {  
if(v.getId()==R.id.*loginb*)  
 {  
userEdit = (EditText) findViewById(R.id.*username*);  
 String str = userEdit.getText().toString();  
passEdit = (EditText) findViewById(R.id.*view*);  
 String pass = passEdit.getText().toString();  
  
 String storedPassword =loginDatabaseAdapter.getSingleEntry(str);  
*//String password=helper.searchpass(str);*if(pass.equals(storedPassword))  
 {  
 Intent intent = new Intent(this,MainActivity.class);  
 intent.putExtra("username", str);  
*makeText*(Main2Activity.this, "Successfully Login", Toast.*LENGTH\_SHORT*).show();  
 startActivity(intent);  
  
 }  
else  
{  
 Toast temp = *makeText*(this, "Incorrect Username or Password", Toast.*LENGTH\_SHORT*);  
 temp.show();  
 }  
  
  
 }  
  
if(v.getId() == R.id.*signupButton*)  
 {  
  
 Intent i = new Intent(Main2Activity.this, Registers.class);  
 startActivity(i);  
 }  
  
  
 }  
  
@Override  
protected void onDestroy() {  
super.onDestroy();  
*// Close The Database*loginDatabaseAdapter.close();

}  
}

**Registers.java**

package com.example.android.learningportal;  
  
public class Registers extends AppCompatActivity {  
  
 LoginDatabaseAdapter loginDatabaseAdapter;  
*//DatabaseHelper helper=new DatabaseHelper(this);*EditText fnameEdit, lnameEdit, emailEdit, dobEdit, passEdit, confirmEdit, phEdit;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_registers*);  
loginDatabaseAdapter=new LoginDatabaseAdapter(this);  
loginDatabaseAdapter=loginDatabaseAdapter.open();  
  
  
 }  
  
public void onSignupClicked(View v)  
 {  
if(v.getId() == R.id.*signupButton*)  
 {  
  
fnameEdit = (EditText) findViewById(R.id.*fname\_editText*);  
lnameEdit = (EditText) findViewById(R.id.*lname\_editText*);  
emailEdit = (EditText) findViewById(R.id.*email\_editText*);  
dobEdit = (EditText) findViewById(R.id.*dob\_editText*);  
passEdit = (EditText) findViewById(R.id.*pass\_editText*);  
confirmEdit = (EditText) findViewById(R.id.*confirm\_pass\_editText*);  
phEdit = (EditText) findViewById(R.id.*phno\_editText*);  
  
 String fname = fnameEdit.getText().toString();  
 String lname = lnameEdit.getText().toString();  
 String email = emailEdit.getText().toString();  
 String dob = dobEdit.getText().toString();  
 String pass = passEdit.getText().toString();  
 String confirmPass = confirmEdit.getText().toString();  
 String phone = phEdit.getText().toString();  
  
if(fname.equals("")||lname.equals("")||email.equals("")||dob.equals("")||pass.equals("")||confirmPass.equals("")||phone.equals(""))  
 {  
 Toast.*makeText*(getApplicationContext(), "Field Vaccant", Toast.*LENGTH\_LONG*).show();  
return;  
 }  
*// check if both password matches*if(!pass.equals(confirmPass))  
 {  
 Toast.*makeText*(getApplicationContext(), "Password does not match", Toast.*LENGTH\_LONG*).show();  
return;  
 }  
else  
{  
 Compact compact = new Compact();  
 compact.setfname(fname);  
 compact.setlname(lname);  
 compact.setemail(email);  
 compact.setdob(dob);  
 compact.setpwd(pass);  
 compact.setcpwd(confirmPass);  
 compact.setphno(phone);  
  
loginDatabaseAdapter.insertEntry(compact);  
 Toast.*makeText*(getApplicationContext(), "Account Successfully Created ", Toast.*LENGTH\_LONG*).show();  
 }  
  
 }  
 }  
  
}

**ResultActivity.java**

package com.example.android.learningportal;  
  
public class ResultActivity extends Activity {  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_result*);  
*//get rating bar object*RatingBar bar=(RatingBar)findViewById(R.id.*ratingBar1*);  
*//get text view*TextView t=(TextView)findViewById(R.id.*textResult*);  
 TextView t1=(TextView)findViewById(R.id.*textScore*);  
*//get score  
 // Bundle b = getIntent().getExtras();  
// int score= b.getInt("score");  
  
 // makeText(this, ""+score, Toast.LENGTH\_SHORT).show();  
//display score*int score=5;  
 bar.setRating(score);  
switch (score)  
 {  
case 1:t.setText("Oopsie! Better Luck Next Time!");  
case 2: t.setText("Oopsie! Better Luck Next Time!");  
t1.setText("Your Score ="+score+"/5");  
break;  
case 3:  
 t.setText("Hmmmm.. Someone's been reading a lot of Quiz");  
t1.setText("Your Score ="+score+"/5");  
break;  
  
case 4:t.setText("Hmmmm.. Someone's been reading a lot of Quiz");  
t1.setText("Your Score ="+score+"/5");  
break;  
case 5:t.setText("Who are you? A Quiz wizard???");  
t1.setText("Your Score ="+score+"/5");  
break;  
 }  
 }  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
*//Inflate the menu; this adds items to the action bar if it is present.*getMenuInflater().inflate(R.menu.*activity\_result*, menu);  
return true;  
 }  
}

**XML Code :**

**Splashscreen.xm**l

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout\_width="fill\_parent"  
android:layout\_height="fill\_parent"  
android:background="#242729"  
android:layout\_gravity="center"  
android:id="@+id/lin\_lay"  
android:gravity="center"  
android:orientation="vertical"  
>  
<ImageView  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:id="@+id/splash"  
android:background="@drawable/splash\_img" />  
  
</LinearLayout>

**Loginactivity.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:paddingBottom="@dimen/activity\_vertical\_margin"  
android:paddingLeft="@dimen/activity\_horizontal\_margin"  
android:paddingRight="@dimen/activity\_horizontal\_margin"  
android:paddingTop="@dimen/activity\_vertical\_margin"  
android:id="@+id/login"  
tools:context="com.example.android.learningportal.Main2Activity">  
  
<TextView  
android:layout\_width="200dp"  
android:layout\_height="50dp"  
android:textAppearance="?android:attr/textAppearanceLarge"  
android:textAlignment="center"  
android:textSize="40dp"  
android:text="WELCOME"  
android:id="@+id/welcome"  
android:layout\_gravity="center\_horizontal"  
android:layout\_weight="0.02"  
android:layout\_alignParentTop="true"  
android:layout\_centerHorizontal="true"  
android:layout\_marginTop="40dp" />  
  
<Button  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:text="LogIn"  
android:id="@+id/loginb"  
android:layout\_marginTop="44dp"  
android:background="@color/accent\_material\_dark"  
android:layout\_below="@+id/view"  
android:layout\_alignStart="@+id/textView4"  
android:layout\_alignLeft="@+id/textView4"  
android:onClick= "Clicked"  
/>  
<Button  
android:layout\_width="80dp"  
android:layout\_height="45dp"  
android:text="SignUp"  
android:layout\_alignTop="@+id/loginb"  
android:layout\_toEndOf="@+id/textView2"  
android:background="@color/accent\_material\_dark"  
android:layout\_toRightOf="@+id/textView2"  
android:id="@+id/signupButton"  
android:onClick="Clicked"  
/>  
</RelativeLayout>

**Register.xml**

*<?*xml version="1.0" encoding="utf-8" *?>*<ScrollView android:id="@+id/ScrollView01"  
android:layout\_alignParentTop="true"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:layout\_above="@+id/LinearLayout01"  
xmlns:android="http://schemas.android.com/apk/res/android"  
>  
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
android:orientation="vertical"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:background="@color/white"  
android:weightSum="1">  
  
<TextView  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:paddingBottom="20dp"  
android:paddingTop="80dp"  
android:text="Sign Up"  
android:textStyle="bold"  
android:textSize="25dp"  
android:textAlignment="center"  
/>  
  
<LinearLayout  
android:layout\_width="fill\_parent"  
android:layout\_height="wrap\_content"  
android:weightSum="2"  
>  
<Button  
android:id="@+id/signupButton"  
android:layout\_width="0dp"  
android:layout\_height="50dp"  
  
android:layout\_weight="1"  
android:layout\_marginTop="30dp"  
  
android:textStyle="bold"  
android:textSize="20dp"  
android:text="SignUp"  
android:layout\_marginRight="10dp"  
android:layout\_marginLeft="10dp"  
android:onClick="onSignupClicked" />  
<Button  
android:id="@+id/cancelButton"  
android:layout\_width="0dp"  
android:layout\_height="50dp"  
android:background="@color/colorAccent"  
android:layout\_weight="1"  
android:layout\_marginTop="30dp"  
</LinearLayout>

</ScrollView>

**Courses.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:paddingBottom="@dimen/activity\_vertical\_margin"  
android:paddingLeft="@dimen/activity\_horizontal\_margin"  
android:paddingRight="@dimen/activity\_horizontal\_margin"  
android:paddingTop="@dimen/activity\_vertical\_margin"  
android:background="#bfff00"  
  
tools:context="com.example.android.learningportal.Course">  
  
<TextView  
android:layout\_width="match\_parent"  
android:layout\_height="50dp"  
android:background="@color/white"  
android:text="COURSE"  
android:textAlignment="center"  
android:textSize="40dp"  
android:textAllCaps="true"  
android:textColor="@color/colorPrimaryDark"  
android:layout\_centerHorizontal="true"  
android:id="@+id/quize"  
/>  
<TextView  
android:layout\_below="@+id/quize"  
android:layout\_width="match\_parent"  
android:layout\_height="80dp"  
android:id="@+id/space"  
  
/>  
  
<Button  
  
  
android:layout\_below="@+id/space"  
android:layout\_width="200dp"  
android:onClick="Clickly"/>  
  
  
<TextView  
android:layout\_below="@+id/java"  
android:layout\_width="match\_parent"  
android:layout\_height="50dp"  
android:id="@+id/space2"  
  
/>  
  
  
<Button  
  
android:id="@+id/hadoop"  
android:layout\_below="@+id/space2"  
android:layout\_width="200dp"  
android:layout\_height="50dp"  
android:layout\_centerHorizontal="true"  
android:background="#ff0000"  
android:text="HADOOP"  
android:onClick="Clickly"  
/>  
<TextView  
android:layout\_width="wrap\_content"  
android:layout\_height="30dp"  
android:textColor="@color/colorPrimary"  
android:layout\_centerHorizontal="true"  
  
android:layout\_alignParentBottom="true"  
  
/>  
</RelativeLayout>

**Quiz**

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:paddingBottom="@dimen/activity\_vertical\_margin"  
android:background="@color/cyan"  
  
tools:context="com.example.android.learningportal.quiz">  
<TextView  
android:layout\_width="match\_parent"  
android:layout\_height="50dp"  
android:background="@color/white"  
android:text="Quiz"  
android:textAlignment="center"  
android:textSize="40dp"  
android:textAllCaps="true"  
android:textColor="@color/colorPrimaryDark"  
android:layout\_centerHorizontal="true"  
android:id="@+id/quize"  
/>  
<TextView  
android:layout\_below="@+id/quize"  
android:layout\_width="match\_parent"  
android:layout\_height="80dp"  
android:id="@+id/space"  
  
/>  
  
<Button  
  
  
android:layout\_below="@+id/space"  
android:layout\_width="200dp"  
android:layout\_height="50dp"  
android:layout\_centerHorizontal="true"  
android:background="@color/white"  
android:text="JAVA"  
android:id="@+id/javaQuiz"  
android:onClick="Clicked"/>  
  
  
<TextView  
android:layout\_below="@+id/javaQuiz"  
android:layout\_width="match\_parent"  
android:layout\_height="50dp"  
android:id="@+id/space2"  
  
/>  
  
  
<Button  
  
android:id="@+id/hadoopQuiz"  
android:layout\_below="@+id/space2"  
android:layout\_width="200dp"  
android:layout\_height="50dp"  
android:layout\_centerHorizontal="true"  
android:background="@color/white"  
android:text="HADOOP"  
android:onClick="Clicked"  
/>  
</RelativeLayout>

**QuizStart**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
<LinearLayout  
<TextView  
android:id="@+id/textView1"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:text="Large Text"  
android:textAppearance="?android:attr/textAppearanceLarge" />  
<RadioGroup  
android:id="@+id/radioGroup1"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:layout\_weight="0.04" >  
<RadioButton  
android:id="@+id/radio0"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:checked="true"  
android:text="RadioButton" />  
<RadioButton  
android:id="@+id/radio1"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:text="RadioButton"  
android:checked="false" />  
<RadioButton  
android:id="@+id/radio2"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:text="RadioButton" />  
<RadioButton  
android:id="@+id/radio3"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:checked="true"  
android:text="RadioButton" />  
</RadioGroup>  
<Button  
android:id="@+id/button1"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:text="@string/str\_next" />  
</LinearLayout>  
</RelativeLayout>

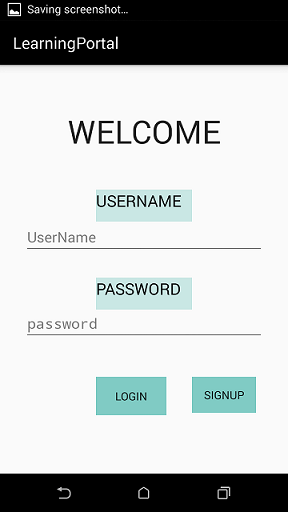
**QuizResult :**

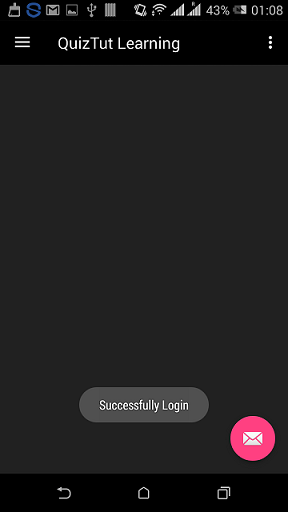
<RelativeLayout  
  
<LinearLayout  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:layout\_alignParentBottom="true"  
android:layout\_alignParentLeft="true"  
android:layout\_alignParentRight="true"  
android:layout\_alignParentTop="true"  
android:orientation="vertical" >  
  
<RatingBar  
android:id="@+id/ratingBar1"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:numStars="5"  
android:stepSize="1.0"  
android:rating="0.0"/>  
<TextView  
android:id="@+id/textScore"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:layout\_weight="0.06"  
android:text="Medium Text"  
android:textAppearance="?android:attr/textAppearanceMedium" />  
</LinearLayout>  
  
</RelativeLayout>

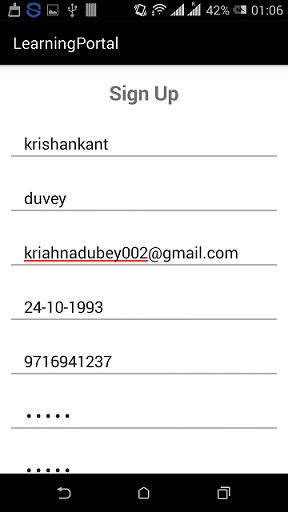
**InterviewQuestion :**

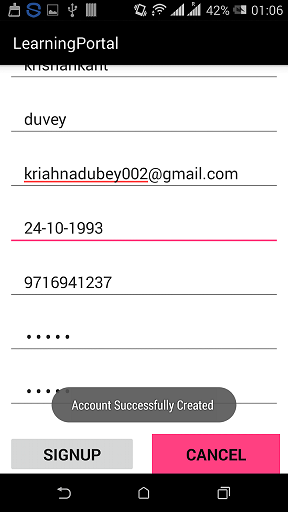
*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
  
tools:context="com.example.android.learningportal.quiz">  
<TextView  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:background="@color/white"  
android:text="InterView Quetions"  
android:textAlignment="center"  
android:textSize="40dp"  
android:textAllCaps="true"  
android:textColor="@color/colorPrimaryDark"  
android:layout\_centerHorizontal="true"  
android:id="@+id/quize"  
/>  
<TextView  
android:layout\_below="@+id/quize"  
android:layout\_width="match\_parent"  
android:layout\_height="80dp"  
android:id="@+id/space"  
  
/>  
  
  
<Button  
  
android:id="@+id/hadoopInterview"  
android:layout\_below="@+id/space2"  
android:layout\_width="200dp"  
android:layout\_height="50dp"  
android:layout\_centerHorizontal="true"  
android:background="#912"  
android:text="HADOOP"  
android:textColor="@color/white"  
android:onClick="Clicked"  
/>  
</RelativeLayout>

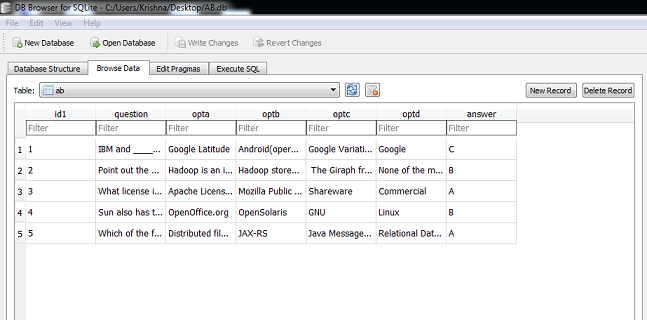
**TestResult**

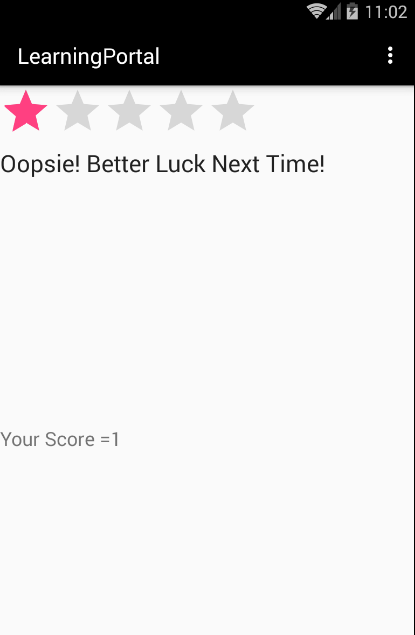
****

****

****

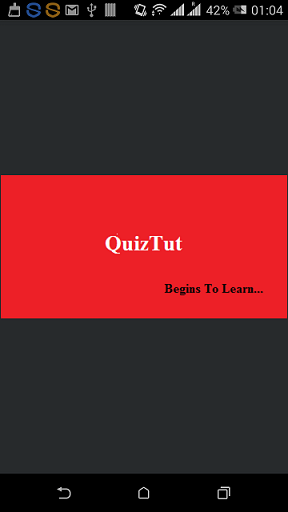
****

****

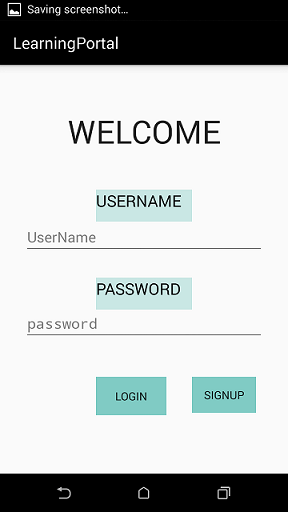
****

**Snapshots of forms**

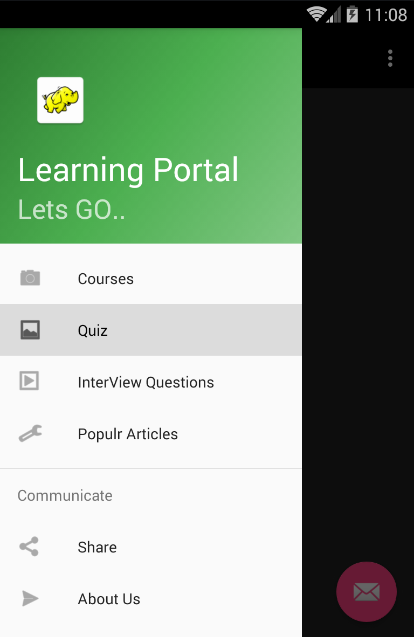
**SplashScreen.xml**

****

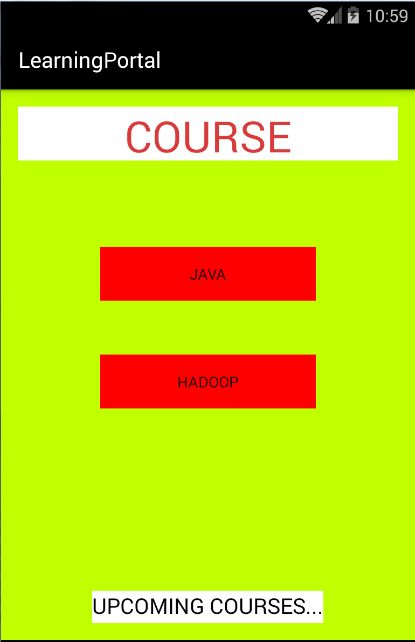
**Login.xml**

****

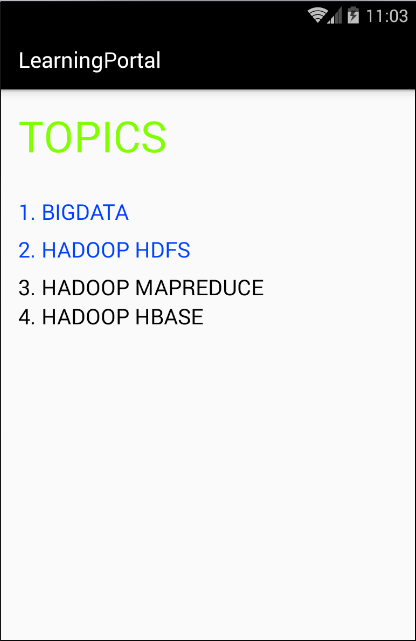
**Navigation.xml**

****

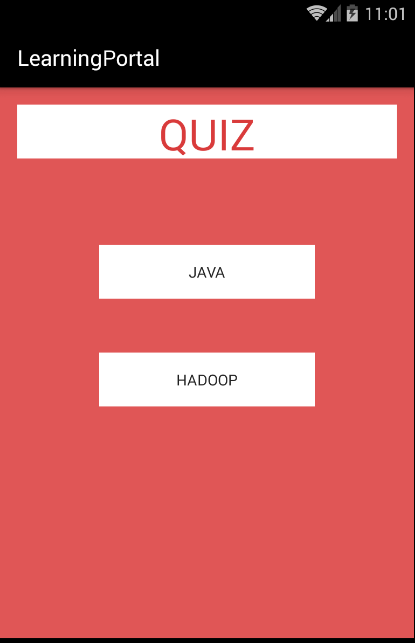
**Courses.xml**

****

**CourseContent.xml**

**l **

**Quiz.xml**

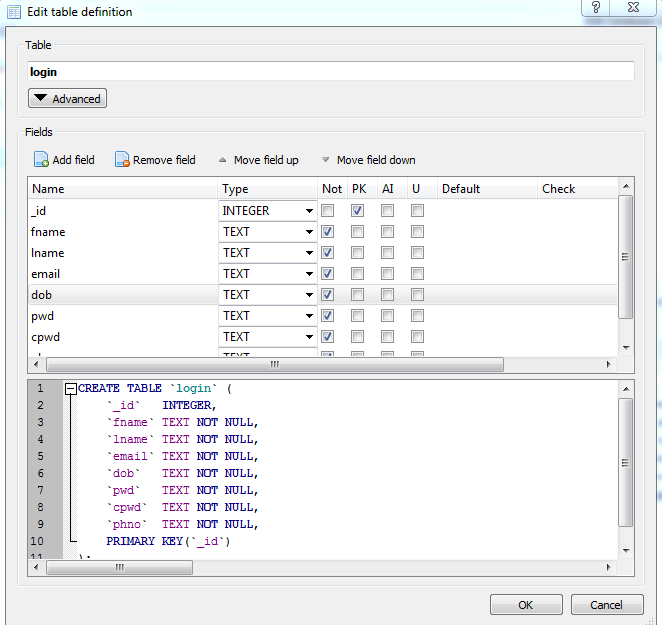
****

**SNAPSHOTS OF TABLE**

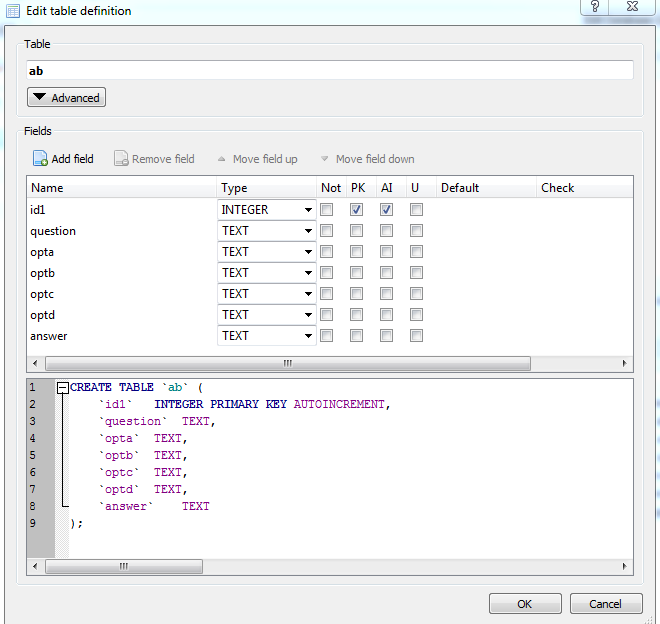
**Database Description**

**Attributes name’s of Table:**

**Login:**

****

**QusetionsTB:**

****