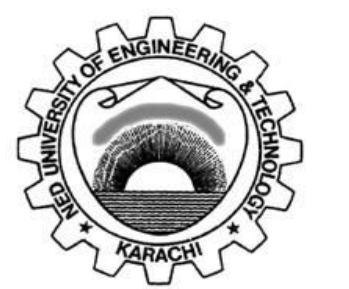
CS-219 - Computer Engineering Workshop

Complex Engineering Problem (CEP) - Report

Fall Semester 2020







### 

### 

### Group Members:

Sohaib Ahmed Abbasi (CS-19096)

Rohan Ahmed (CS-19091)

Abdul Moiz Ali (CS-19087)

Zain Ammad (CS-19079)

# **Table Of Contents**

**Topic Page Number**

[**Project Title**](#_nvriclvk9z6d) **3**

[**Abstract**](#_u8kwz0m19sj5) **3**

[**Details**](#_vbnbcvg8f14n) **3**

[Launcher Icon and Splash Screen](#_5dwk8o99ppkl) 3

[Login Screen](#_7iqz254lt7p6) 3

[Home Screen](#_onrgtuea8luh) 5

[Pomodoro Timer](#_u2omvdz75md3) 5

[Todo List](#_qamikpr3zrus) 6

[Formulas Cheatsheet](#_sngpsth9bhue) 7

[Wikipedia Search](#_oucfgdearw5c) 8

[Google Search](#_6jln8592m0hd) 8

[Useful Links](#_pzmep9ubkdwh) 9

[**Program Flow**](#_k8te88jj42p2) **10**

[**Codes of some important classes**](#_ykk23nknyo55) **10**

[MainActivity.java](#_nz5k16k4sb6) 10

[PomodoroRest.java](#_6pg1a4mz6p3k) 14

[TodoListActivity.java](#_pf8pvncyus6x) 16

[TodoEditorActivity.java](#_pk2v76iqospg) 17

[FormulasCheatSheetActivity.java](#_nt0gf0rm0ko4) 20

[WikipediaSearchActivity.java](#_xbxnqoarg2ae) 21

[UsefulLinks.java](#_8lwqgsz68awj) 21

## 

## Project Title

Studestant

## Abstract

Studestant, short for Student-Assistant, is an android app aimed to facilitate students, boost their productivity, and improve their focus, by providing various useful features in one place. It has the features of To-do List, Pomodoro Timer, Formulas Cheat-Sheet, Wikipedia Search, Google search, and a collection of useful links. The app secures your data with its user account system which uses Firebase authentication, and it includes options to sign in with Email ID and Google ID. Moreover, it uses Firebase's Cloud Firestore as its cloud storage system.

## **Details**

### Launcher Icon and Splash Screen

This is how our app’s icon looks like in a typical launcher (see image on right).

Tapping on the icon opens up the app where you’re first greeted with the following splash screen (see image below):

### 

### 

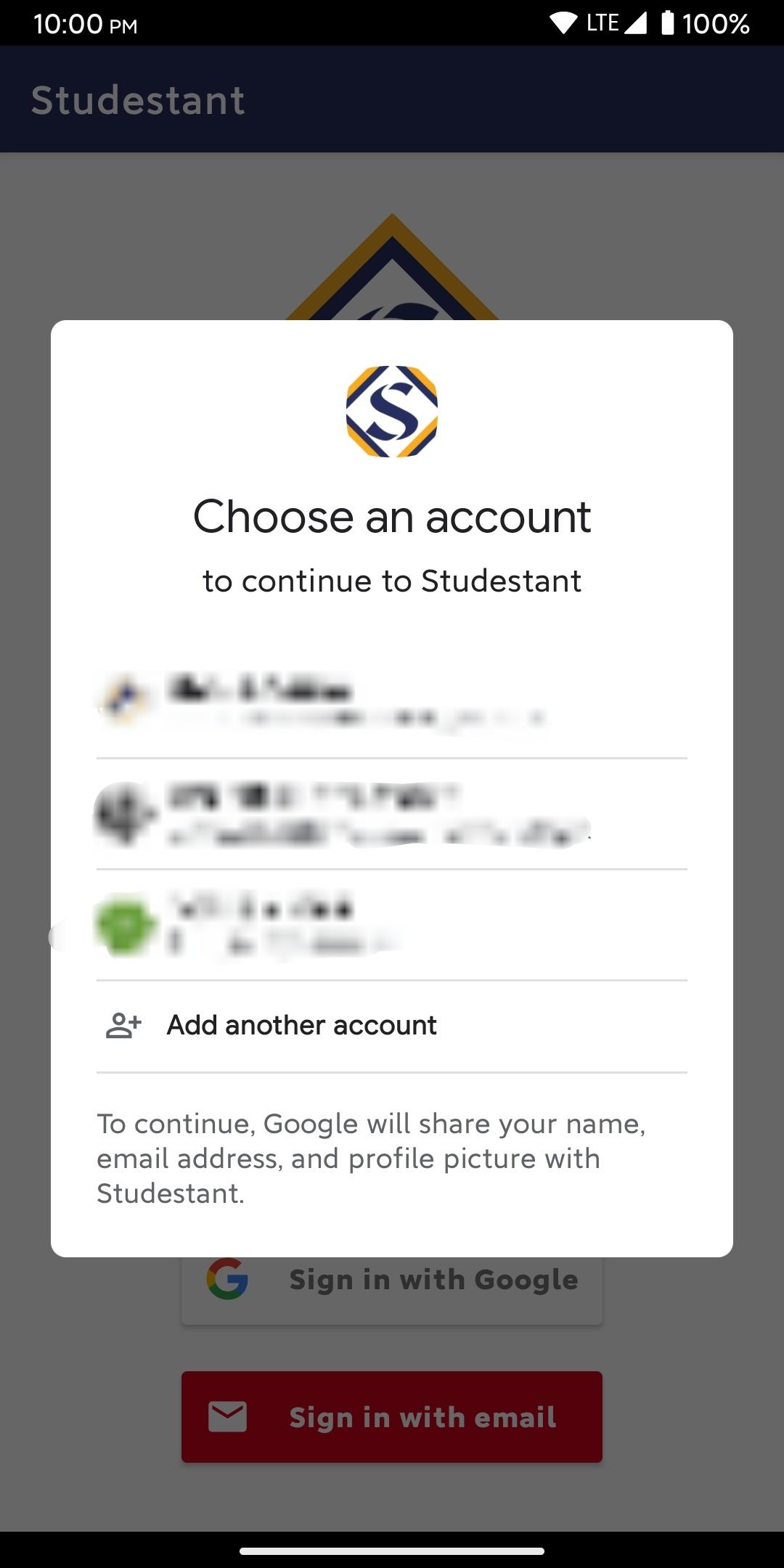
### 

### 

### 

### Login Screen

The app uses Firebase for user authentication as well as the login UI. It has the option to sign in with Google ID and email ID. Tapping the Sign in with Google button opens up a menu with a list of accounts already signed in on the device as well as an option to add another account. Tapping the Sign in with email button opens up an input field where you can type in your email ID. If you type in an email ID that isn’t registered with the app, you will be prompted for your name and a new password, if you type in an email ID that is registered with the app you’ll be prompted for your password; and if you type in an email ID that corresponds to a Google ID that has been used to log in to the app before, you will be asked to sign in with that google ID.



### 

### 

### Home Screen

After logging in, the following home screen opens up (see image on right):

The home screen consists of an options menu in the top right corner which gives you the option to sign out; the app logo; buttons leading to the different features of the app; name and email ID of the signed-in user in the bottom right and a sign out button in the bottom left.

### 

### 

### 

### Pomodoro Timer

This feature helps you to follow the Pomodoro timer which follows the Pomodoro technique which is a technique to improve focus and productivity by having alternate periods of work and rest. It follows the 25-5 pattern of the Pomodoro technique (25 minutes work, 5 minutes rest). First, a screen with details of the Pomodoro timer and how to use it alongside buttons to start work and rest timer is displayed. Tapping the work button starts a 25-minute timer whereas tapping the rest button starts a 5-minute timer; both of these timers end with an alarm sound.

### 

### Todo List

This is to be used to store a list of upcoming tasks that one has to do with optional details of those tasks. It uses the Firestore database to store the information of every user’s to-do items. Tapping the “+” button at the bottom of the screen gives you the option to make a new to-do item with a title and optional details. Tapping on any to-do item opens up a screen where you can see the full to-do item, edit it and delete it. A to-do item can be marked as done or not done by tapping the checkbox on the right of the respective item; marking as done strikethroughs the title and details text.

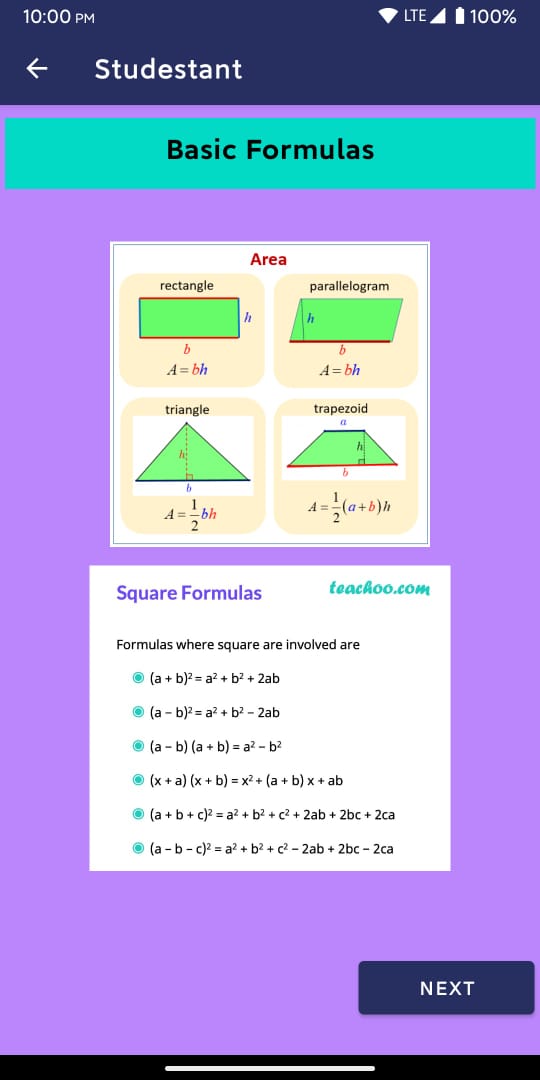
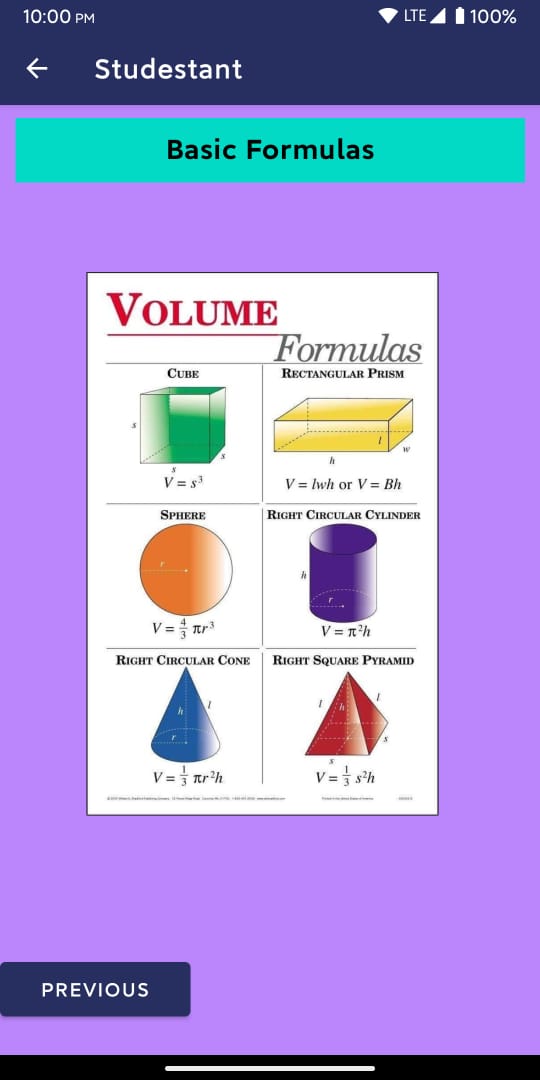
### 

### 

### 

### Formulas Cheatsheet

This part of the app features a bunch of formulas categorized under basic formulas, trigonometric formulas, and physics formulas that you can quickly lookup. It has a screen with buttons that lead you to screens having the respective formula categories.



### 

### 

### Wikipedia Search

This feature can be used to open up any Wikipedia page you want. It has an input field where you can type the name of the Wikipedia page you want to go to and a button that can be tapped to open up the page you want to go in your mobile’s browser. The reason for having this feature in the app is to prevent the user from going into their browser and searching for the Wikipedia page they want because that can easily lead to the user falling to the temptation of going to some other time-wasting / non-productive website or application.

### 

### 

### Google Search

Very similar to the Wikipedia search feature, this feature has a similar purpose and functionality. It can be used to type in and open up the google search page of any thing you want to look up. It can particularly be useful to search up specific definitions, formulas, etc.

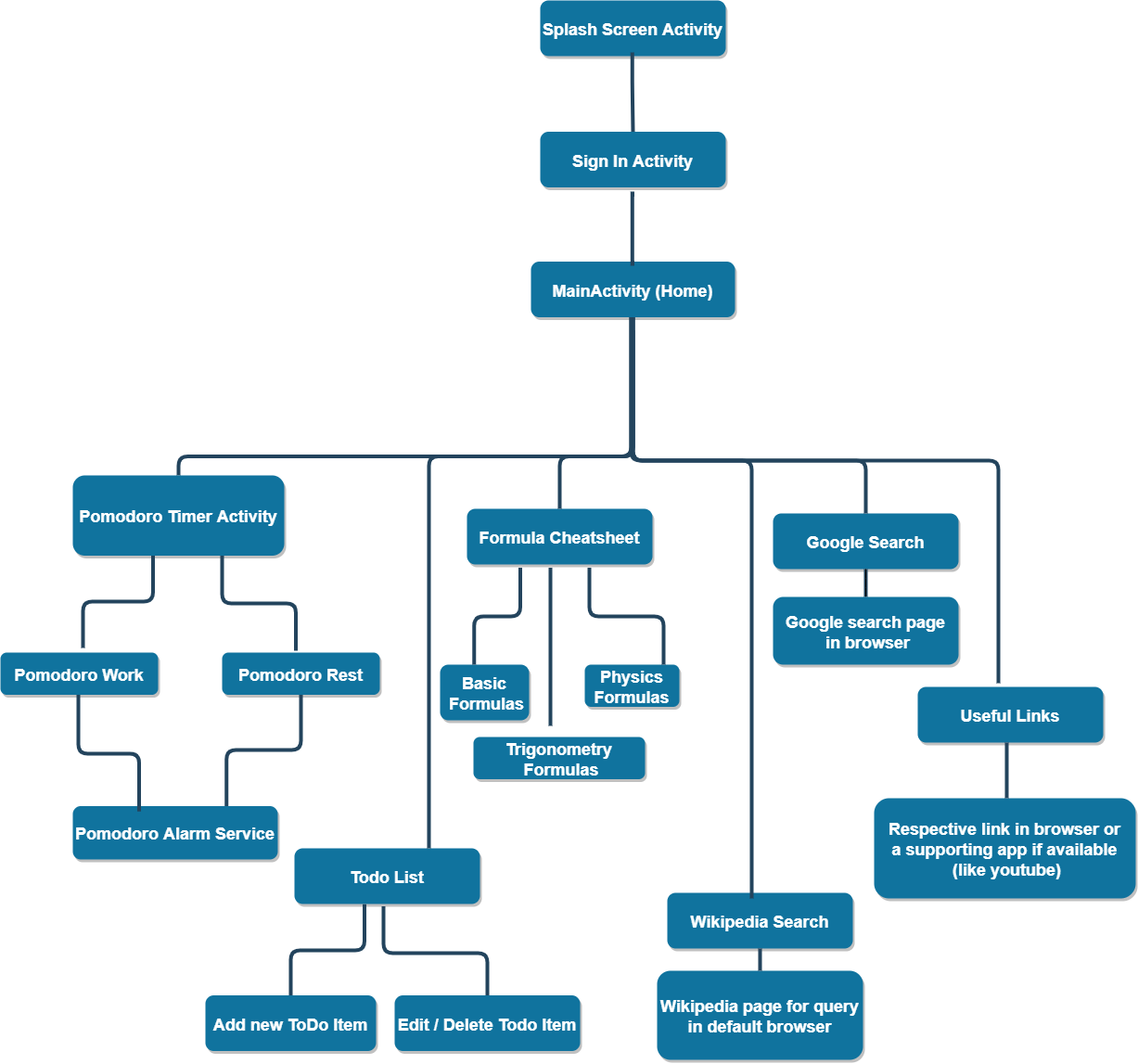
### 

### Useful Links

The useful links section of the app contains a list of links that students may not know about but can be quite useful. Each link can be opened up by clicking its respective “Click here” text.

*(Program flow on next page)*

## Program Flow



## Codes of some important classes

### MainActivity.java

public class MainActivity extends AppCompatActivity {

private static final int RC\_SIGN\_IN = 1;

Button btn\_todo\_list;

Button btn\_wikipedia\_search;

Button btn\_google\_search;

Button btn\_formulas\_cheatsheet;

Button btn\_pomodoro\_timer;

Button btn\_useful\_links;

TextView txtview\_userinfo;

Button btn\_signout;

FirebaseAuth mAuth; // firebase authentication instance

FirebaseAuth.AuthStateListener mAuthStateListener;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

startActivity(new Intent(this, SplashScreenActivity.class));

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO); // Disable darkmode

btn\_todo\_list = findViewById(R.id.btn\_todo\_list);

btn\_wikipedia\_search = findViewById(R.id.btn\_wikipedia\_search);

btn\_google\_search = findViewById(R.id.btn\_google\_search);

btn\_formulas\_cheatsheet = findViewById(R.id.btn\_formulas\_cheatsheet);

btn\_pomodoro\_timer = findViewById(R.id.btn\_pomodoro\_timer);

btn\_useful\_links = findViewById(R.id.btn\_useful\_links);

txtview\_userinfo = findViewById(R.id.txtview\_userinfo);

btn\_signout = findViewById(R.id.btn\_signout);

mAuth = FirebaseAuth.getInstance();

btn\_todo\_list.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, TodoListActivity.class);

startActivity(intent);

}

});

btn\_wikipedia\_search.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, WikipediaSearchActivity.class);

startActivity(intent);

}

});

btn\_google\_search.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, GoogleSearchActivity.class);

startActivity(intent);

}

});

btn\_formulas\_cheatsheet.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, FormulasCheatsheetActivity.class);

startActivity(intent);

}

});

btn\_pomodoro\_timer.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, PomodoroTimerActivity.class);

startActivity(intent);

}

});

btn\_useful\_links.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, UsefulLinks.class);

startActivity(intent);

}

});

btn\_signout.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

mAuth.signOut();

// Auth.GoogleSignInApi.signOut(apiClient); // check this for google signout problems

}

});

List<AuthUI.IdpConfig> providers = Arrays.asList(

new AuthUI.IdpConfig.GoogleBuilder().build(),

new AuthUI.IdpConfig.EmailBuilder().build());

// new AuthUI.IdpConfig.PhoneBuilder().build()

mAuthStateListener = new FirebaseAuth.AuthStateListener() {

@Override

public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {

FirebaseUser current\_user = mAuth.getCurrentUser();

if (current\_user == null) // if user isn't logged in

{

// Create and launch sign-in activity intent

startActivityForResult(

AuthUI.getInstance()

.createSignInIntentBuilder()

.setAvailableProviders(providers)

.setTheme(R.style.Theme\_Studestant)

.setLogo(R.drawable.login)

.setIsSmartLockEnabled(false)

.build(),

RC\_SIGN\_IN);

}

updateUI();

}

};

}

@Override

protected void onResume() {

super.onResume();

// adding authStateListener

mAuth.addAuthStateListener(mAuthStateListener);

}

@Override

protected void onPause() {

super.onPause();

// removing authstateListener

mAuth.removeAuthStateListener(mAuthStateListener);

}

@Override

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

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == RC\_SIGN\_IN) // if result is from firebase signin activity

{

IdpResponse response = IdpResponse.fromResultIntent(data);

if (resultCode == RESULT\_OK) {

Toast.makeText(this, "Successfully signed in", Toast.LENGTH\_SHORT).show();

}

else

{

if (response == null) // user pressed back on login screen

{

finish(); // close the app

}

}

}

}

@Override

public void onBackPressed() {

super.onBackPressed();

finish();

}

private void updateUI() {

FirebaseUser currentUser = mAuth.getCurrentUser();

if (currentUser == null) // if not logged in

{

txtview\_userinfo.setText("User info:\nNot signed in");

btn\_signout.setVisibility(View.GONE);

}

else // if logged in

{

btn\_signout.setVisibility(View.VISIBLE);

String name = currentUser.getDisplayName();

if (TextUtils.isEmpty(name)){

name = "Not available";

}

String email = currentUser.getEmail();

if (TextUtils.isEmpty(email)){

email = "Not available";

}

// set user name and email in text view

txtview\_userinfo.setText(String.format("Signed in as:\nName: %s\nEmail: %s", name, email));

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.home\_activity\_menu, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

switch (item.getItemId())

{

case R.id.signout\_menu\_option:

mAuth.signOut();

return true;

default:

return super.onOptionsItemSelected(item);

}

}

}

### PomodoroRest.java

public class PomodoroRest extends AppCompatActivity {

public static final long start\_time\_ms= 300000;

TextView tv\_time\_to\_rest;

TextView tv\_j\_rest;

CountDownTimer countdown\_timer;

long left\_time= start\_time\_ms;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_pomodoro\_rest);

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO); // Disable darkmode

tv\_time\_to\_rest = findViewById(R.id.tv\_ttr);

tv\_j\_rest= (TextView)findViewById(R.id.tv\_pr);

tv\_time\_to\_rest.setText("Time to rest");

start\_timer();

}

public void start\_timer() {

countdown\_timer= new CountDownTimer(left\_time,1000) { //1000ms= 1 second interval

@Override

public void onTick(long millisUntilFinished) {

left\_time= millisUntilFinished;

update\_countdown\_text(); //updated the text view of timer

}

@Override

public void onFinish() {

Intent intent\_notify = new Intent(PomodoroRest.this, Tone\_Service.class);

startService(intent\_notify); //play the notification Sound

tv\_time\_to\_rest.setText("Time's Up!\nGo back and start the work timer.");

update\_countdown\_text();

new Handler(Looper.myLooper()).postDelayed(new Runnable() {

@Override

public void run() {

stopService(intent\_notify);

}

}, 2500); // stop music player service after 2.5 seconds

}

}.start();

}

public void update\_countdown\_text() {

int minutes = (int) (left\_time / 1000) / 60;

int seconds = (int) (left\_time / 1000) % 60;

String time\_left\_format = String.format(Locale.getDefault(), "%02d:%02d", minutes, seconds);

tv\_j\_rest.setText(time\_left\_format);

}

}

### TodoListActivity.java

public class TodoListActivity extends AppCompatActivity {

private CollectionReference todo\_collection\_reference;

private TodoAdapter adapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_todo\_list);

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO); // Disable darkmode

findViewById(R.id.btn\_add\_note).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(TodoListActivity.this, TodoAdderActivity.class));

}

});

FirebaseUser current\_user = FirebaseAuth.getInstance().getCurrentUser();

if (current\_user == null) // if no user logged in

{

finish(); // end the activity

}

todo\_collection\_reference = FirebaseFirestore.getInstance(). // get an instance of firestore database

collection("users"). // go to the users collection

document(current\_user.getUid()). // go inside the document corresponding to current user

collection("todoItems"); // go to the todoItems collection inside that user's document

setUpRecyclerView();

}

@Override

protected void onResume() {

super.onResume();

if (FirebaseAuth.getInstance().getCurrentUser() == null) // if user not logged in

{

finish(); // end activity

}

adapter.startListening();

}

@Override

protected void onPause() {

super.onPause();

adapter.stopListening();

}

private void setUpRecyclerView()

{

Query query = todo\_collection\_reference.orderBy("isDone", Query.Direction.ASCENDING);

FirestoreRecyclerOptions<TodoItem> options = new FirestoreRecyclerOptions.Builder<TodoItem>()

.setQuery(query, TodoItem.class)

.build();

adapter = new TodoAdapter(options);

RecyclerView recyclerView = findViewById(R.id.recycler\_view\_todo);

recyclerView.setHasFixedSize(true);

recyclerView.setLayoutManager(new LinearLayoutManager(this));

recyclerView.setAdapter(adapter);

}

}

### TodoEditorActivity.java

public class TodoEditorActivity extends AppCompatActivity {

FirebaseUser current\_user;

DocumentReference todo\_item\_reference;

AlertDialog.Builder alert\_dialog\_builder;

AlertDialog delete\_alert\_dialog;

EditText et\_todo\_editor\_title;

EditText et\_todo\_editor\_details;

String doc\_id;

String todo\_title;

String todo\_details;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_todo\_editor);

setTitle("Edit");

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO); // Disable darkmode

current\_user = FirebaseAuth.getInstance().getCurrentUser();

if (current\_user == null) // if not logged in

{

finish();

}

Intent intent = getIntent();

// validation

if ( ! (intent.hasExtra("todo\_document\_id") &&

intent.hasExtra("todo\_title") && intent.hasExtra("todo\_details")))

{

Toast.makeText(this, "Error, couldn't find the todo item.", Toast.LENGTH\_SHORT).show();

finish();

}

doc\_id = intent.getStringExtra("todo\_document\_id");

todo\_title = intent.getStringExtra("todo\_title");

todo\_details = intent.getStringExtra("todo\_details");

todo\_item\_reference =

FirebaseFirestore.getInstance(). // get an instance of firestore database

collection("users"). // go to the users collection

document(current\_user.getUid()). // go inside the document corresponding to current user

collection("todoItems"). // go to the todoItems collection inside that user's to-dolist collection

document(doc\_id); // go to the specific to-do item document

et\_todo\_editor\_title = findViewById(R.id.edittxt\_todo\_editor\_title);

et\_todo\_editor\_details = findViewById(R.id.edittxt\_todo\_editor\_details);

et\_todo\_editor\_title.setText(todo\_title);

et\_todo\_editor\_details.setText(todo\_details);

// building alert dialog for delete

alert\_dialog\_builder = new AlertDialog.Builder(this);

alert\_dialog\_builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

todo\_item\_reference.delete();

Toast.makeText(TodoEditorActivity.this, "Item deleted", Toast.LENGTH\_SHORT).show();

finish();

}

});

alert\_dialog\_builder.setNegativeButton("No", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// pass

}

});

alert\_dialog\_builder.setTitle("Are you sure you want to delete this item?");

alert\_dialog\_builder.setIcon(R.drawable.ic\_warning);

delete\_alert\_dialog = alert\_dialog\_builder.create();

findViewById(R.id.btn\_todo\_editor\_cancel).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish();

}

});

findViewById(R.id.btn\_todo\_editor\_save).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

saveChanges();

}

});

findViewById(R.id.btn\_todo\_editor\_delete).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

delete\_alert\_dialog.show();

}

});

}

private void saveChanges() {

if (current\_user == null) // if no user logged in

{

finish(); // end the activity

}

String updated\_todo\_title = et\_todo\_editor\_title.getText().toString().trim();

if (updated\_todo\_title.equals(""))

{

Toast.makeText(this, "Can't save a todo item without a title.", Toast.LENGTH\_SHORT).show();

return;

}

String updated\_todo\_details = et\_todo\_editor\_details.getText().toString().trim();

// update the to-do item in cloud firestore (which will also automatically update the recyclerview)

todo\_item\_reference.update("title", updated\_todo\_title, "details", updated\_todo\_details);

Toast.makeText(this, "Successfully saved changes", Toast.LENGTH\_SHORT).show();

finish();

}

}

### FormulasCheatSheetActivity.java

public class FormulasCheatsheetActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_formulas\_cheatsheet\_activty);

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO); // Disable darkmode

}

public void basic\_formulas(View view) {

Intent intent=new Intent(FormulasCheatsheetActivity.this,Basic\_Formulas.class);

startActivity(intent);

}

public void trigonometry\_formulas(View view) {

Intent intent=new Intent(FormulasCheatsheetActivity.this,Trigonometry\_Formulas.class);

startActivity(intent);

}

public void physics\_formulas(View view) {

Intent intent=new Intent(FormulasCheatsheetActivity.this,Physics\_Formulas.class);

startActivity(intent);

}

}

### WikipediaSearchActivity.java

public class WikipediaSearchActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_wikipedia\_search);

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO); // Disable darkmode

}

public void btnclick(View view) {

EditText et\_wiki= (EditText)findViewById(R.id.editText\_wiki);

String et\_w=et\_wiki.getText().toString();

if (et\_w.isEmpty()) {

Toast.makeText(this,"You did not write",Toast.LENGTH\_SHORT).show();

}

else {

String url = "https://en.wikipedia.org/wiki/" + et\_w; //making url which is going to be redirected to browser

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse(url));

startActivity(intent);

}

}

}

### UsefulLinks.java

public class UsefulLinks extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_useful\_links);

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO); // Disable darkmode

TextView tv\_moshLink = findViewById(R.id.tv\_moshLink);

tv\_moshLink.setMovementMethod(LinkMovementMethod.getInstance());

TextView tv\_w3Link = findViewById(R.id.tv\_w3Link);

tv\_w3Link.setMovementMethod(LinkMovementMethod.getInstance());

TextView tv\_stackLink = findViewById(R.id.tv\_stackLink);

tv\_stackLink.setMovementMethod(LinkMovementMethod.getInstance());

TextView tv\_grammarlyLink = findViewById(R.id.tv\_grammarlyLink);

tv\_grammarlyLink.setMovementMethod(LinkMovementMethod.getInstance());

TextView tv\_kAcadLink = findViewById(R.id.tv\_kAcadLink);

tv\_kAcadLink.setMovementMethod(LinkMovementMethod.getInstance());

}

}