**TECHNICAL REPORT OF ASSIGNMENT OF CSY2061**

1. Username and password

For admin

email: [admin123@test.com](mailto:admin123@test.com)

Password: admin123

1. List of all the features

|  |  |  |
| --- | --- | --- |
| **Features** | **Implemented** | **Any comments** |
| User registration | Full | User can fully register and there’s no error |
| User login | Full | Anyone that are registered can login and use the application |
| Add a new post | Full | Anyone logged in can add a post without any error |
| View all post | Full | All the post upload can be viewed in home page which comes right after login page |
| View all my post | Full | A user can view all his/her post that he/she added in page called my timeline |
| Update of email and password | Full | A user can update his/her email and password by clicking on update |
| Viewing email and password | Full | A user can view theirs email and password in my profile page |
| Add a comment | Full | A user can add comment in their post and in other’s post. |
| Deleting a post | Full | A post can be deleted by only the admin and a user can delete post if it’s their own post |
| Managing user | Full | Admin can view all user and even delete the profile. |
| Edit a post | Full | A user can edit their’s own post |
| Delete comment | Partial | At the moment only admin can delete their comment. User cannot delete their’s own comment |
| Logout | Full | A user can logout which will lead to login page after |

1. Explanation of Main Fraction of code

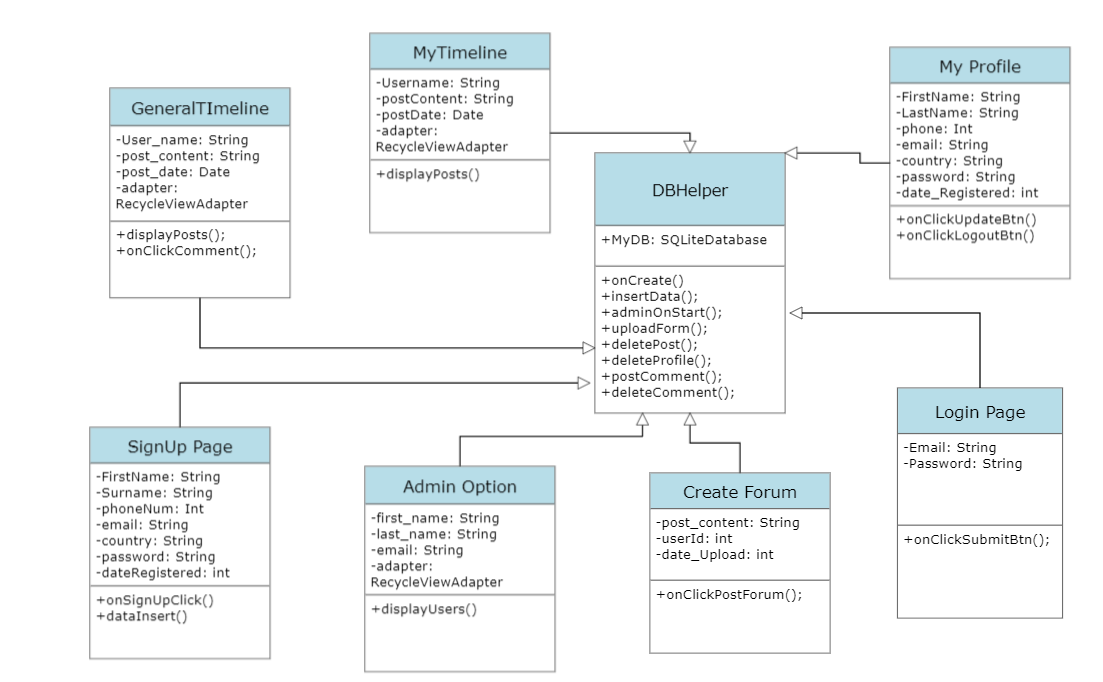


Fig: class diagram

1. Signup

The signup activity allows a user to create a new account by inputting their personal information such as name, email and phone number.The activity then checks if the provided email already exists in the database by using the checkUniqueEmail()

Method. If theemail is unique, the activity proceeds to insert the user’s information into

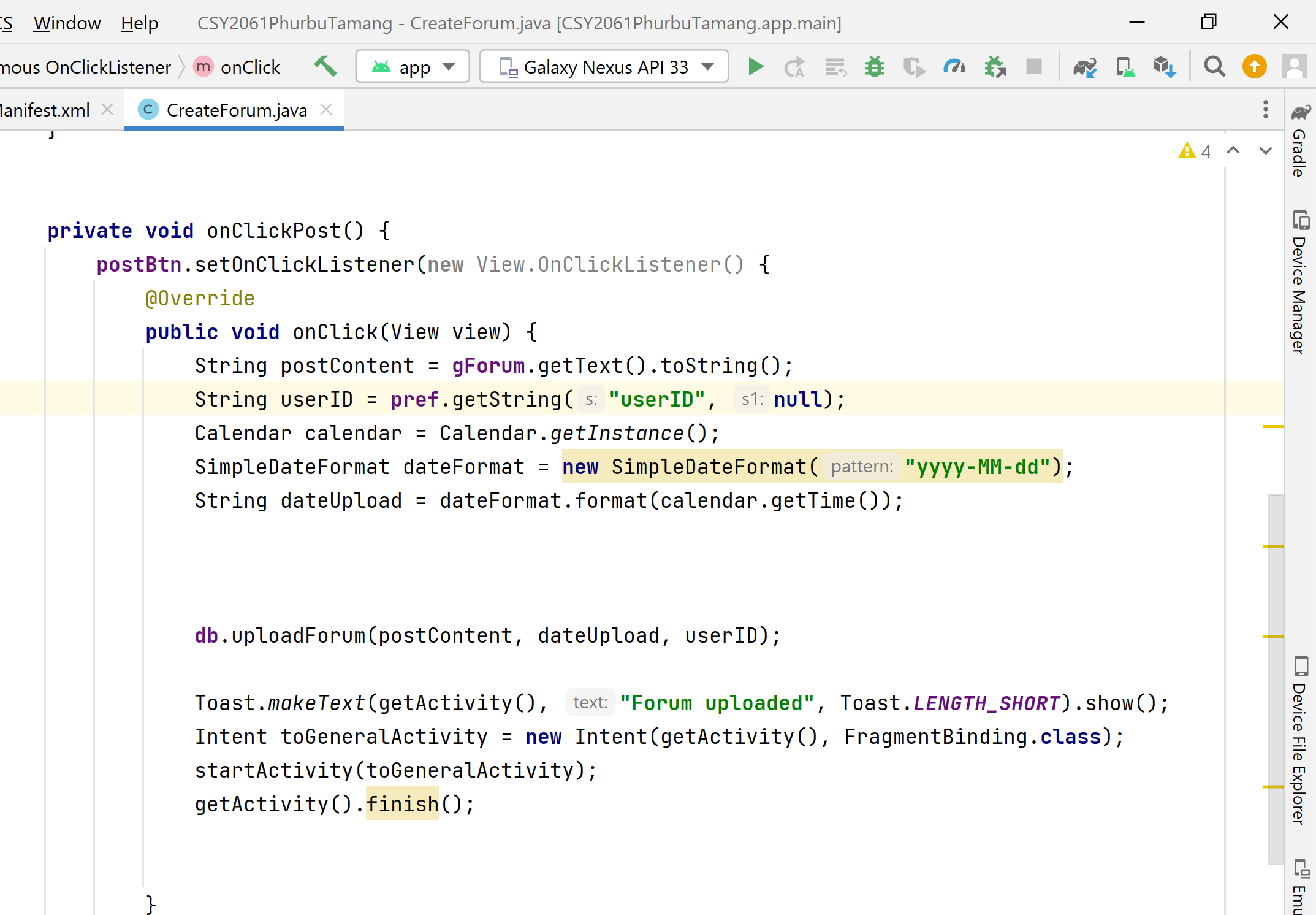
the database, completing the signup process.

1. Login Page

The Login Activity allows a user to acess their existing account by inputting their email and password. When the user clicks the submit button, the activity initiates the verifyEmailPassword() method which compares the entered email and password with the existing information in the database. If the entered email and password atch with the data stored in the databse, the user is granted access to their account and is logged in.

1. CreateForum

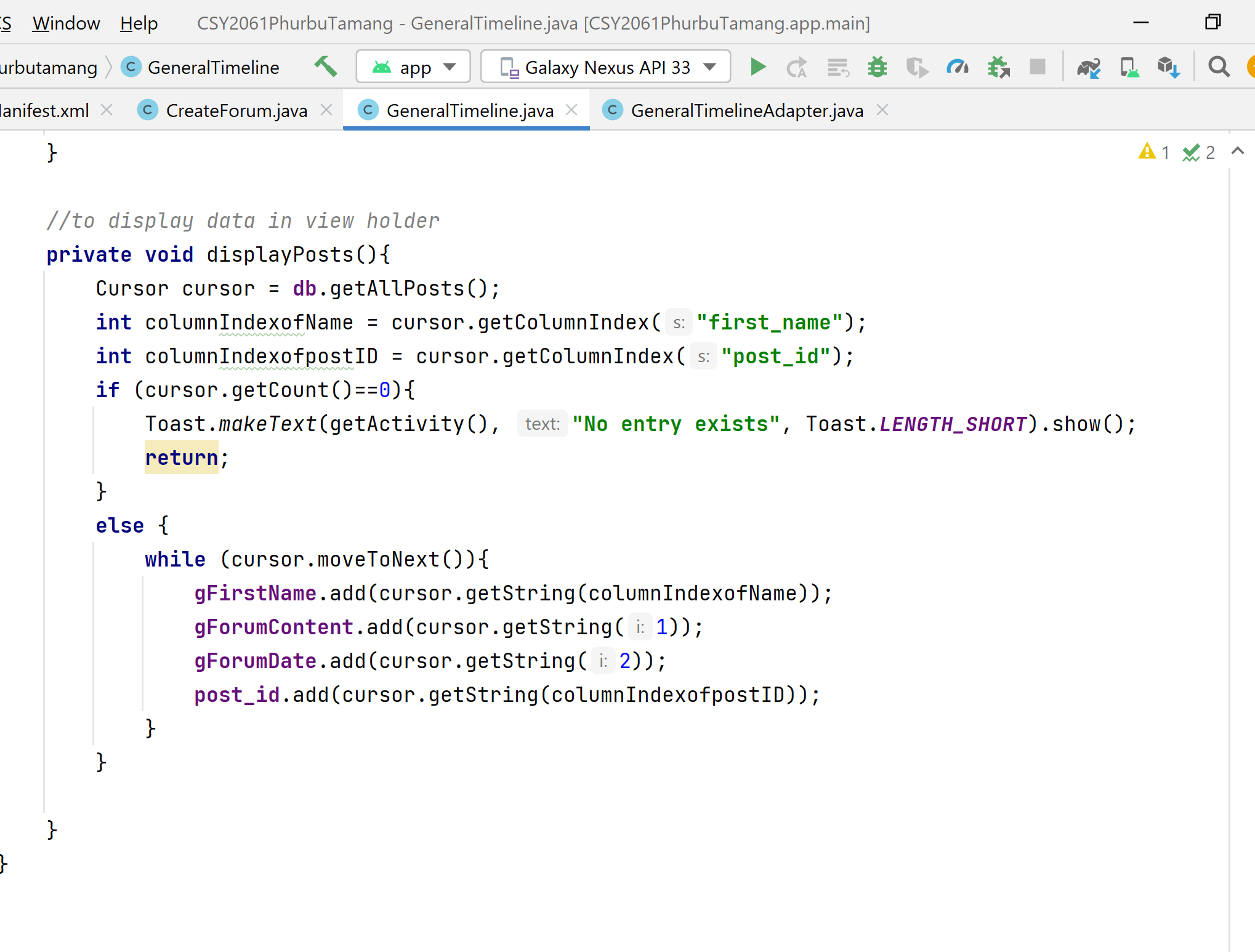
The createForum activity allows a user to create a new post that is visible to other users in the General timeline. The activity allows the user to input the post’s text and/or multimedia content, and then submits the post to the database. The post is then displayed on the General Timeline for others users to see and interact with. It has onClickPost() method which will get the information from editText and uploadForum() will upload it to database.



1. General timeline

The General Timeline section is composed of two parts: an adapter class and the General Timeline activity.

The General Timeline Activity retrieves all the posts from the database and stores them in a cursor. The displayPosts() method is used to retrieve this data and a while loop is used to place the data into an Array String. This Array String is then sent to the General Timeline Adapter, which uses it to populate the General Timeline withn the retrieved data.



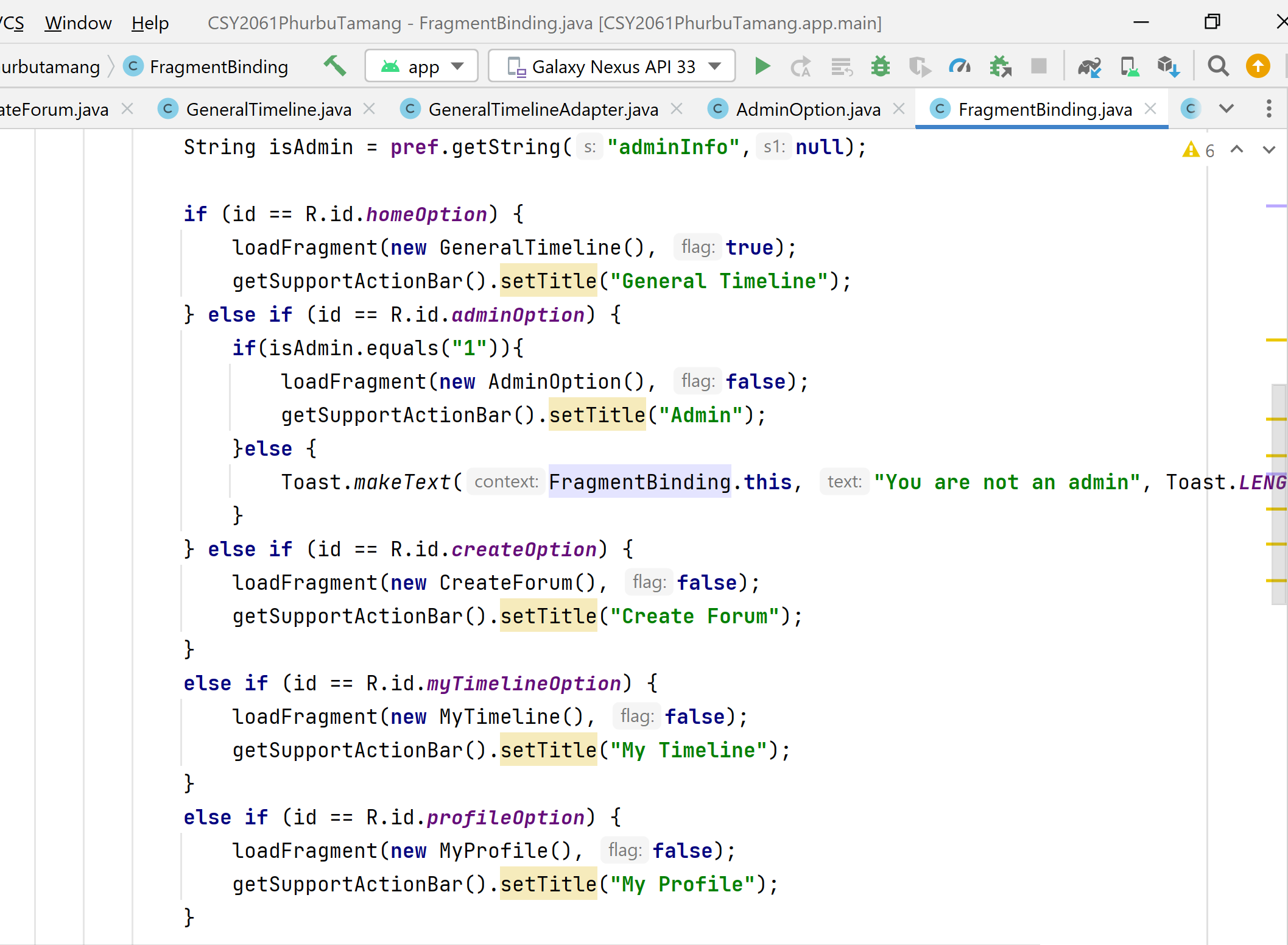
1. General Timeline Adapter

This is an adapter class for the General Timeline activity in an Android application. The class is used to display a list of posts retrieved from a database in the desired format on the General Timeline.

The class has several methods that are used to bind the data from the database to the appropriate views within the “posttemplateuser” cardview. The onBindViewHolder() method binds the data to the views, the isUserAdmin() method checks if the logged-in user is administrator or not. The setClickListener() method redirects the user to another activity when they click on a post.

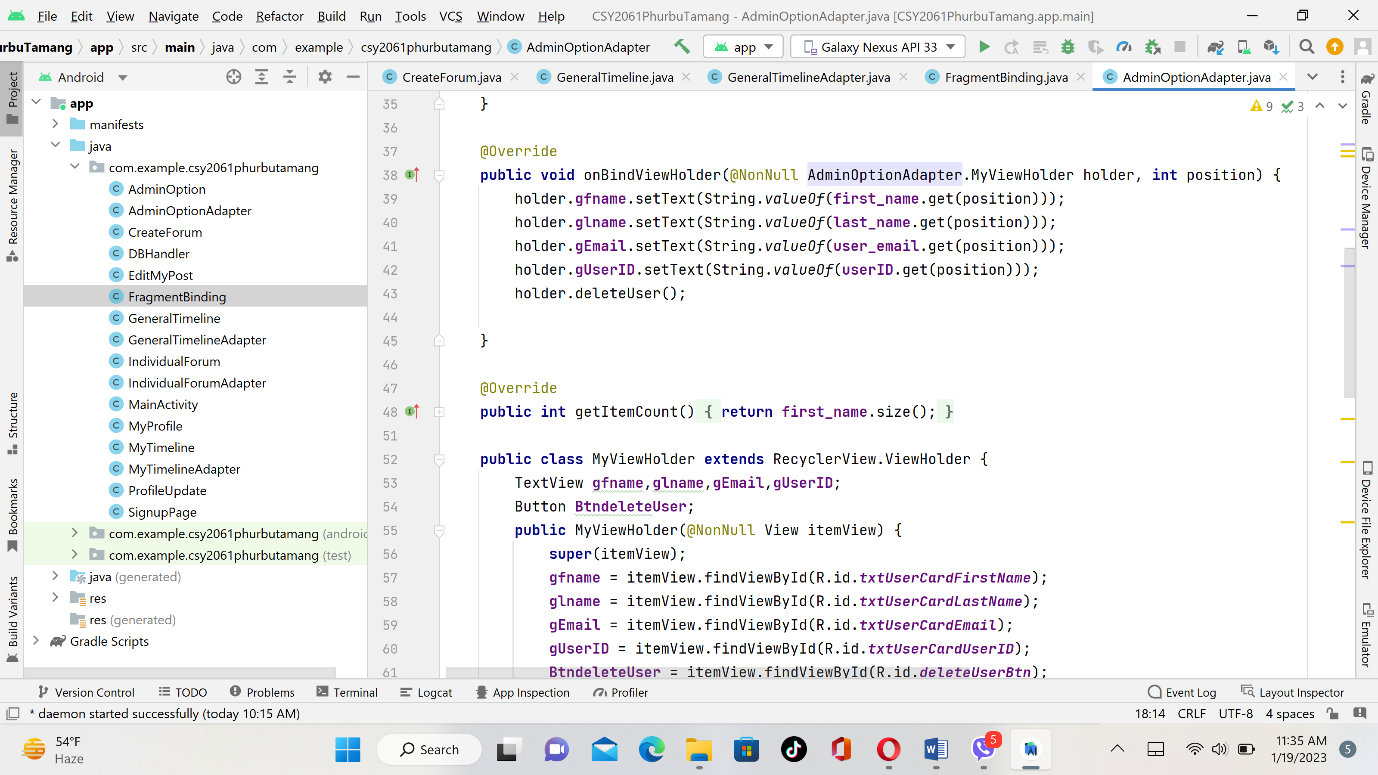
1. Admin Option

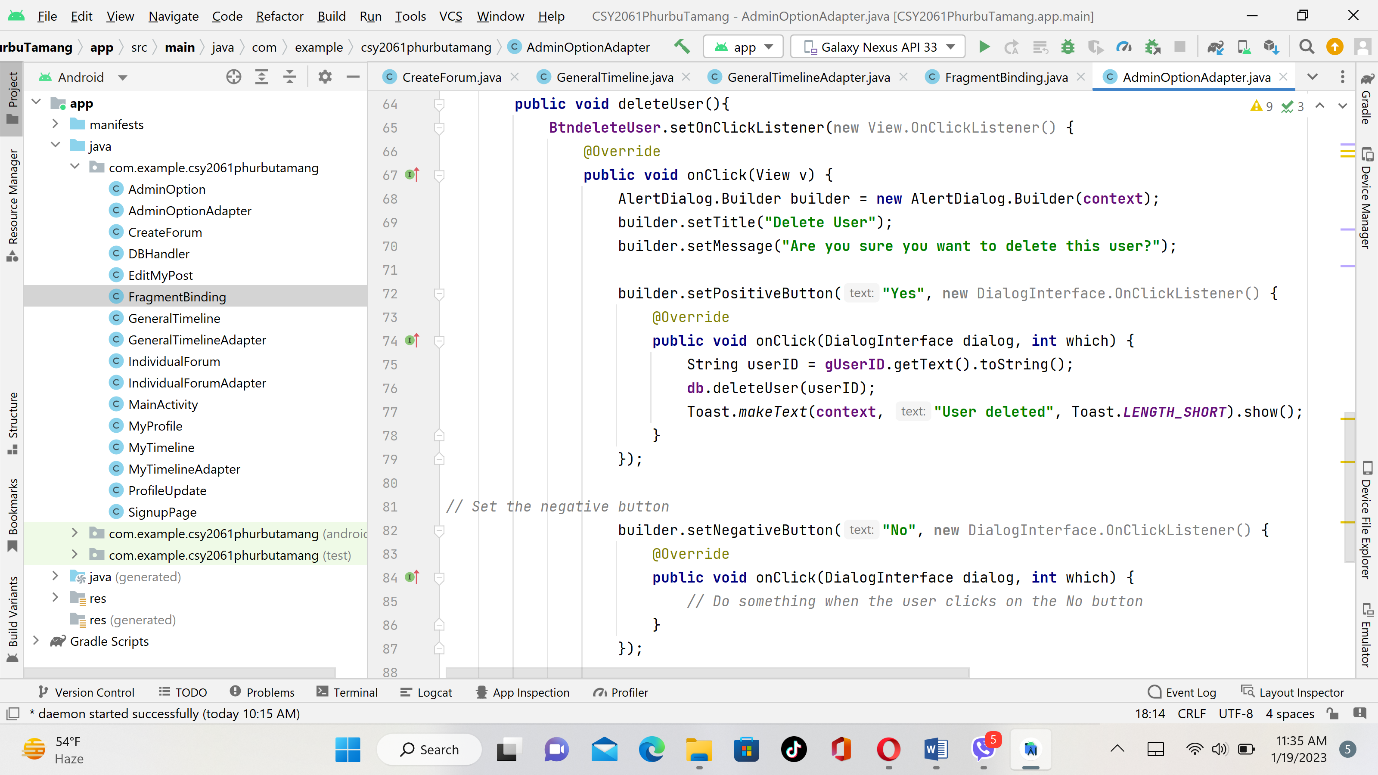
This class makes us of two classes, an Adapter class and an Admin option class. Access to this activity is restricted to only users with administrator privileges. To validate the user’s admin status ,the FragmentBinder activity retrieves the admin status from SharedPreferences and compares it. The code snippet provided serves as an example of how the validation is performed in the FragementBinder activity.



1. Admin option adapter

The Admin Option Adapter class receives data from the AdminControl Class by using a constructor. The constructor is a special type of method that is used to create and initialize an object.When an object of the AdminOptionAdapter class is created,the constructor is called, and it takes data from the AdminOption class as an argument. This data is then used by the adapter to populate the views in the RecyclerView in the Admin option activity.

The AdminOptionAdapter class also includes a delete user button, which is only visible to the admin. When this button is clicked by the admin, it executes the deleteUser() method. This method is responsible for executing a delete query in SQLite, which results in deleting a specific user from database.



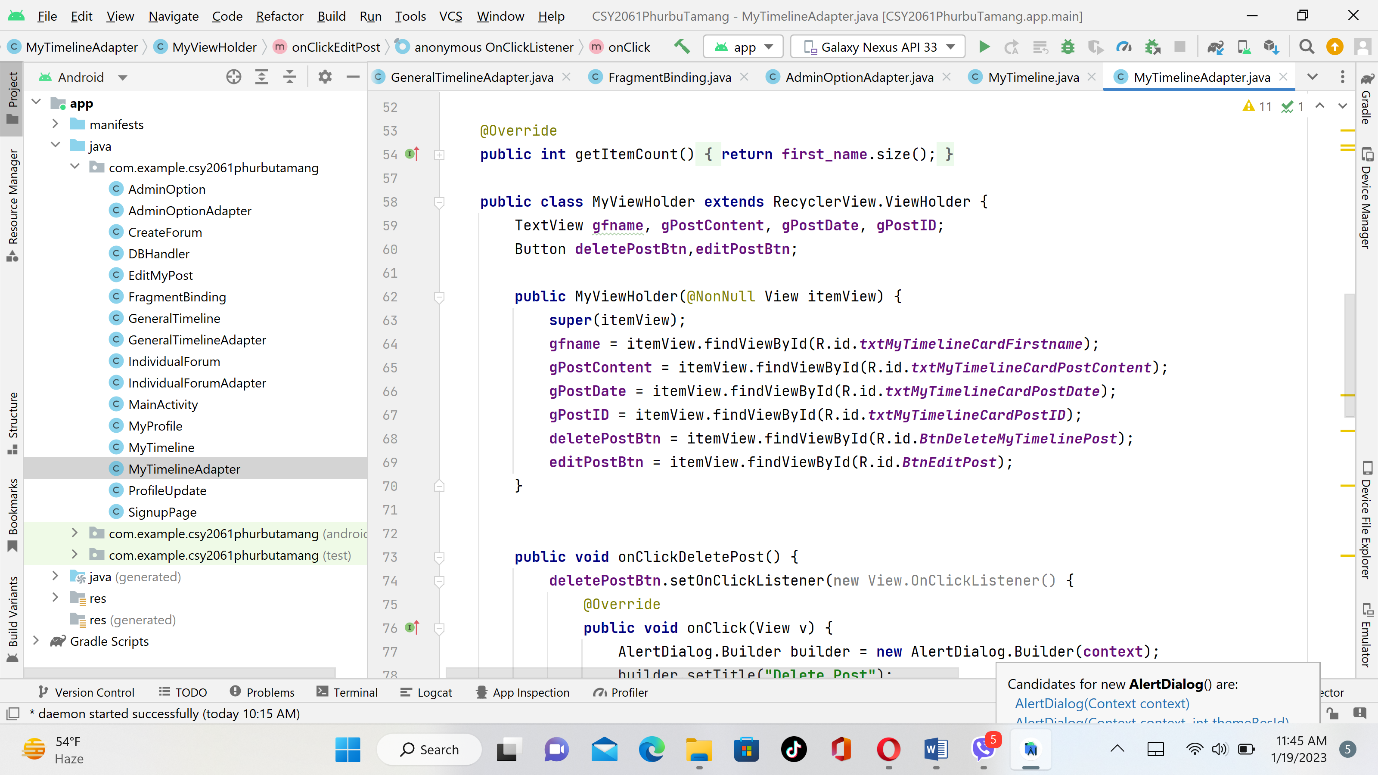
1. My Timeline

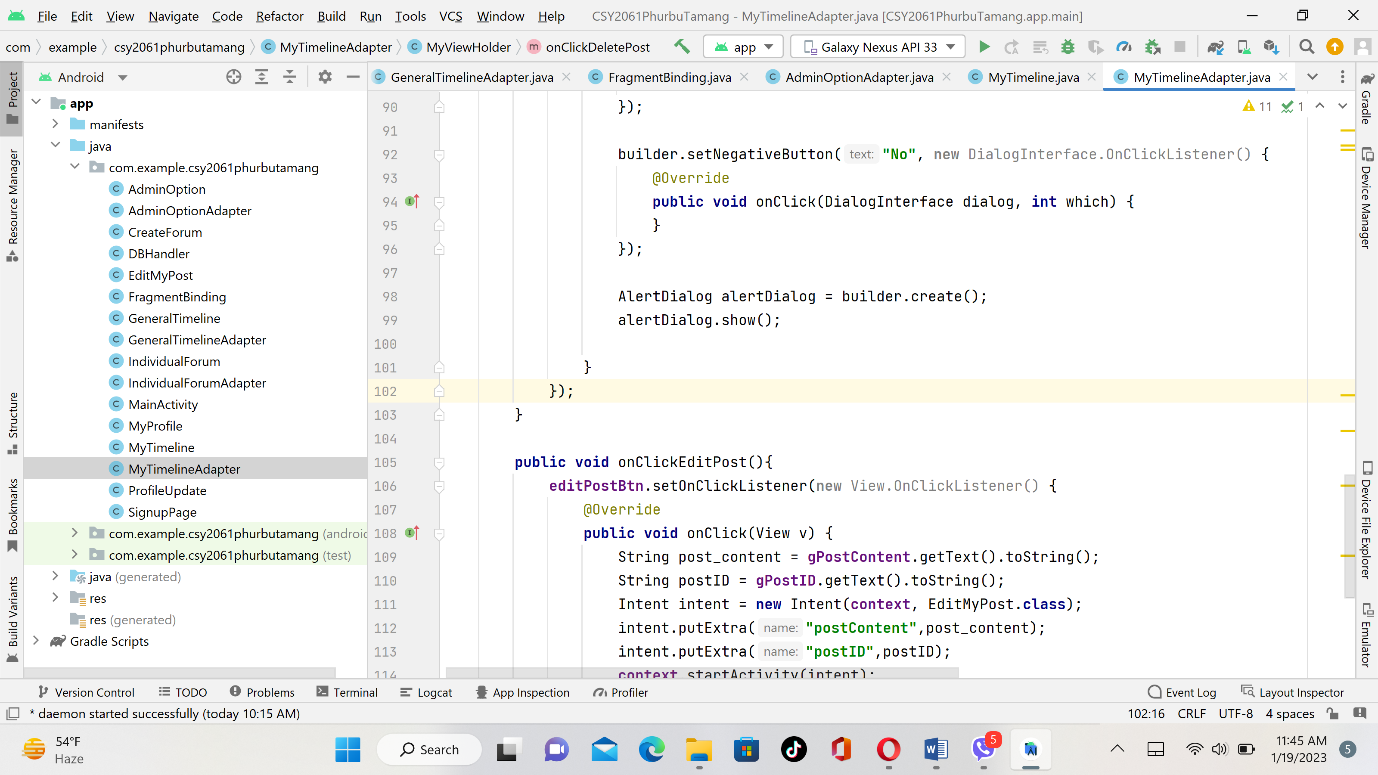
This class is responsible for displaying all the posts that the currently logged-in user has created. This class allows the user to delete or edit their own posts. The class retrieves all the post data from the database using a cursor and stores it in an Array<String> using

displayPosts() method. It then sends this data to MyTimelineAdapter class through the constructor.

1. My Timeline Adapter

The MyTimelineAdapter class will use this data to display the posts in the desired format on the My Timeline activity. The onBindViewHolder() method is used to bind the data extracted from the Array<Struing> to the appropriate views within the “posttemplateuser” cardview. The onClickDeletePost() method is used to delete the post when the delete post butoon is clicked. It shows a dialog box to confirm the deletion and if the user confirms, it deletes the post by calling the deleteForum() method and updates the recycler view.





1. My profile

This class is responsible for displaying the logged-in user’s profile details such as email, password . It also provides a feature that allows the user to update their information. When the onClickUpdateBtn() method is triggered, it takes all the new user input and updates the data in the database. The new information is then displayed on the profile after the activity is refreshed.

1. dbHandler

The class diagram provided indicates that the dbHnadler class is important for almost every other classes

1. system key’s feature screenshot

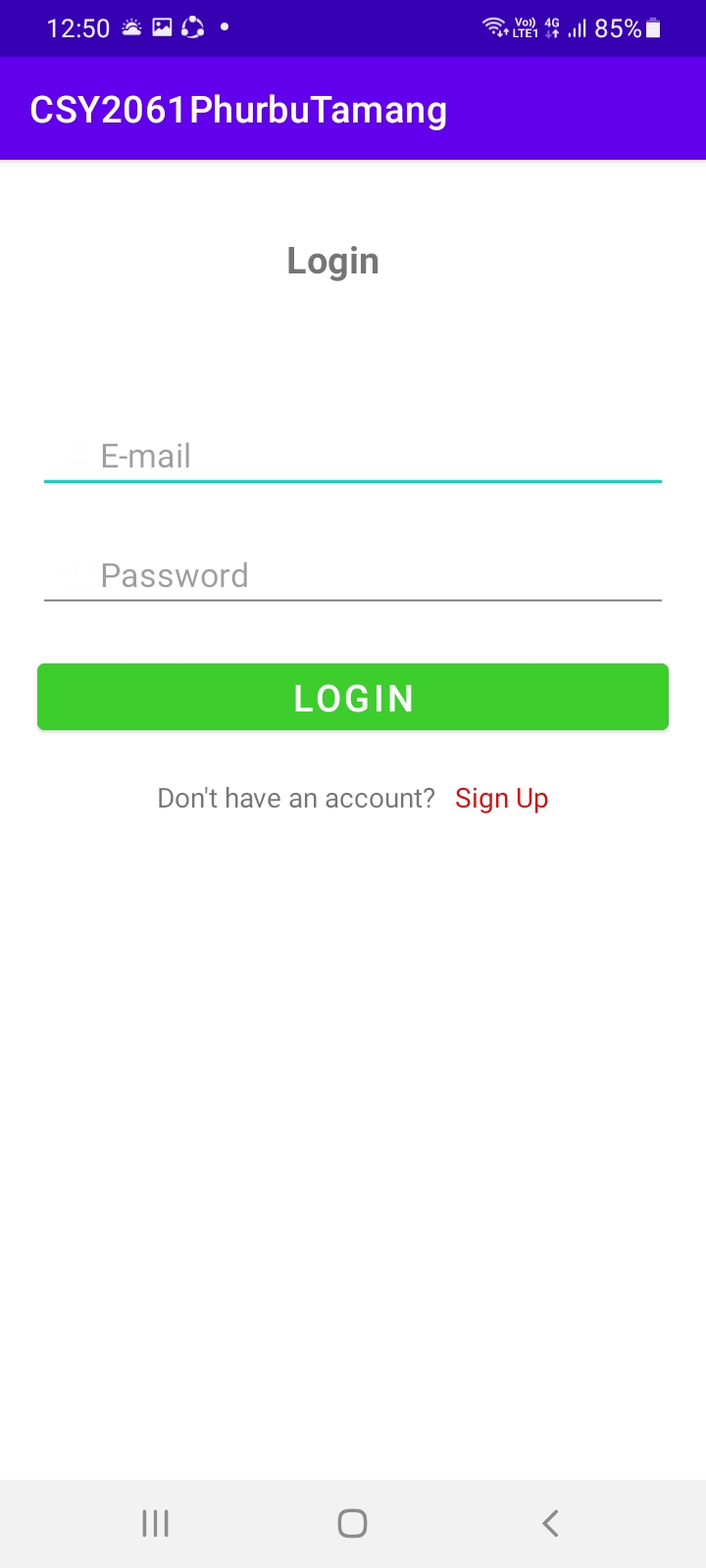


Fig : login page

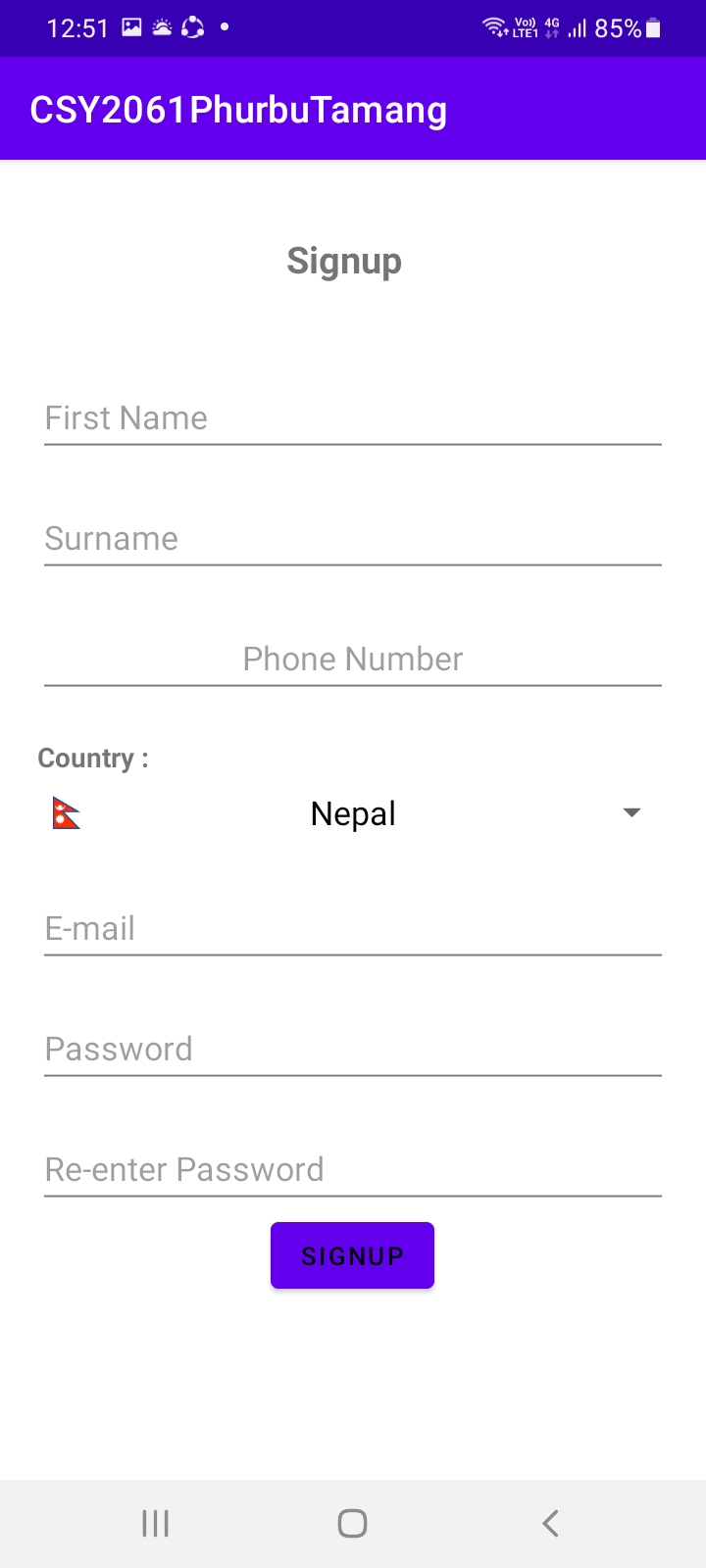


Fig: sign up page

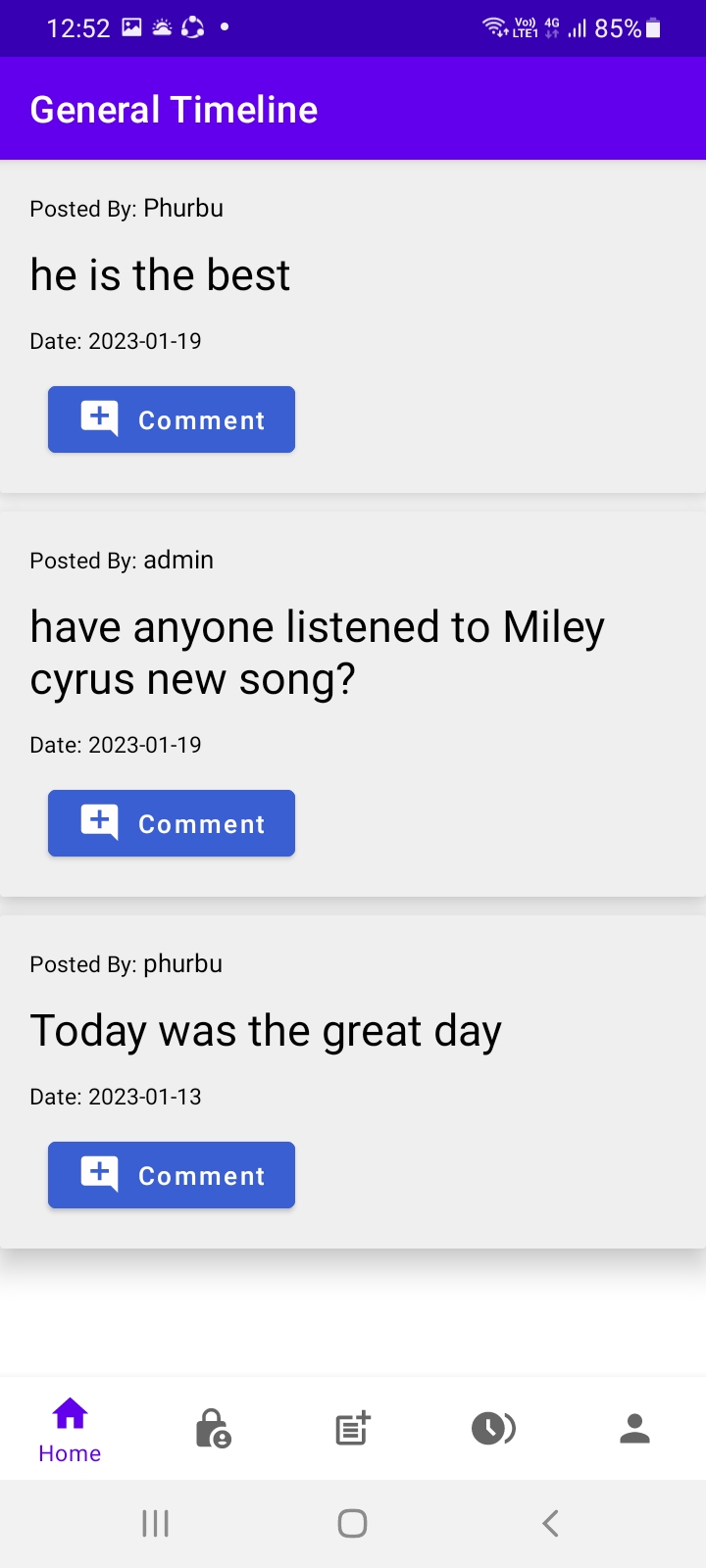


Fig: General timeline user interface

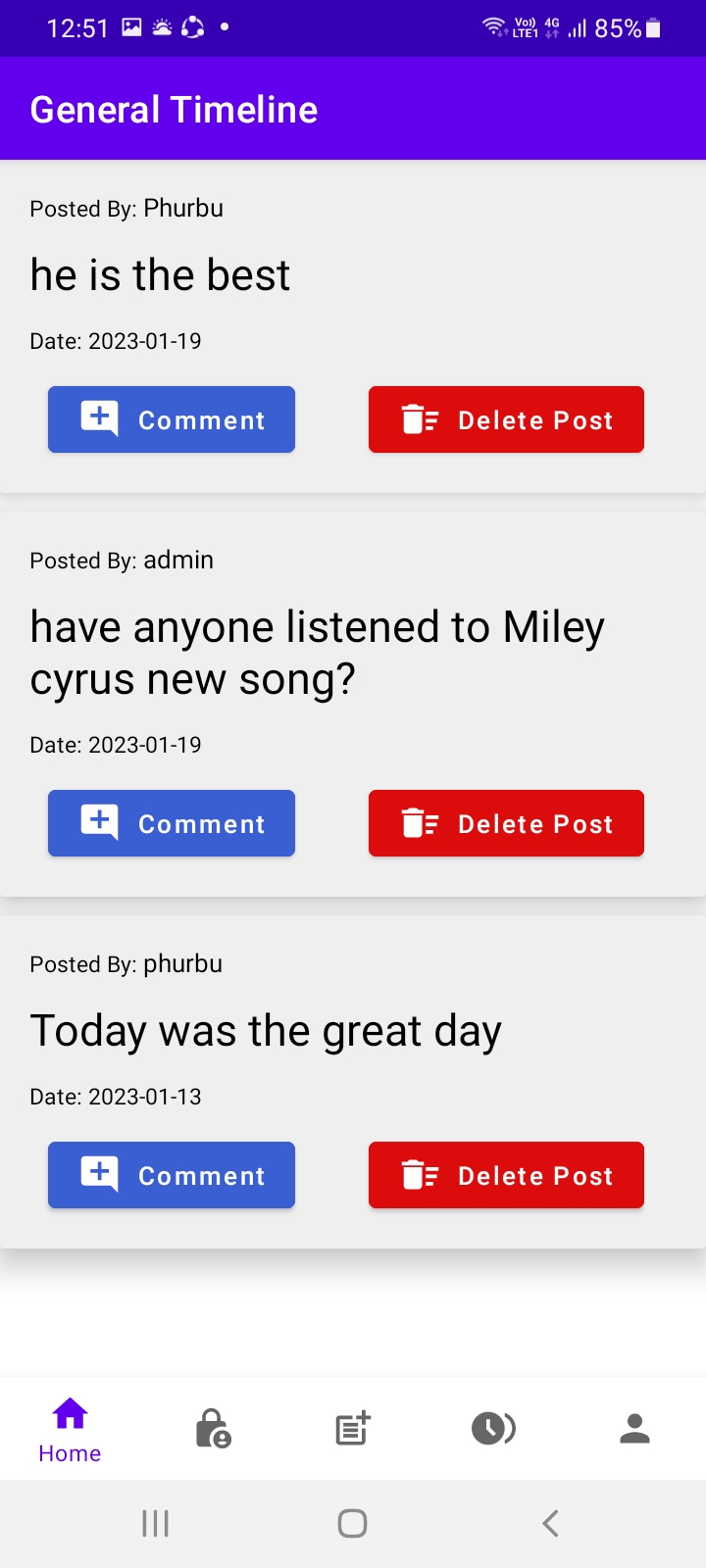


Fig : General timeline admin interface

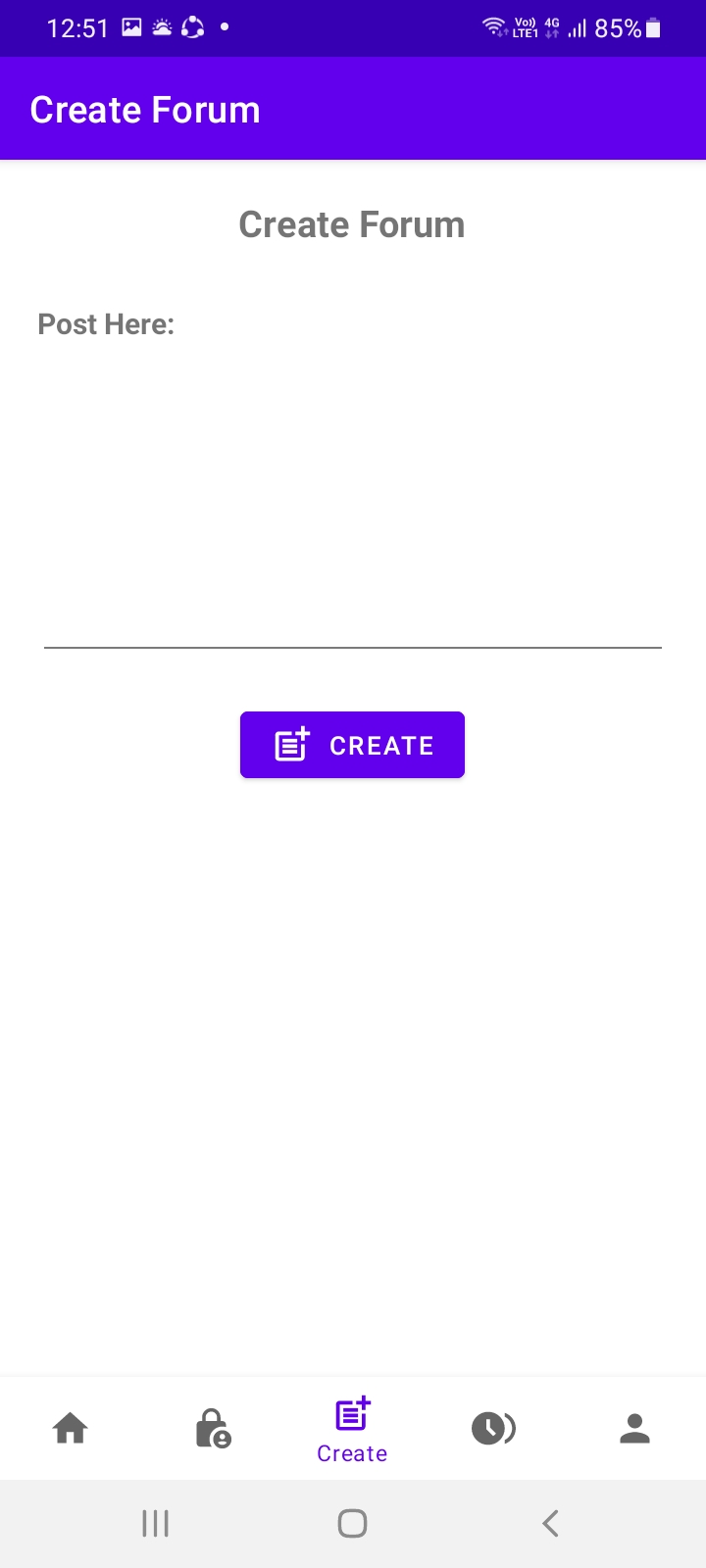


Fig : create forum page

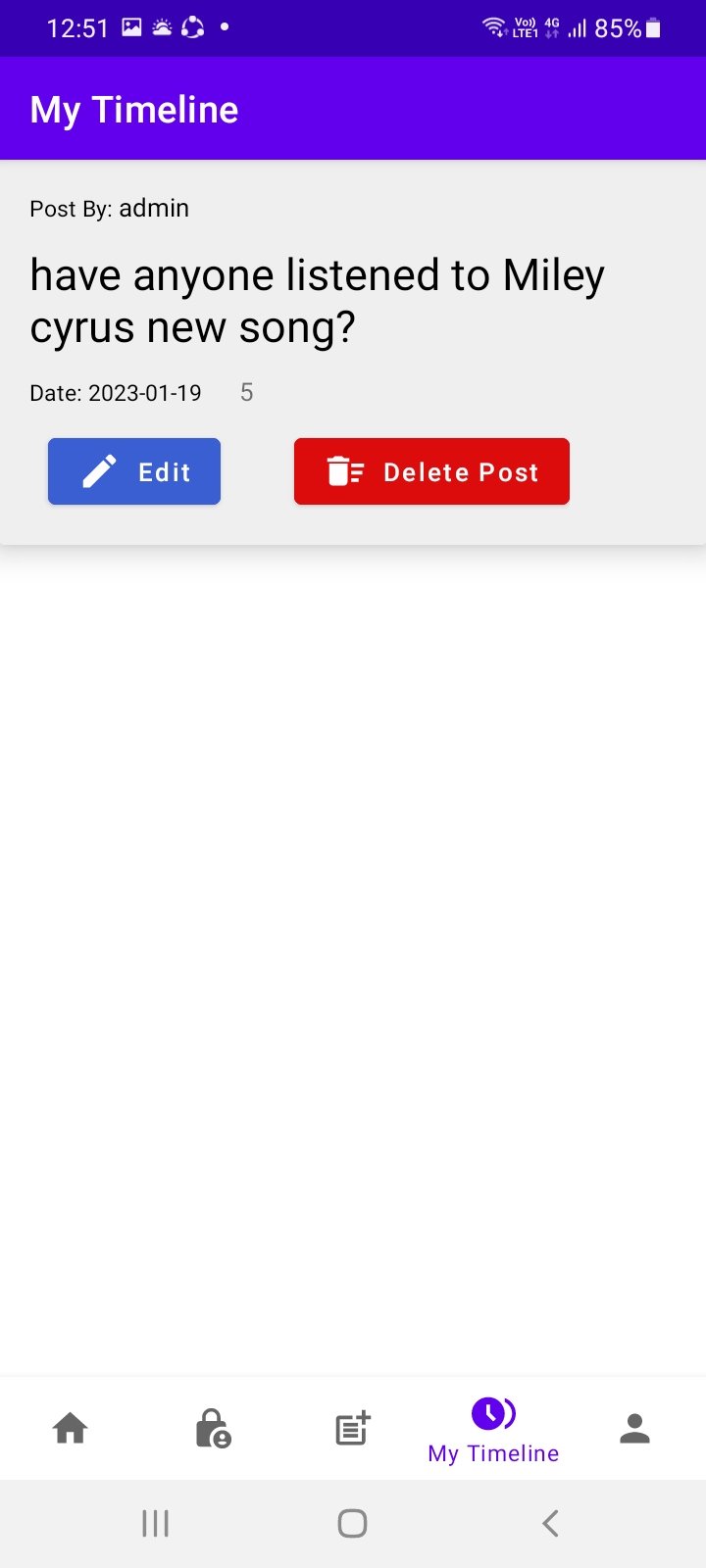


Fig : My timeline page

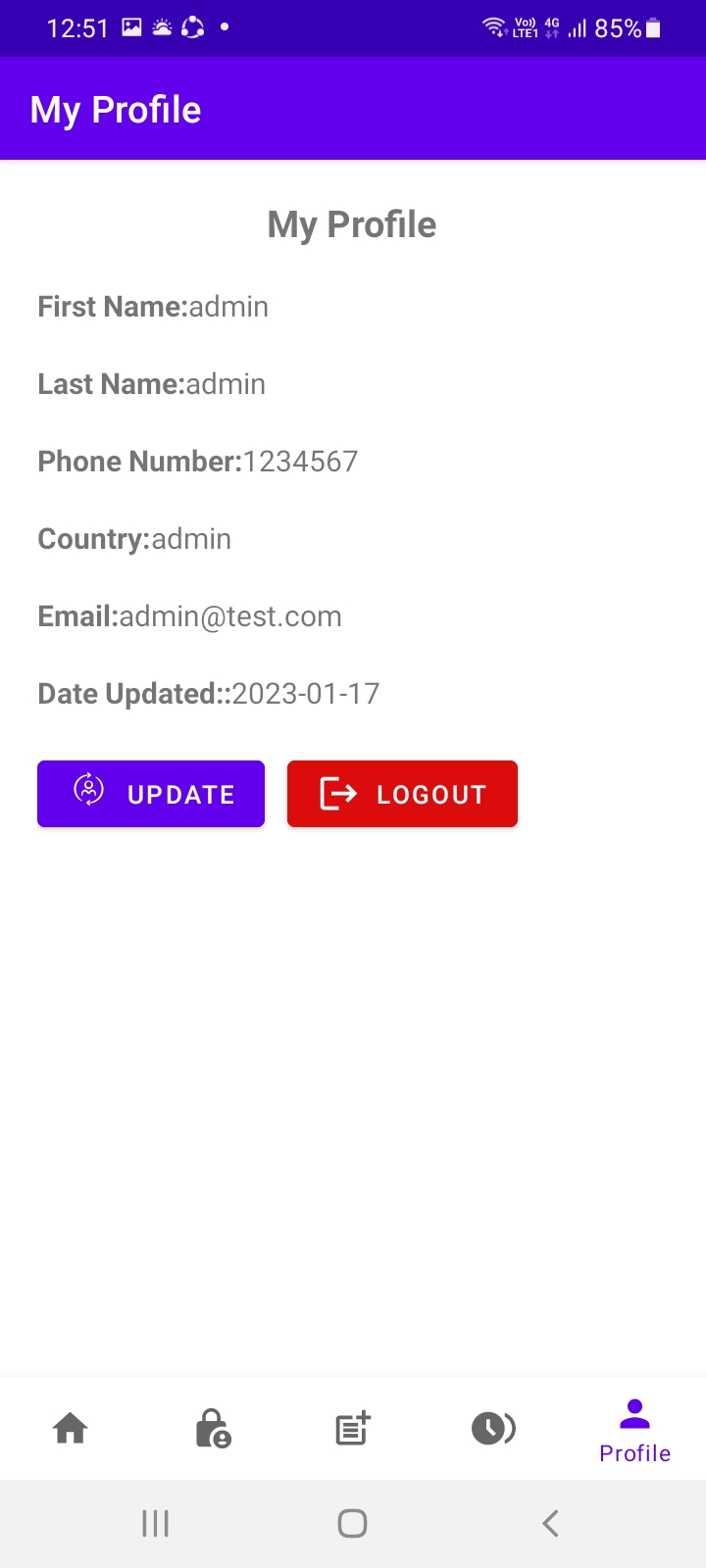


Fig : My profile page

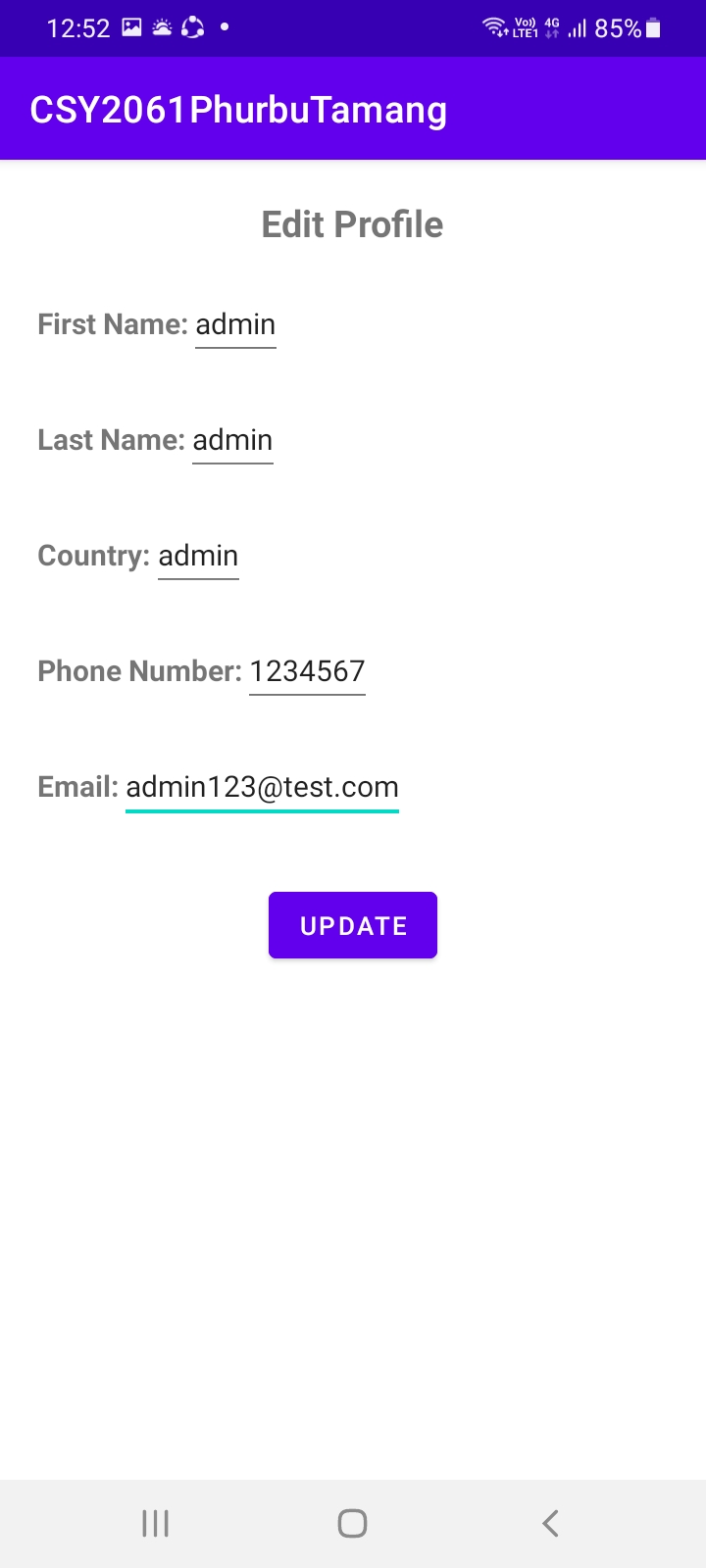


Fig : update page

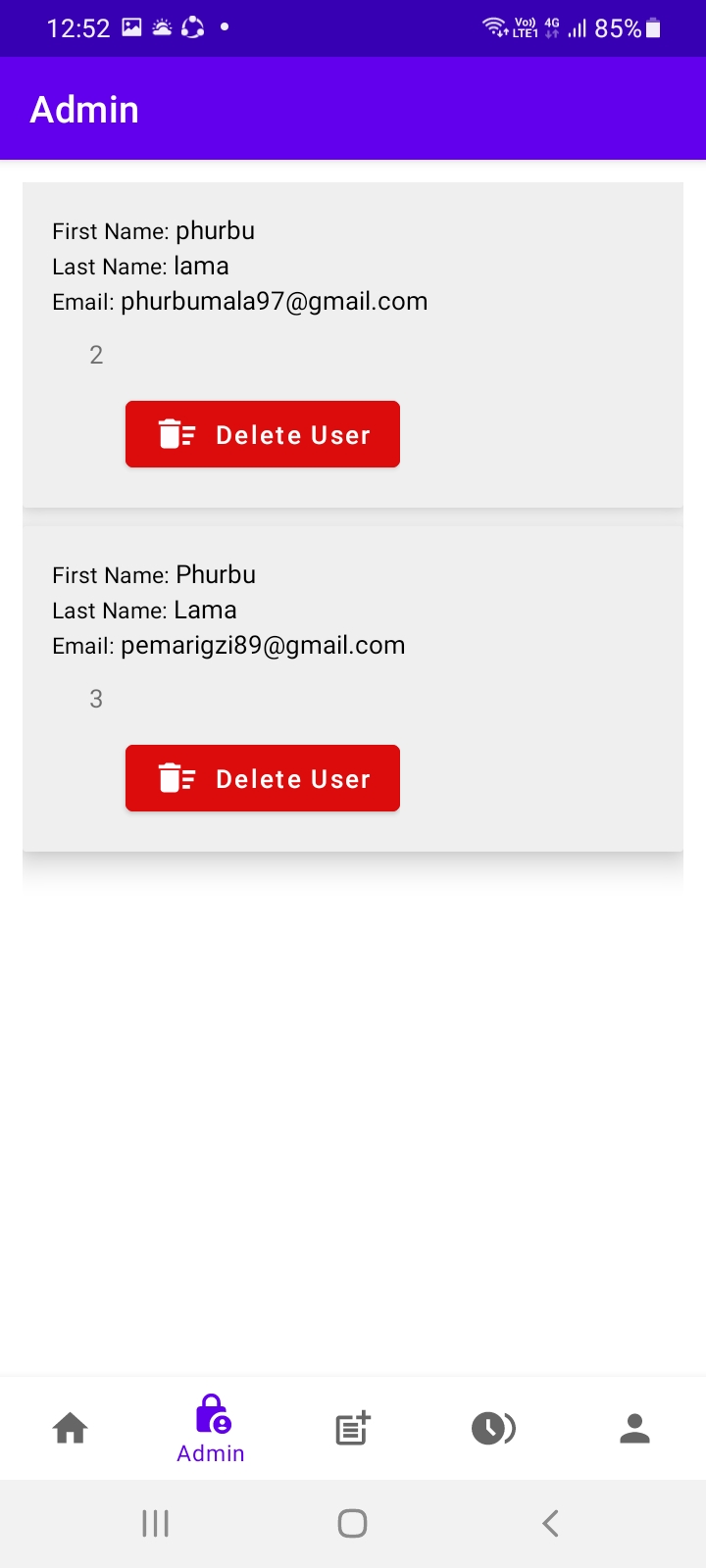


Fig: admin option page

1. Testing
2. **Test 1**

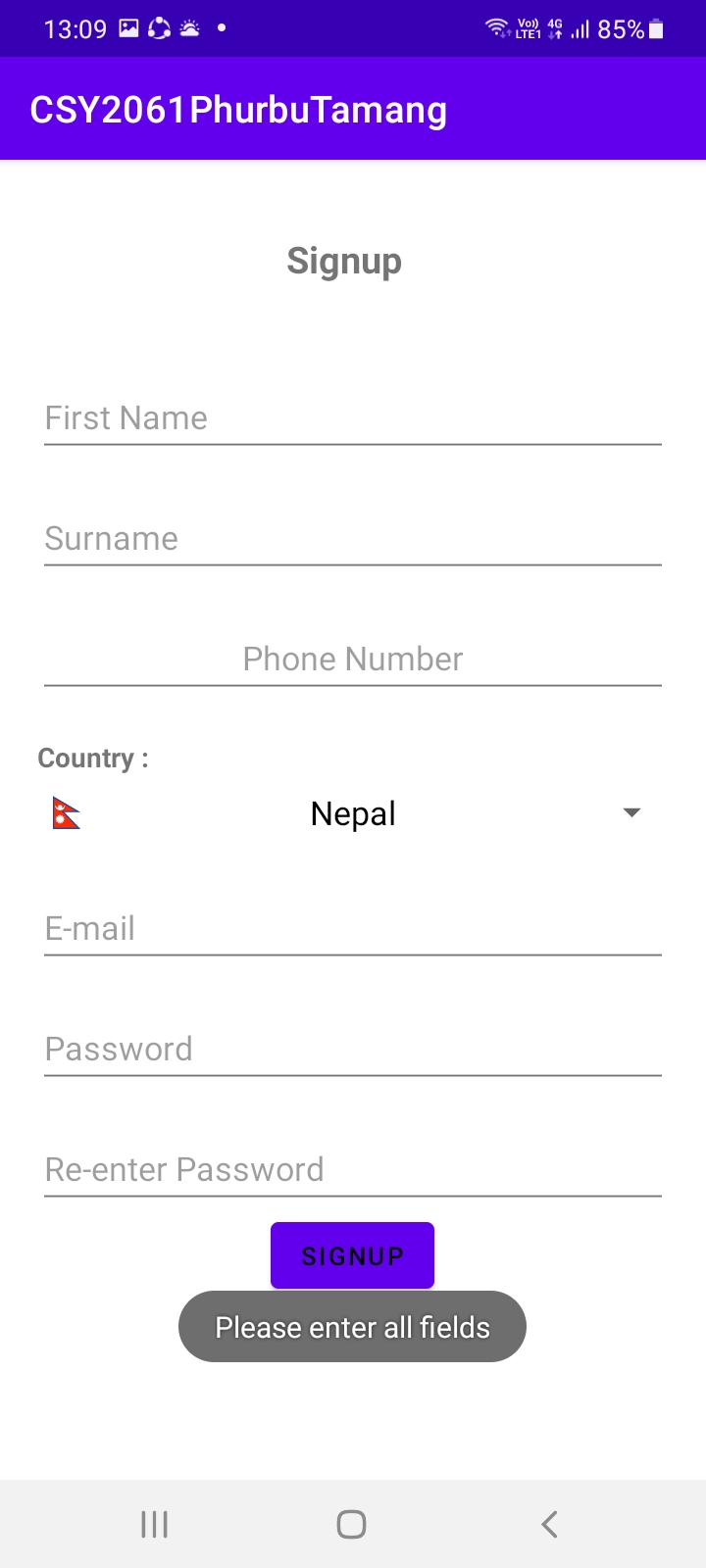
**Test case:** Running if signup activity register a user even if the fields are empty

**Test Procedure:** Required fields are left empty

**Expected Result:** “Please Enter All Fields” to be toasted

**Actual Result:** Same as expected result

**Status:** passed



1. **Test 2**

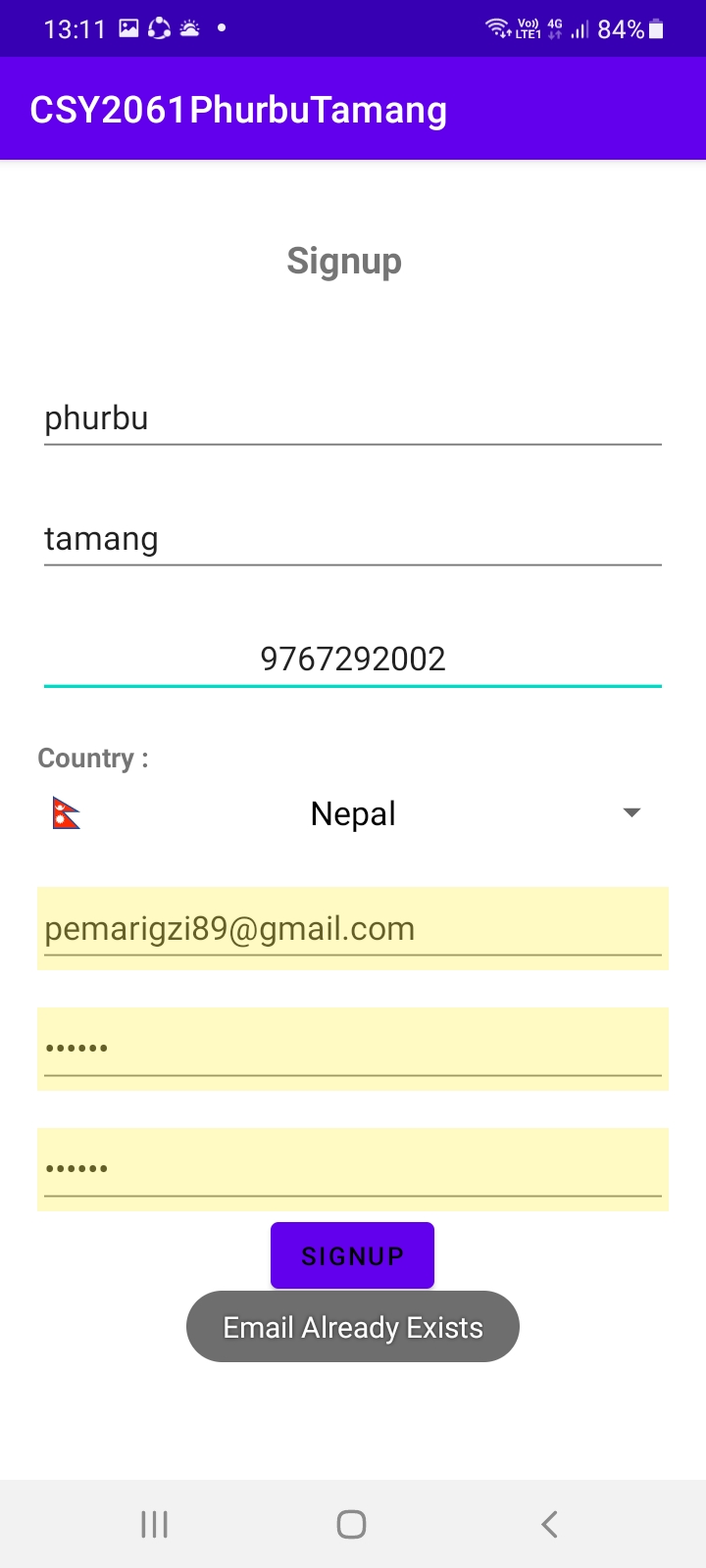
**Test Case:** Registering an account with existing email

**Test Procedure:** Inputting existing email with registering

**Expected Result:** application to toast “Email already exist”

**Actual Result:** same as expected result

**Status:** passed



1. **Test 3**

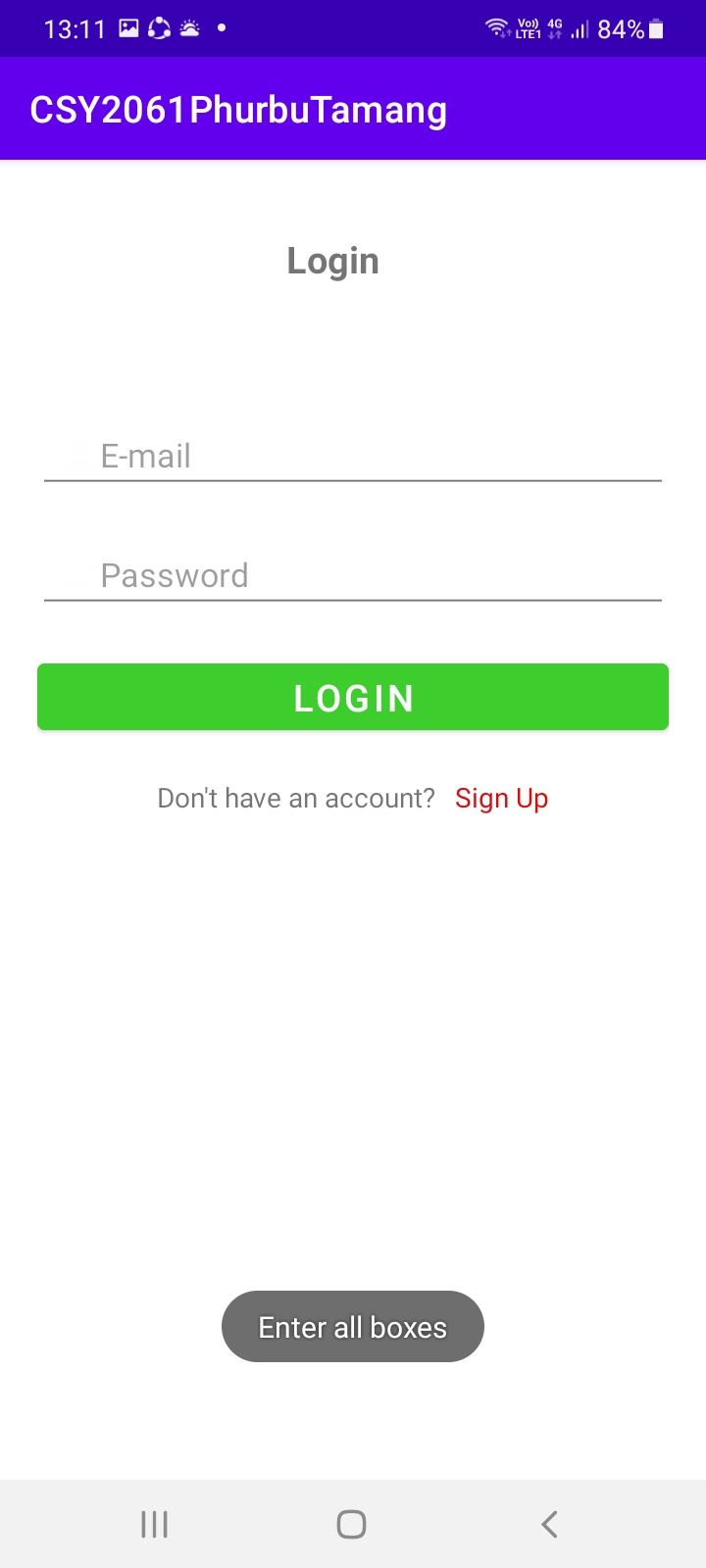
**Case:** Empty username and password in login

**Procedure:** No username and password provided during login

**Expected Result:** application to toast “Enter all boxes”

**Actual Result:** same as expected result

**Status:** pass



1. **Test 4**

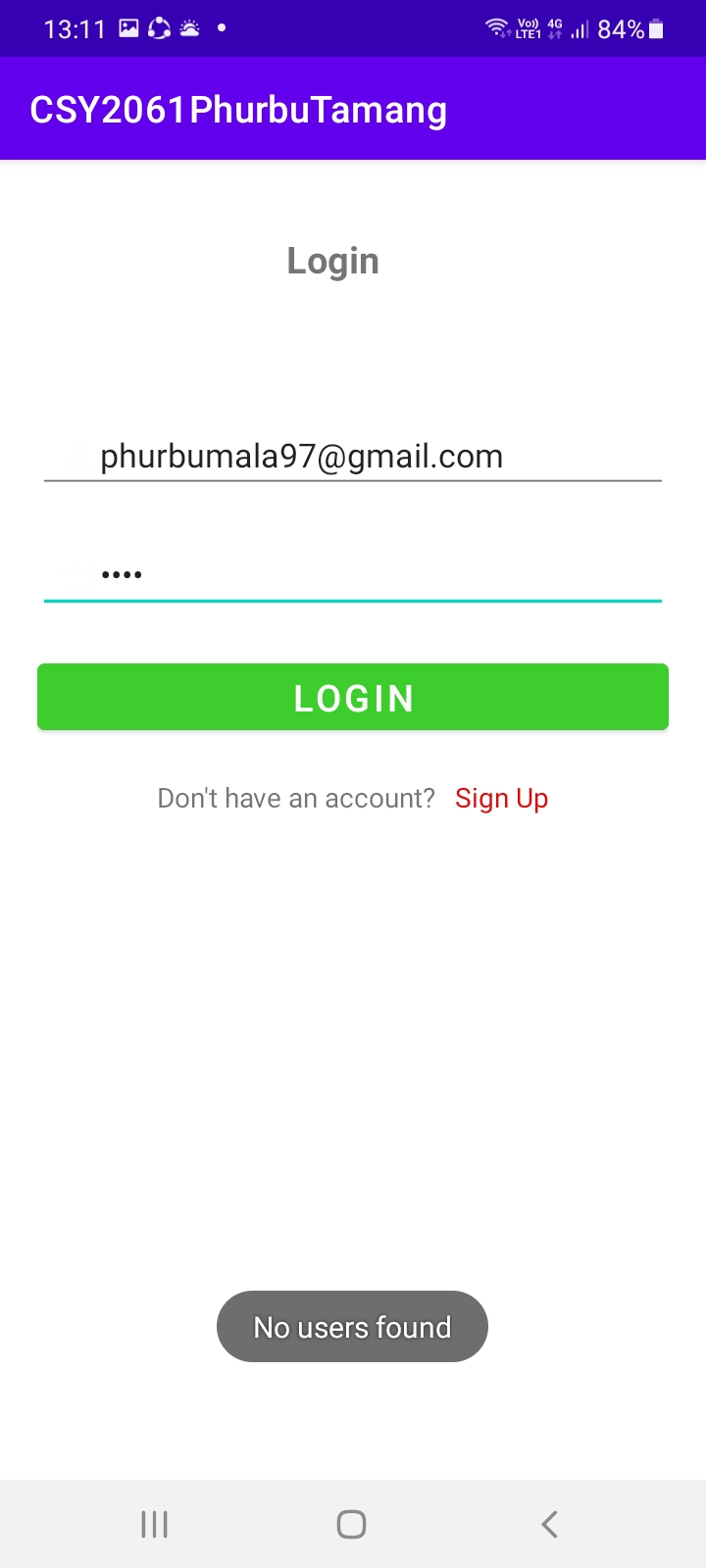
**Case:** wrong username and password in login

**Procedure:** inputting wrong username and password provided during login

**Expected Result:** application to toast “user not found”

**Actual Result:** same as expected result

**Status:** pass



1. **Test 5**

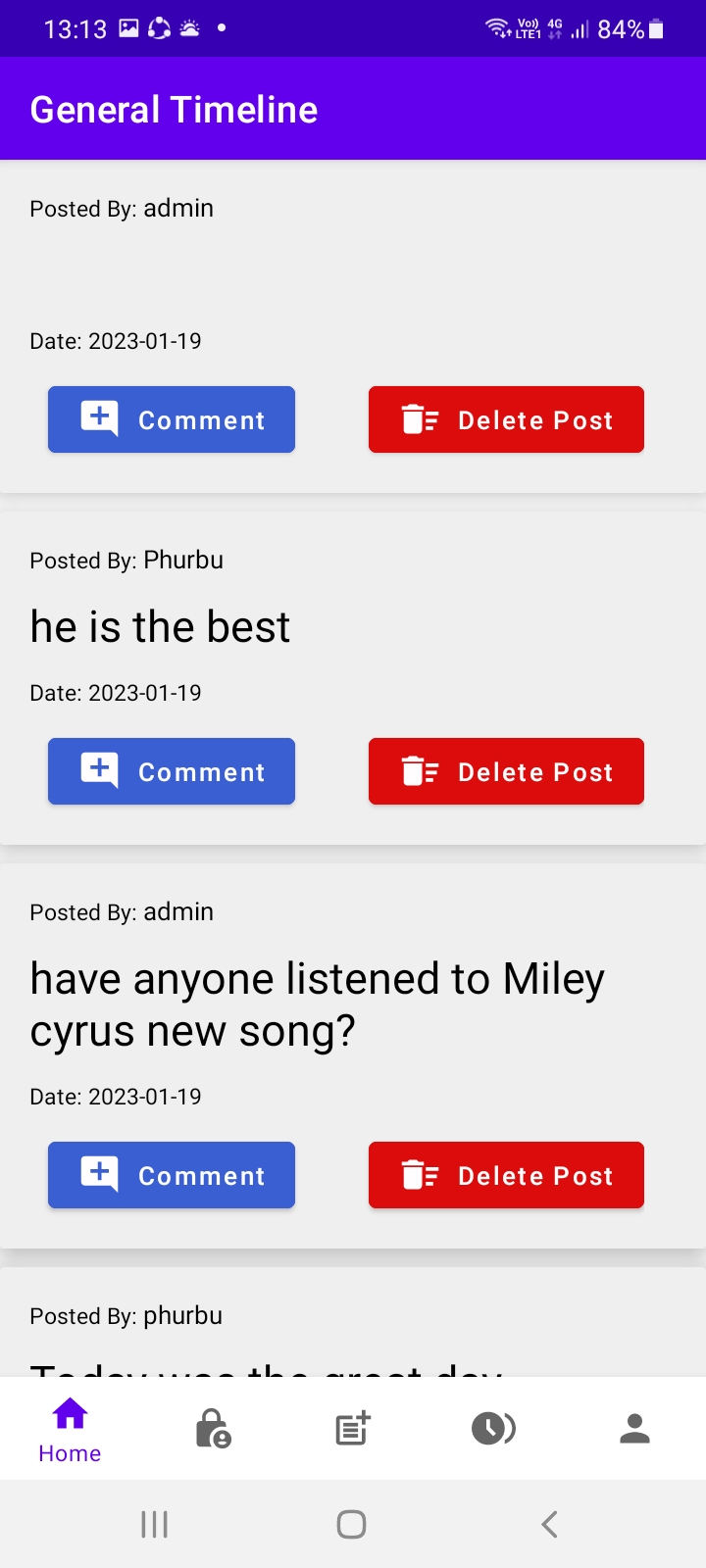
**Case:** empty post

**Procedure:** clicking create with empty post

**Expected Result:** application not to create post

**Actual Result:** post was created

**Status:** fail



List of problems

1. When the content is added , edited or deleted it doesn’t show the changes right after but shows when you visit the page where you expect the changes at next visit.