PLOT TWISTER APP REPORT

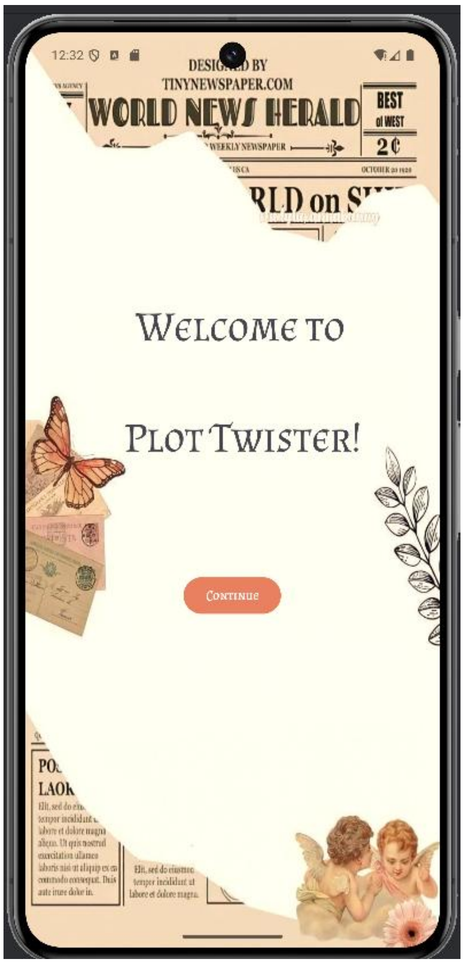
Abstract

Writing can be challenging, especially when faced with writer's block. *Plot Twister* is an Android application designed to help writers overcome creative stagnation by generating random plot ideas based on user-defined parameters. The app also provides a writing space with additional tools like a timer for writing sprints, a date and time picker for reminders, a font size adjustment option, and the ability to save or export stories. Users can sign up, log in, or use a guest mode with limited features. The app integrates SharedPreferences for authentication and story storage, ensuring a seamless experience across devices.

Implementation Details

The application was developed using **Android Studio with Kotlin** and has 6 activities, 4 fragments and 4 menus. It includes the following key components:

Main Activity:

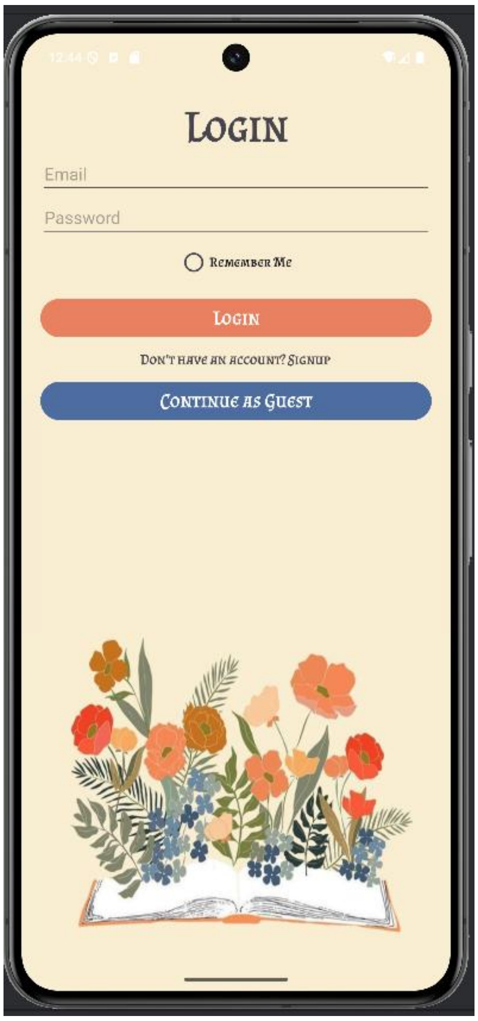


This activity is the opening page of the app.

Concepts Used:

- UI Design: ConstraintLayout, TextView, Button, and ImageView.
- View Handling: findViewById() initializes UI components, and setOnClickListener() handles button clicks.
- Intents & Navigation: Intent(this, LoginActivity::class.java) navigates to the LoginActivity.
- Toast Messages: Toast.makeText() displays the "Log in to continue" notification.
- Edge-to-Edge UI: enableEdgeToEdge() for a full-screen experience.

Login Activity:

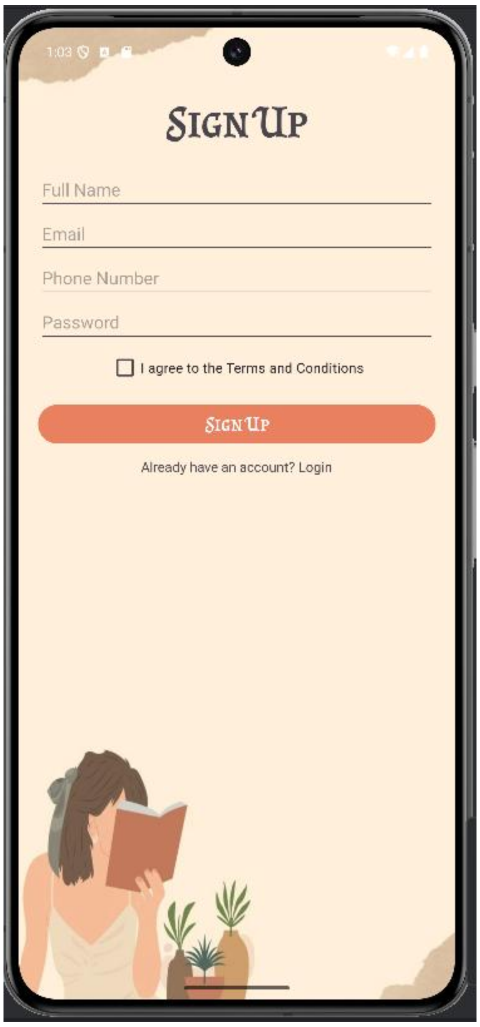


This activity helps the existing users to log in, or take new users to a signup page. It also provides the option of using the app in guest mode.

Concepts Used:

- **UI Design**: LinearLayout, TextView, EditText, Button, Radio Button and ImageView.
- **Saving Data**: SharedPreferences database is used to save the login credentials. If the credentials do not exist, a Toast message “invalid credentials” is displayed.
- **Radio Button**: When the radio button is clicked during login, the database retrieves the credentials and displays them automatically in the required fields for the next time a user logs in.
- **View Handling**: findViewById() initializes UI components, and setOnClickListener() handles button clicks.
- **Intents & Navigation**: When the LOGIN button is clicked, Intent(this, HomeActivity::class.java) navigates to the HomeActivity. When the “DON’T HAVE AN ACCOUNT? SIGNUP” text is clicked, Intent(this, SignupActivity::class.java) navigates to the SignupActivity. When the “CONTINUE AS GUEST” button is clicked, the app goes into guest mode and Intent(this, HomeActivity::class.java) navigates to the HomeActivity, with limited features.
- **Toast Messages**: Toast.makeText() displays “Login Successfull”, if the login is successful and “Please enter email and password” if any of the credentials are missing.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

SignUp Activity:

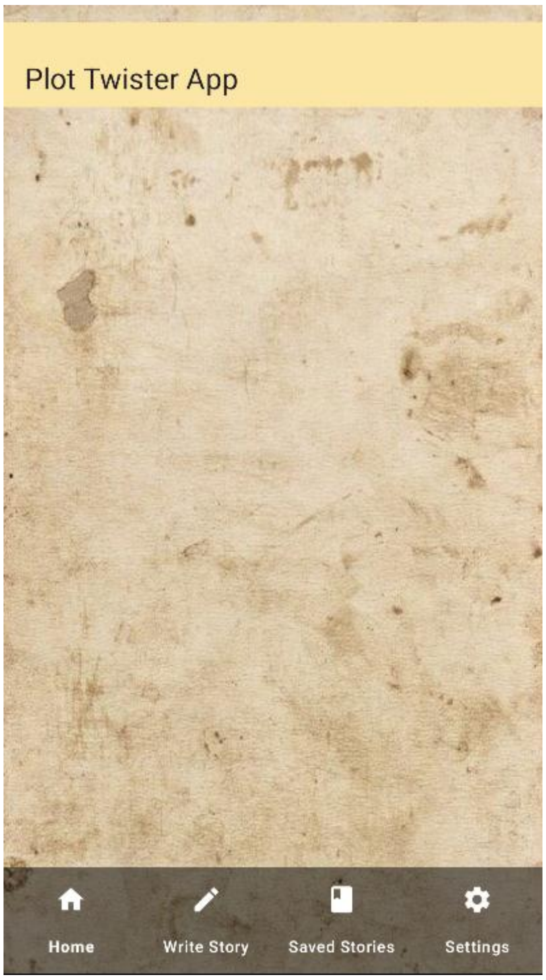

1:03
SIGN UP
Full Name
Email
Phone Number
Password
I agree to the Terms and Conditions
SIGN UP
Already have an account? Login

This activity helps new users to sign up using their full name, email phone number and password.

Concepts Used:

- **UI Design**: LinearLayout, TextView, EditText, Button, CheckBox and ImageView.
- **Saving Data**: SharedPreferences database is used to save the user credentials.
- **Check Box**: This "Terms & Conditions" checkbox is compulsory to click. If not clicked, "Please accept Terms and Conditions" toast notification appears.
- **View Handling**: findViewById() initializes UI components, and setOnClickListener() handles button clicks.
- **Intents & Navigation**: When the SIGNUP button is clicked, Intent(this, LodinActivity::class.java) navigates to the LoginActivity. When the "ALREADY HAVE AN ACCOUNT? LOGIN" text is clicked, Intent(this, LoginActivity::class.java) navigates to the LoginActivity.
- **Toast Messages**: Toast.makeText() displays "Signup Successful", if the Signup is successful and "Please fill all fields" if any of the credentials are missing.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

Home Activity:

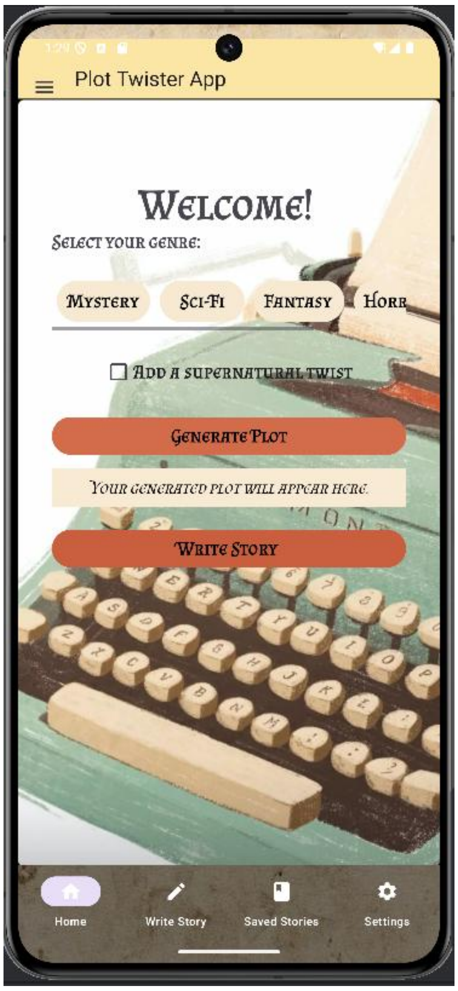


This is the Home Page of the app. This contains all the fragments, menus and ToolBar. Bottom and Side Drawer Navigations also come under this.

Concepts Used:

- **UI Design**: Constraintlayout, Dawerlayout, LinearLayout, Toolbar, FrameLayout, BottomNavigationView, NavigationView and ImageView.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

Home Fragment:

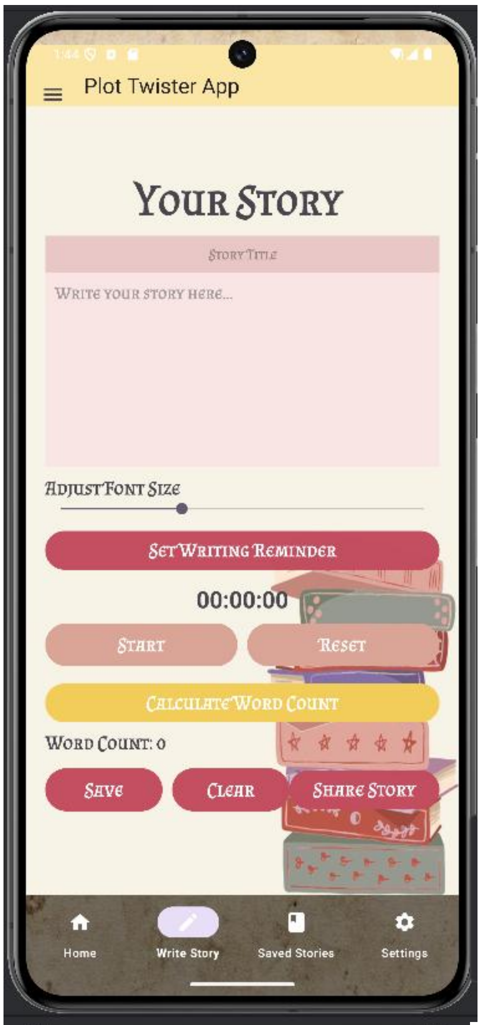


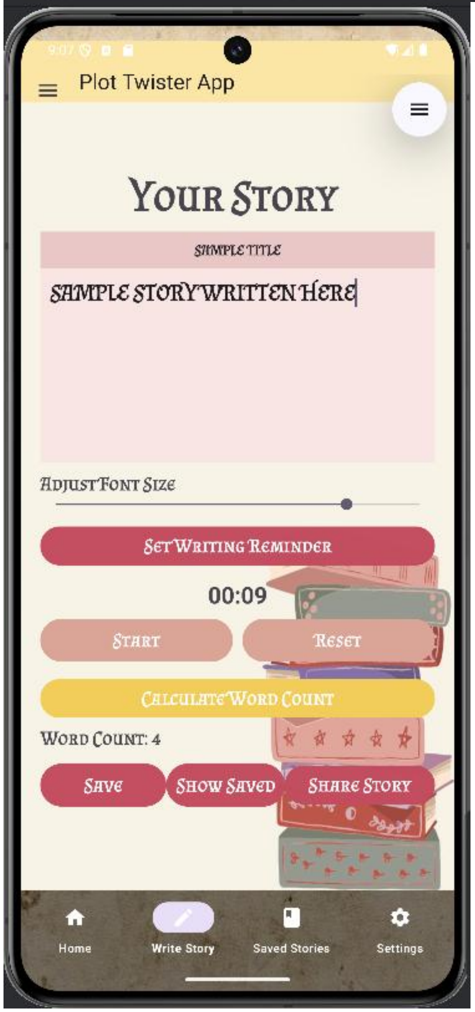
This is the Home Fragment of the app. This is where users can select different filters to generate a random story plot, and write a story based on the plot.

Concepts Used:

- **UI Design**: ScrollView, HorizontalScrollView, LinearLayout, TextView, EditText, Button, CheckBox and ImageView.
- **Genre Buttons**: Horizontal Scroll View is used to display all the buttons (Mystery, Sci-Fi, Fantasy, Horror and Romance).
- **Plot Generation**: Users can select any one genre using the buttons, and check the "Add Supernatural Element" checkbox to generate a story plot. The plot is generated though a random selection of any one of the predefined plot lines embedded in the code, for that particular genre.
- **Check Box**: This "Add Supernatural Element" checkbox adds the line "but there is a supernatural element involved" to the end of the generated plot line, to make the plot more mysterious and promote creative thinking.
- **View Handling**: view.findViewById() initializes UI components, and setOnClickListener() handles button clicks.
- **Navigation**: When the GENERATE PLOT button is clicked, the plot gets generated. When the "WRITE STORY" button is clicked, Intent(requireContext(), WriteStoryFragment::class.java) navigates to the WriteStoryFragment.
- **Toast Messages**: Toast.makeText() displays "Plot Generated", when the plot is generated and "Please select a genre!" if no genre button is clicked.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

Write Story Fragment:



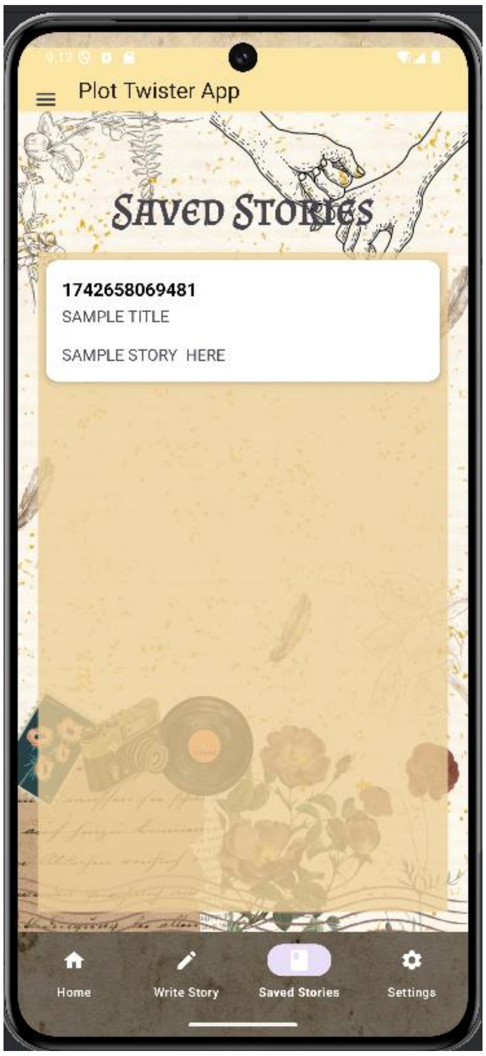


This is the WriteActivityFragment of the app. This is where users can write their stories. They can either do it based on the plot generated in the previous page, or write a story based on a plot of their wish.

Concepts Used:

- **UI Design**: LinearLayout, TextView, EditText, Buttons, Seekbar, Stopwatch, Calculator, Date/Time Picker and ImageView.
- **Buttons**:
  - The SET WRITING REMINDER button opens the Date/Time Picker and is useful for setting reminders to write.
  - The START AND RESET button are used for the timer.
  - The CALCULATE WORD COUNT button calculated the amount of words written in the writing area, and displays it in the textview below.
  - The SAVE button saves the story into the database. The stories can then be viewed in the "Saved Stories" page. Once this button is clicked, the CLEAR button is replaced by SHOW SAVED button, which navigates to the "Saved Stories" page.
  - The CLEAR button clears the editText area and resets the word count to 0.
  - The SHARE STORY button gives a PopUp Menu.
- **Seekbar**: Users can select the font size based on this seekbar.
- **View Handling**: view.findViewById() initializes UI components, and setOnClickListener() handles button clicks.
- **Toast Messages**: Toast.makeText() displays "Story Saved", when the written story is saved, "Title and Story cannot be empty!" if no no title or story is written, and "Reminder set for (date selected) on (time selected)" when the reminder button is clicked and the date/time is selected by the user.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

Saved Story Fragment:



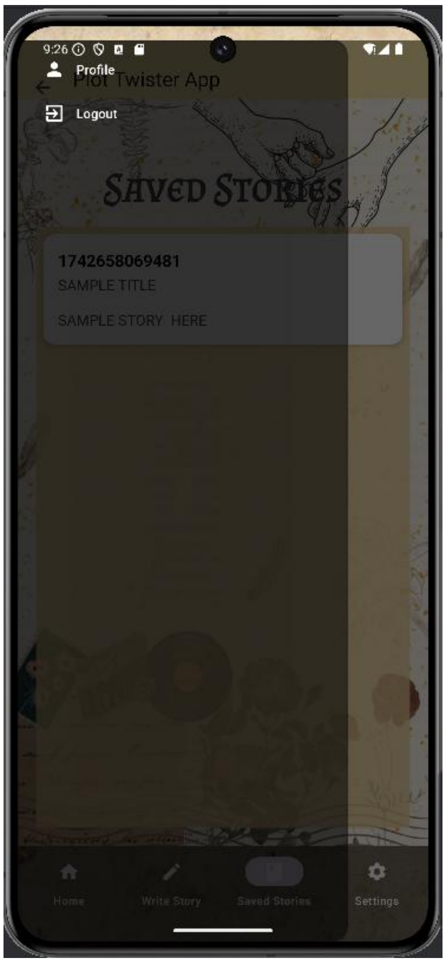
1742658069481
SAMPLE TITLE
SAMPLE STORY HERE

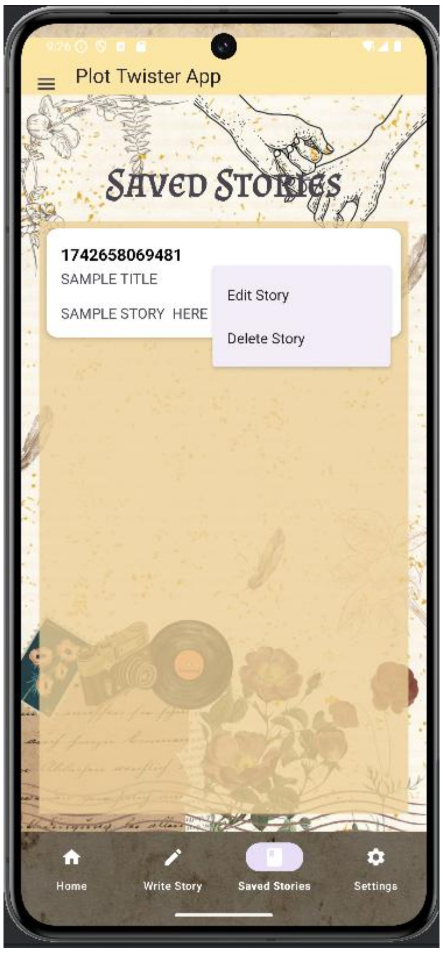
In this fragment, the user can view their saved stories, and edit/delete them using the context menu for each story.

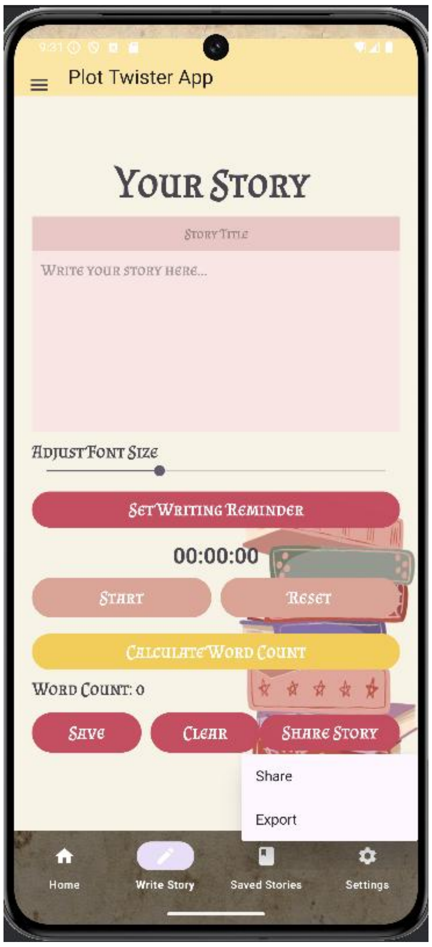
Concepts Used:

- **UI Design**: LinearLayout, TextView, NestedScrollView, RecyclerView and ImageView.
- **RecyclerView**: All the story cards are displayed using the RecyclerView card view.
- **Story Saving**: The stories are saved using a unique numeric key value, and then is fetched from the database. The stories can be viewed ONLY by the user that is logged in, no one else (as it is saved according to the user id in the database).
- **View Handling**: view.findViewById() initializes UI components.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

Bottom Navigation, Side Navigation, PopUp Menu and Context Menu:







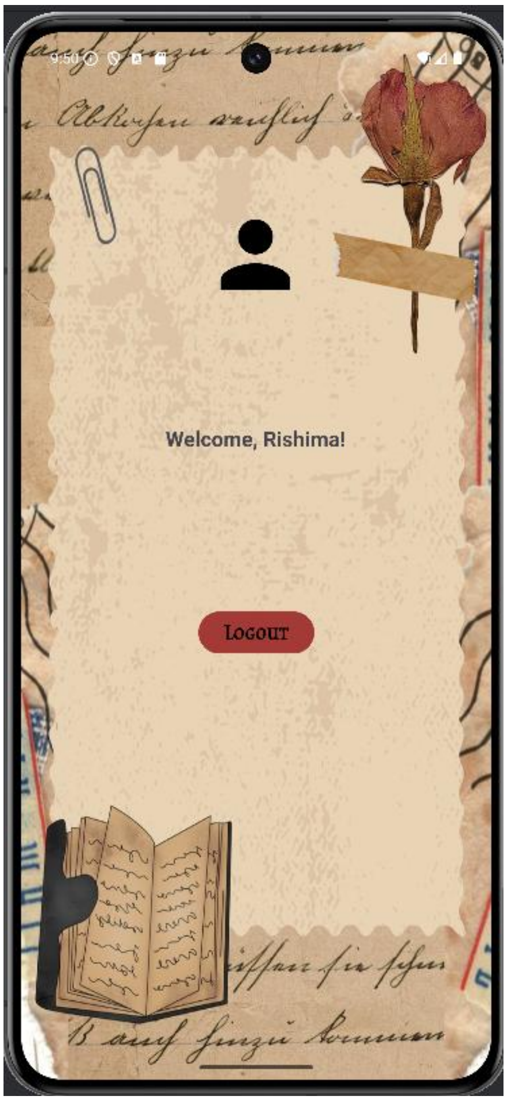
The Side Drawer Navigation consists of a Profile option, which navigates to the Profile Activity, and a Logout option, which logs the user out of the app, and navigates back to the Login Activity.

The Context Menu in the Saved Stories Fragment appears when a story in the recycle view is long pressed. It has an Edit Story option, which navigates to the Edit Story Activity, and a Delete Story option, which deletes a story from the database.

The PopUp Menu appears when the SHARE STORY button in the Write Story Fragment is clicked. It has a Share option, which uses Implicit Intent to share the story using the web. If the story title or story area is empty, the share option gives a toast message “Nothing to share!”.

All these menus use icons which Vector Asset files. They are made and saved in the res/drawable folder.

Profile Activity:

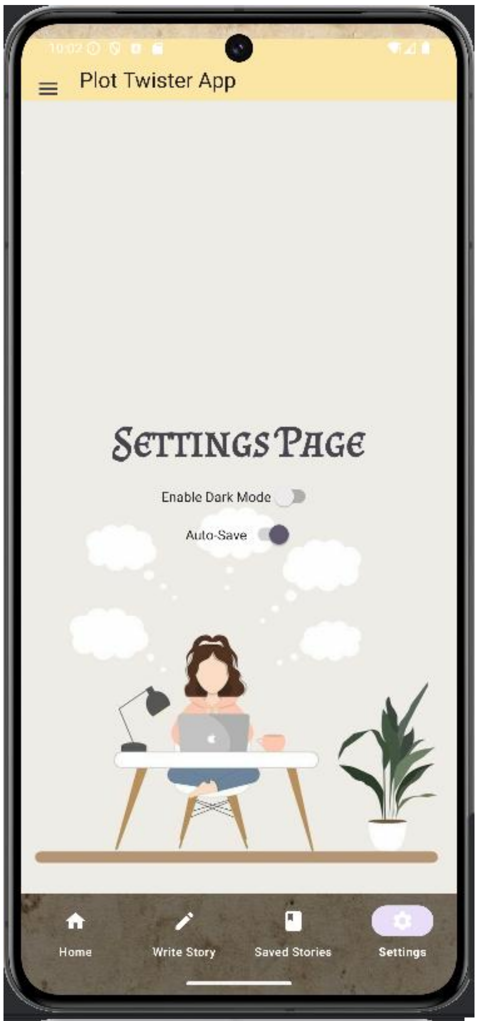


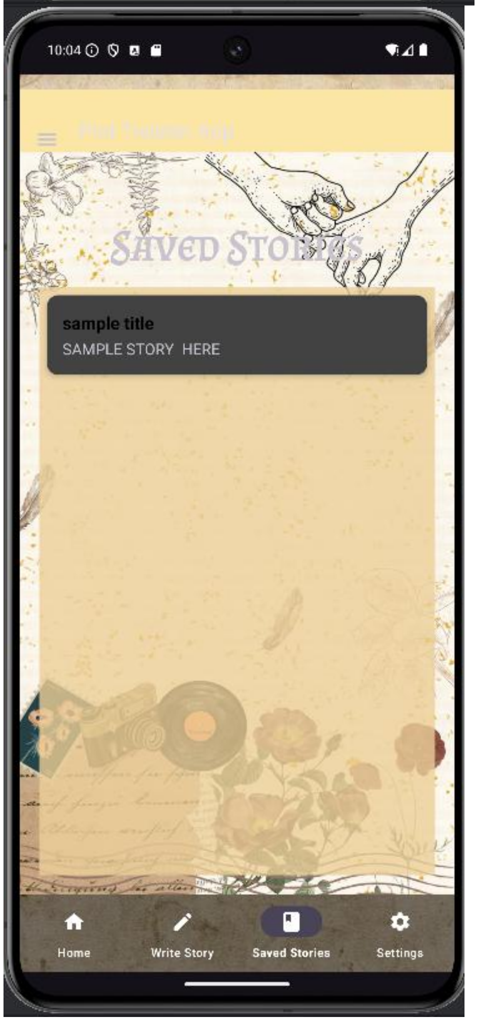
In this activity, users can view their profile, and logout if they want to.

Concepts Used:

- **UI Design**: Constraintlayout, TextView, Button and ImageView (icon).
- **TextView**: The name from the user entered credentials is picked it from the database and is displayed in the TextView.
- **View Handling**: view.findViewById() initializes UI components, and setOnClickListener() handles button clicks.
- **Navigation**: When the LOGOUT button is clicked, Intent(this, LodinActivity::class.java) navigates to the LoginActivity.
- **Toast Messages**: Toast.makeText() displays “Logged Out Successfully!”, when the user logs out.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

Settings Fragment:



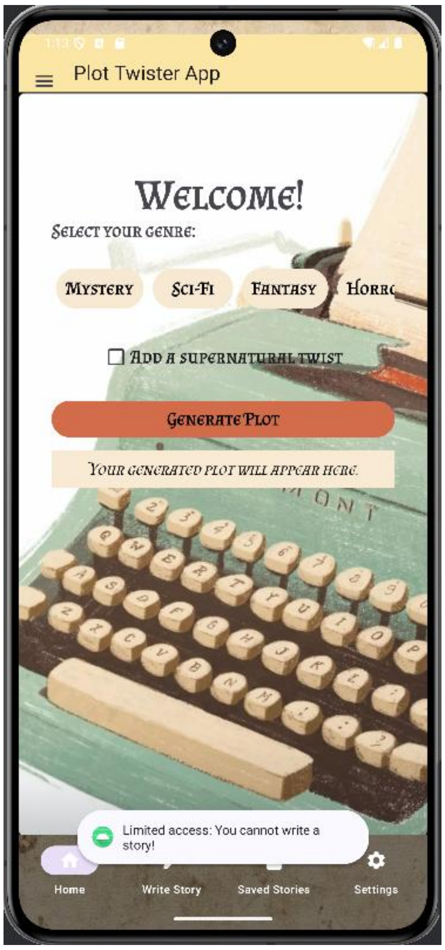


In this fragment, users can use switches to select the dark/light mode, and to turn on/off Auto Save.

Concepts Used:

- **UI Design**: LinearLayout, Switch, and ImageView (icon).
- **DarkMode Switch**: When this swtch is enabled, the app enters a dark mode (as is shown in the second screenshot) and navigated back to the Home Fragment. This Preference is then saved in the database, and is continued the nect time the same user logs into the app.
- **View Handling**: view.findViewById() initializes UI components.
- **Edge-to-Edge UI**: enableEdgeToEdge() for a full-screen experience.

Guest Mode:



Limited access: You cannot write a story!

This mode allows guest users to use different genres and generate different plots, but prevents them from writing a story and saving it. It also prevents the guests from accessing the "Saved Stories" page. However, they can navigate to the "Settings" page, and come back to the "Home" page. They also have access to the Side navigation drawer menu.

Purpose and Practical Relevance

This app aims to:

- Inspire writers by generating creative prompts.
- Provide a distraction-free writing environment with helpful tools.
- Encourage structured writing through time-bound challenges.
- Offer storage so writers can access their work anytime.

The **practical impact** of the application extends to students, professional writers, and hobbyists, making it a valuable tool for creative development.

Conclusion

Plot Twister successfully combines plot generation with essential writing tools, offering a seamless experience for both amateur and experienced writers. By integrating SharedPreferences for user authentication and story storage, the app ensures accessibility and data security.