FINAL PROJECT REPORT

SEMESTER 1, ACADEMIC YEAR: 2025-2026

CT312H: MOBILE PROGRAMMING

- **Project/Application name:** ComicsVerse: A Comics Reading Application
- **GitHub link:** https://github.com/25-26Sem1-Courses/ct312hm01-project-Giang0402.git
- **Youtube link:** https://youtu.be/VMs2ck3VO_s?si=gUpXZYO9iv6fIiWb
- **Student ID 1:** B2205976
- **Student Name 1:** Trần Hữu Giang
- **Student ID 2:** B2205983
- **Student Name 2:** Đỗ Đạt Hoa
- **Class/Group Number :** CT312HM01

I. Introduction

- ComicsVerse is a cross-platform comic book reader. The app provides a modern, smooth reading experience with key features such as discovery, search, reading, bookmark management, and real-time commenting.

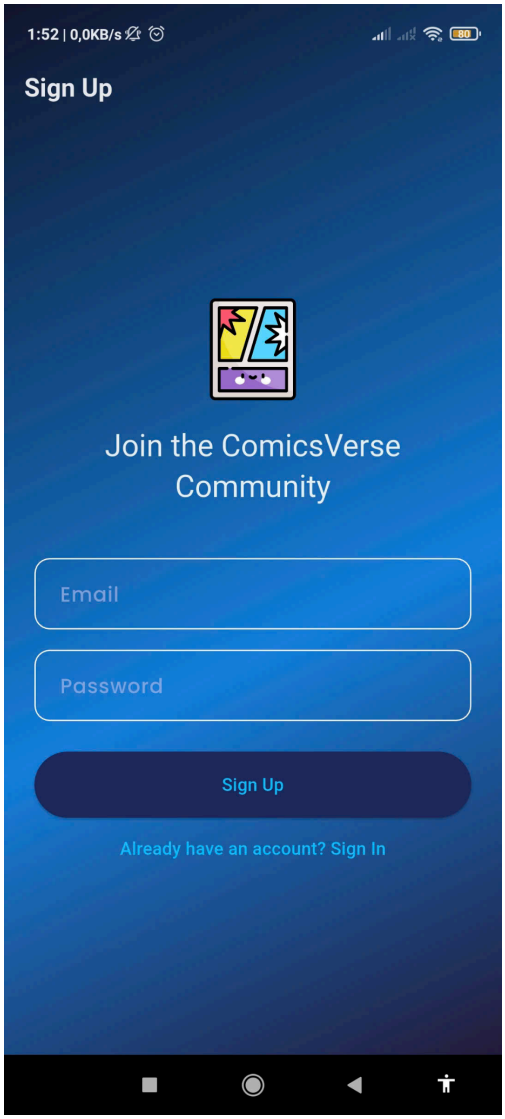
- **Task assignment sheet:**

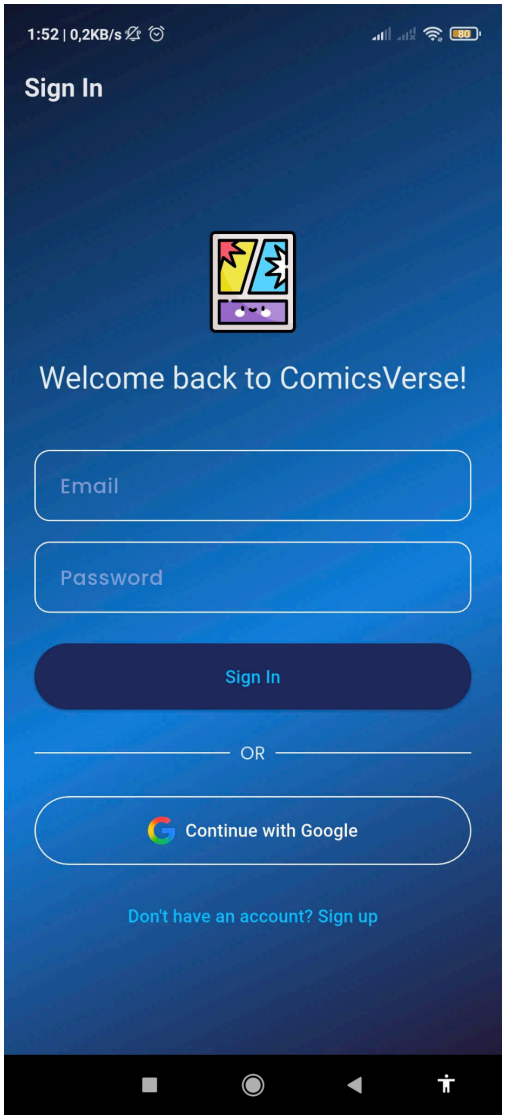
	Tasks	Giang	Hoa
	Project features		
1	**Authentication (Login/Signup)**		**X**
2	**Profile Screen**		**X**
3	**Edit Profile Screen**		**X**
4	**Search and Filter Screen**		**X**
5	**Home Screen**	**X**	
6	**Story Details Screen**	**X**	
7	**Library Screen**	**X**	
8	**Reader Screen**	**X**	
9	**Comments**	**X**	
10	**Ranking Screen**	**X**	**X**

II. Details of implemented features

Feature 1: Authentication (Login / Sign Up)

- **Description:**
  This feature allows users to log into the application using email/password or Google account (OAuth). New users can register for an account using email and password. Supabase Auth manages the authentication process. The screen also includes input fields with validation. Upon successful login, a welcome notification is displayed to the user.
- **Screenshots:**

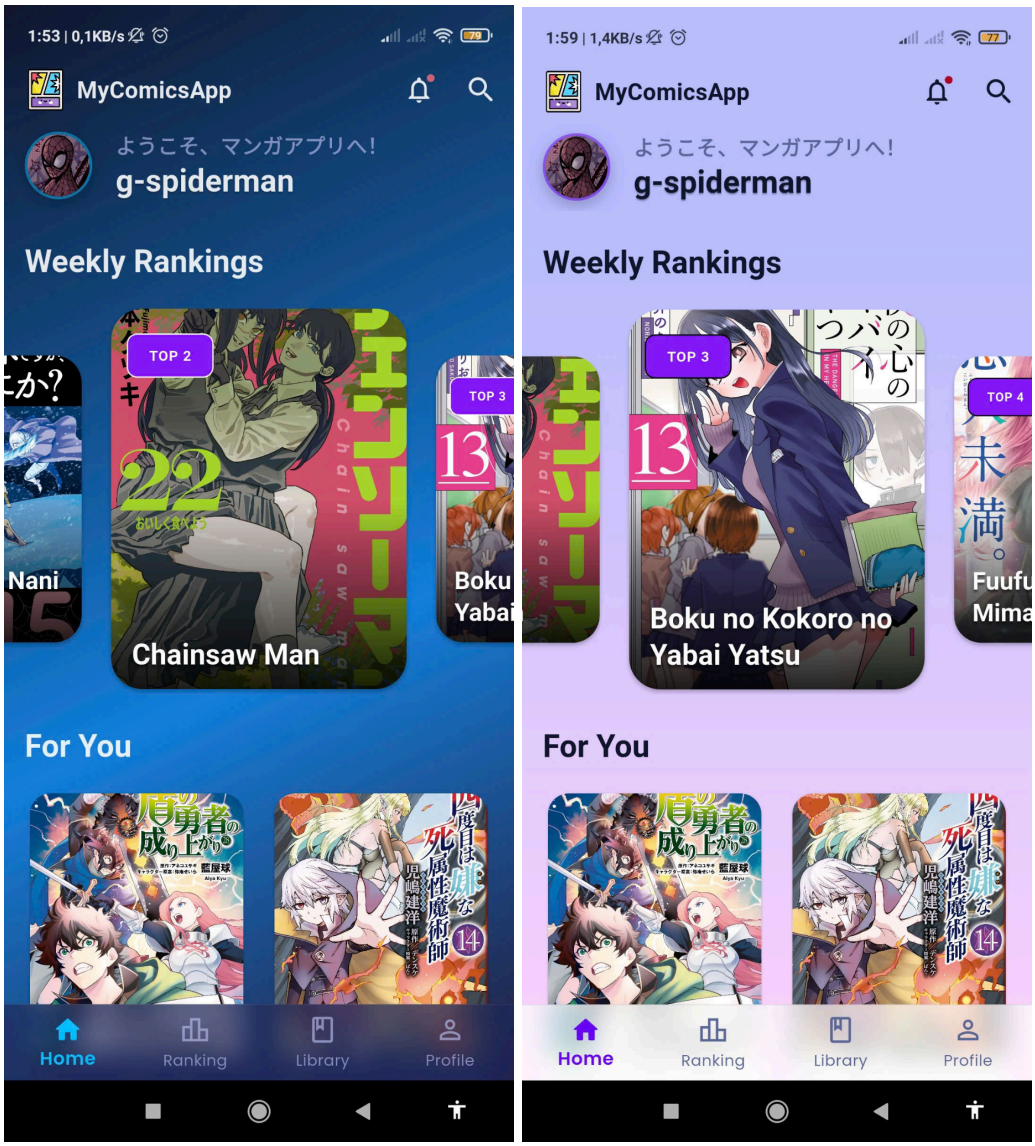




- **Implementation details:**
  - **Widgets:** Scaffold, AppBar, Form, Column, TextFormField (via custom AuthField widget), ElevatedButton, TextButton, OutlinedButton.icon, Icon, Image.asset, CircularProgressIndicator, Flushbar (via custom ToastUtils).
  - **Libraries/Plugins:**
    - supabase_flutter: Interacts with Supabase Auth for login and sign-up operations.
    - flutter_riverpod: Manages state (loading, error) and provides AuthRepository.
    - go_router: Navigates between screens (e.g., from Sign Up to Login).
    - another_flushbar: Displays the success notification toast upon login.
  - **State Management:** Uses Riverpod:
    - authRepositoryProvider: Provides an instance of AuthRepositoryImpl.
    - authStateChangesProvider: A stream that tracks the user's login/logout status.
    - another_flushbar: Displays the success notification toast upon login.
    - Uses ConsumerStatefulWidget and setState to manage the local loading state of buttons.
  - **Data Storage:** User data (email, authentication info) is stored and managed remotely on Supabase Auth. Linked profile information is stored in the profiles table on Supabase Database.
  - **REST API:** Does not directly call REST APIs. Interacts with Supabase Auth via the supabase_flutter library.

Feature 2: Home Screen

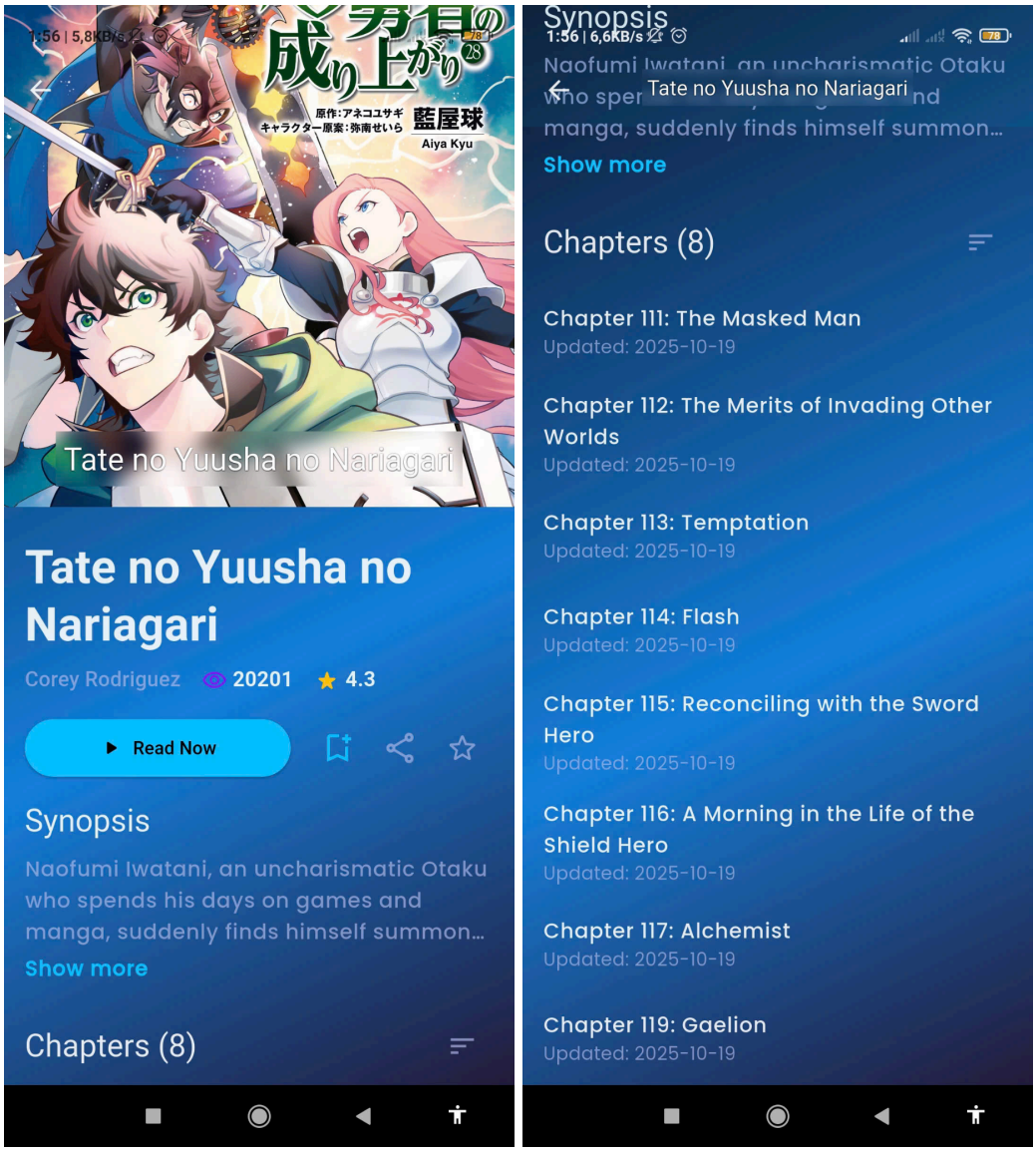
- **Description:**
  The main screen is displayed after a successful user login. Includes a greeting, a section displaying stories by Weekly Rankings (carousel format), and a "For You" recommendation section (grid format). Allows users to navigate to the search screen.
- **Screenshots:**



- **Implementation details:**
  - **Widgets:** Scaffold, CustomScrollView, SliverAppBar (custom HomeSliverAppBar widget), FlexibleSpaceBar, RankingCarouselSection (custom widget), ForYouGridSection (custom widget), StoryCard (custom widget), CachedNetworkImage, Shimmer (for loading effect), ConsumerWidget.
  - **Libraries/Plugins:**
    - **flutter_riverpod:** Provides story data via allStoriesProvider.
    - **cached_network_image:** Displays and caches story cover images.
    - **shimmer:** Displays a loading skeleton effect.
    - **carousel_slider:** Creates the carousel for rankings.
    - **go_router:** Navigates to the search and story details screens.
  - **State Management:** Uses Riverpod:
    - allStoriesProvider: FutureProvider to fetch the list of all stories from StoryRepository.
    - userProfileProvider (from Profile feature): Fetches user information to display the greeting.
    - Widgets use ConsumerWidget or Consumer to listen for changes from providers.
  - **Responsive Design:** Adapts the layout based on screen width using MediaQuery. On larger screens (>600px), the grid displays 4 columns and the carousel adjusts its viewport fraction for a better viewing experience. On smaller screens, it defaults to a 2-column grid optimized for mobile.
  - **Data Storage:** Story data (Story) is fetched remotely via StoryRepository from the Story table on Supabase.
  - **REST API:** None. Interacts with Supabase DB via the client library.

Feature 3: Story Details Screen

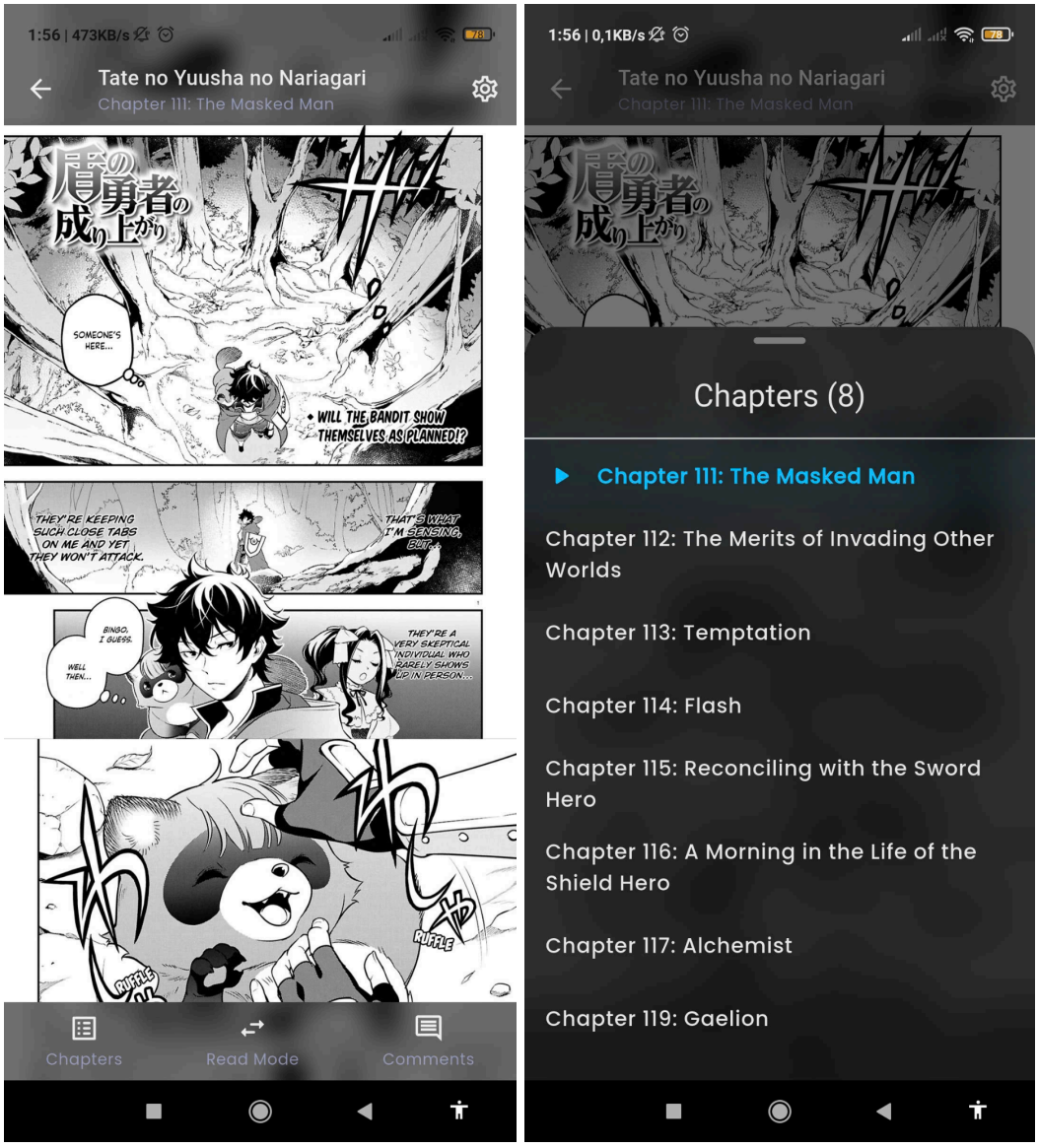
- **Description:**
  Displays detailed information about a selected story, including a large cover image (with parallax effect and Hero animation), title, author, rating, synopsis (expandable/collapsible), action buttons (Read Now, Add/Remove from library, Share, Review), and a list of chapters.
- **Screenshots:**



- **Implementation details:**
  - **Widgets:** Scaffold, CustomScrollView, SliverAppBar, FlexibleSpaceBar, Hero (for cover image), CachedNetworkImage, Column, Row, Text, Icon, ElevatedButton.icon, IconButton, Consumer (to track bookmark status), ListTile (for chapter list), AnimatedSize (for synopsis), BackdropFilter (in AppBar). Custom ChapterListItem widget.
  - **Libraries/Plugins:**
    - flutter_riverpod: Provides story details and bookmark status via storyDetailsProvider and isBookmarkedProvider.
    - cached_network_image: Displays the cover image.
    - go_router: Navigates to the reader screen (ReaderScreen).
  - **State Management:** Uses Riverpod:
    - storyDetailsProvider: FutureProvider.family to fetch story details (including story info and chapter list) based on storyId.
    - isBookmarkedProvider: StateNotifierProvider.family to manage the bookmark status of the current story and allow toggling.
    - Uses StatefulWidget and setState to manage the expanded/collapsed state of the synopsis.
  - **Data Storage:** Story details data (StoryDetails, including Story and List<Chapter>) is fetched remotely via the get_story_details RPC on Supabase. Bookmark status is read/written to the User_Bookmarked_Stories table on Supabase via LibraryRepository.
  - **REST API:** Uses Supabase RPC (get_story_details) via the client library.

Feature 4: Reader Screen

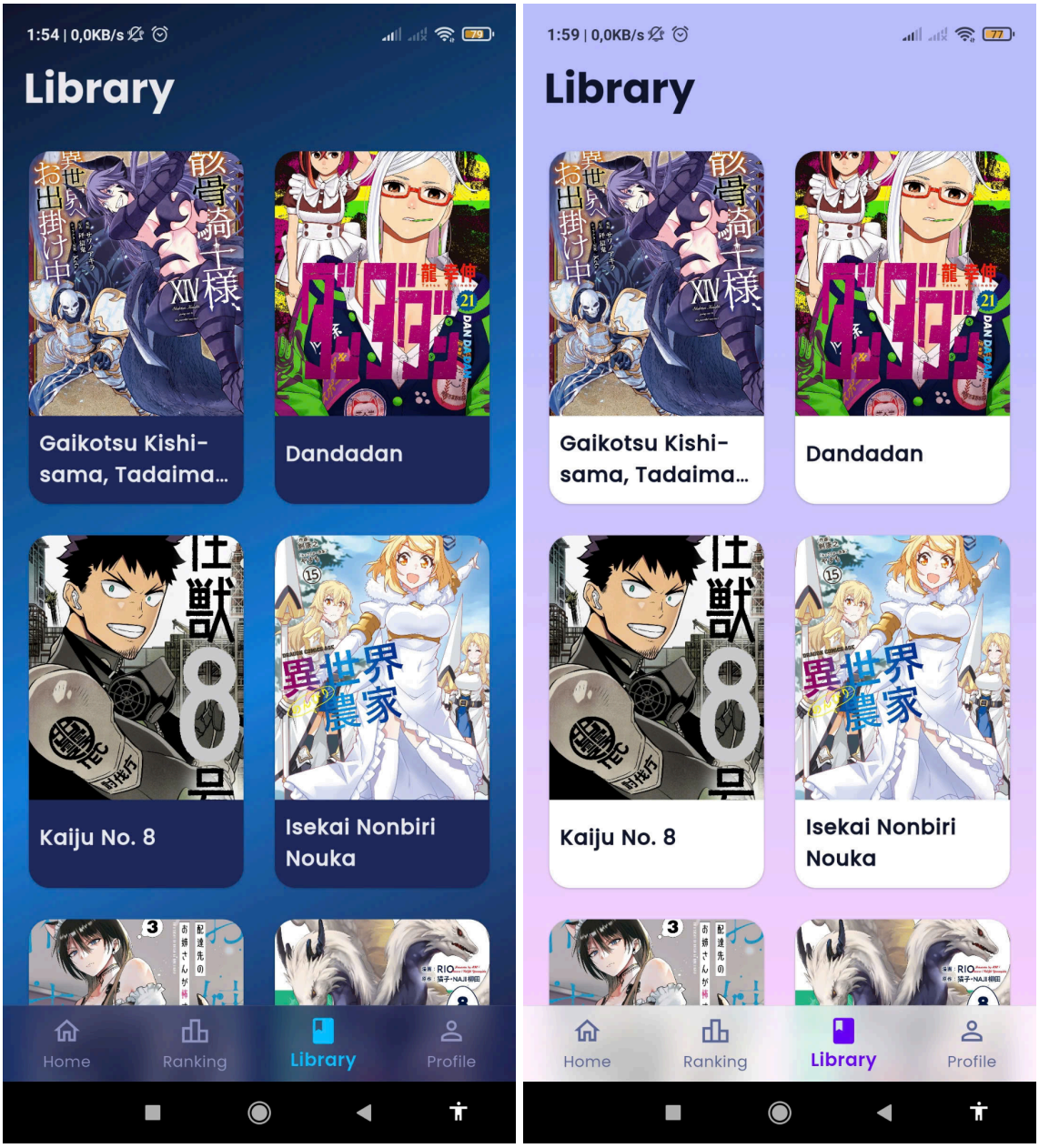
- **Description:**
  Displays the content of a comic chapter as a list of images. Users can show/hide the top navigation bar (AppBar) and bottom bar (BottomBar) by tapping the screen. Supports two reading modes: vertical scroll (Vertical) and horizontal page flip (Horizontal). Provides functions: view the chapter list, open settings (for screen brightness), and access the comment section.
- **Screenshots:**



- **Implementation details:**
  - **Widgets:** Stack, Scaffold, AppBar (custom ReaderAppBar widget), BottomNavigationBar (custom ReaderBottomBar widget), GestureDetector (to show/hide controls), ListView.builder (vertical mode), PageView.builder (horizontal mode), InteractiveViewer (allows image zooming), CachedNetworkImage, Shimmer (image loading effect), ModalBottomSheet (custom ChapterListBottomSheet, CommentsBottomSheet, ReaderSettingsBottomSheet widgets), BackdropFilter (creates glassmorphism effect for AppBar, BottomBar, BottomSheets), AnimatedOpacity.
  - **Libraries/Plugins:**
    - flutter_riverpod: Provides the chapter image URLs via chapterContentProvider.
    - cached_network_image: Displays and caches chapter images.
    - shimmer: Image loading effect.
    - screen_brightness: Adjusts screen brightness in ReaderSettingsBottomSheet.
    - go_router: Navigates between chapters.
  - **State Management:** Uses Riverpod:
    - chapterContentProvider: FutureProvider.family to fetch the list of image URLs based on chapterId.
    - brightnessProvider: StateProvider to manage the brightness value in ReaderSettingsBottomSheet.
    - Uses ConsumerStatefulWidget and setState to manage the show/hide state of controls (_showControls) and the reading mode (_readingMode).
  - **Data Storage:** The list of image URLs (image_urls) is fetched from the Chapter table on Supabase. Comment data is handled separately.
  - **REST API:** None. Interacts with Supabase DB via the client library.

Feature 5: Library Screen

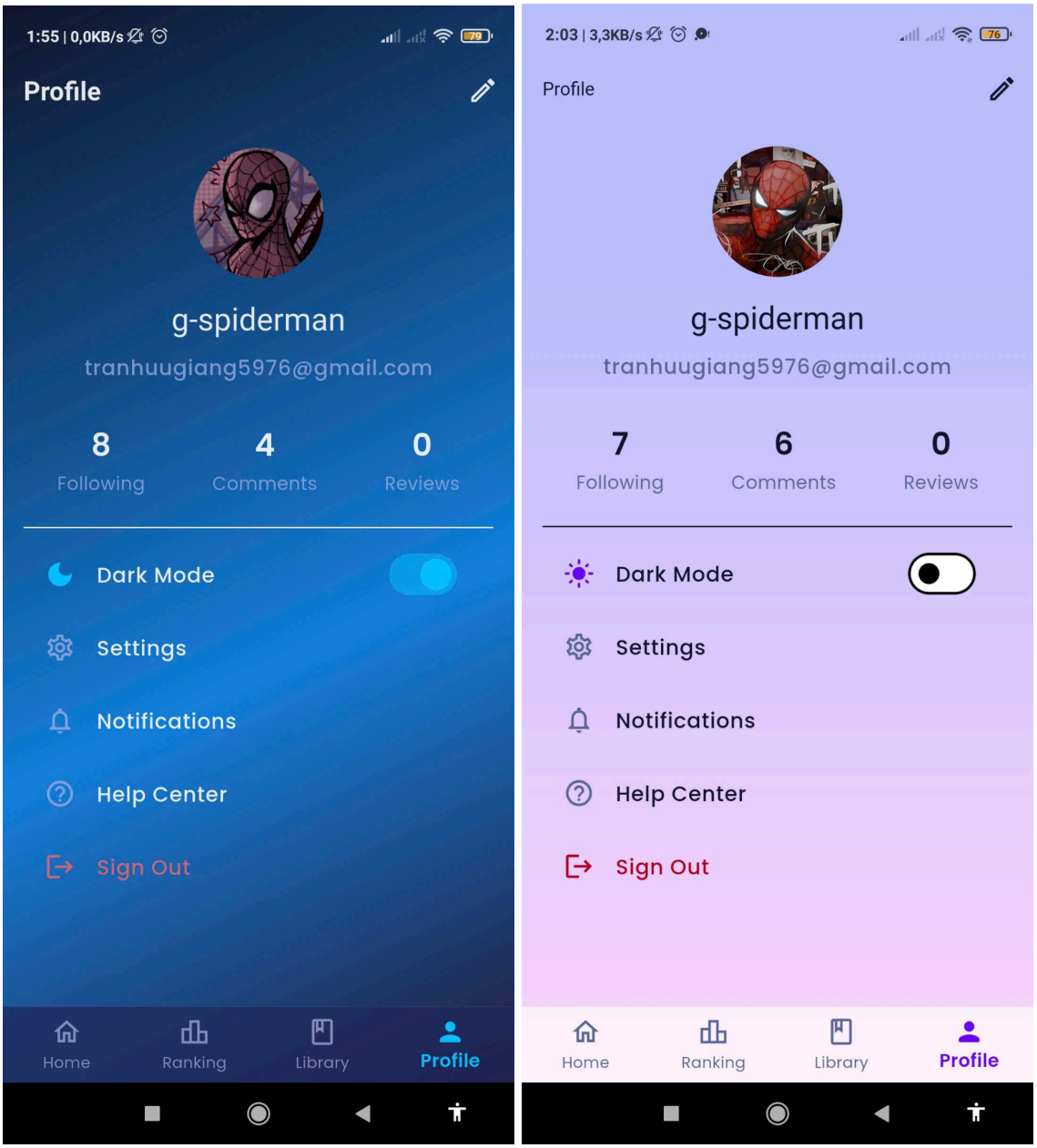
- **Description:**
  Displays a list of stories that the user has bookmarked. Stories are shown in a grid layout. Provides a "Pull to refresh" feature to update the list.
- **Screenshots:**



- **Implementation details:**
  - **Widgets:** Scaffold, AppBar, GridView.builder, StoryCard (custom widget), RefreshIndicator, Shimmer (loading effect), ConsumerWidget, Column, Icon, Text (when library is empty).
  - **Libraries/Plugins:**
    - flutter_riverpod: Provides the list of bookmarked stories via bookmarkedStoriesProvider.
    - cached_network_image: Displays cover images in StoryCard.
    - shimmer: Loading effect.
    - go_router: Navigates to story details when a StoryCard is tapped.
  - **State Management:** Uses Riverpod:
    - bookmarkedStoriesProvider: FutureProvider to fetch the list of bookmarked stories from LibraryRepository. This provider automatically updates when bookmark status changes (due to ref.invalidate in isBookmarkedProvider).
  - **Responsive Design:** The grid layout dynamically changes between 2 columns (mobile) and 4 columns (tablet/desktop) to utilize screen space effectively.
  - **Data Storage:** The list of stories (Story) is fetched remotely by joining the User_Bookmarked_Stories and Story tables on Supabase via LibraryRepository.
  - **REST API:** None. Interacts with Supabase DB via the client library.

Feature 6: Profile Screen

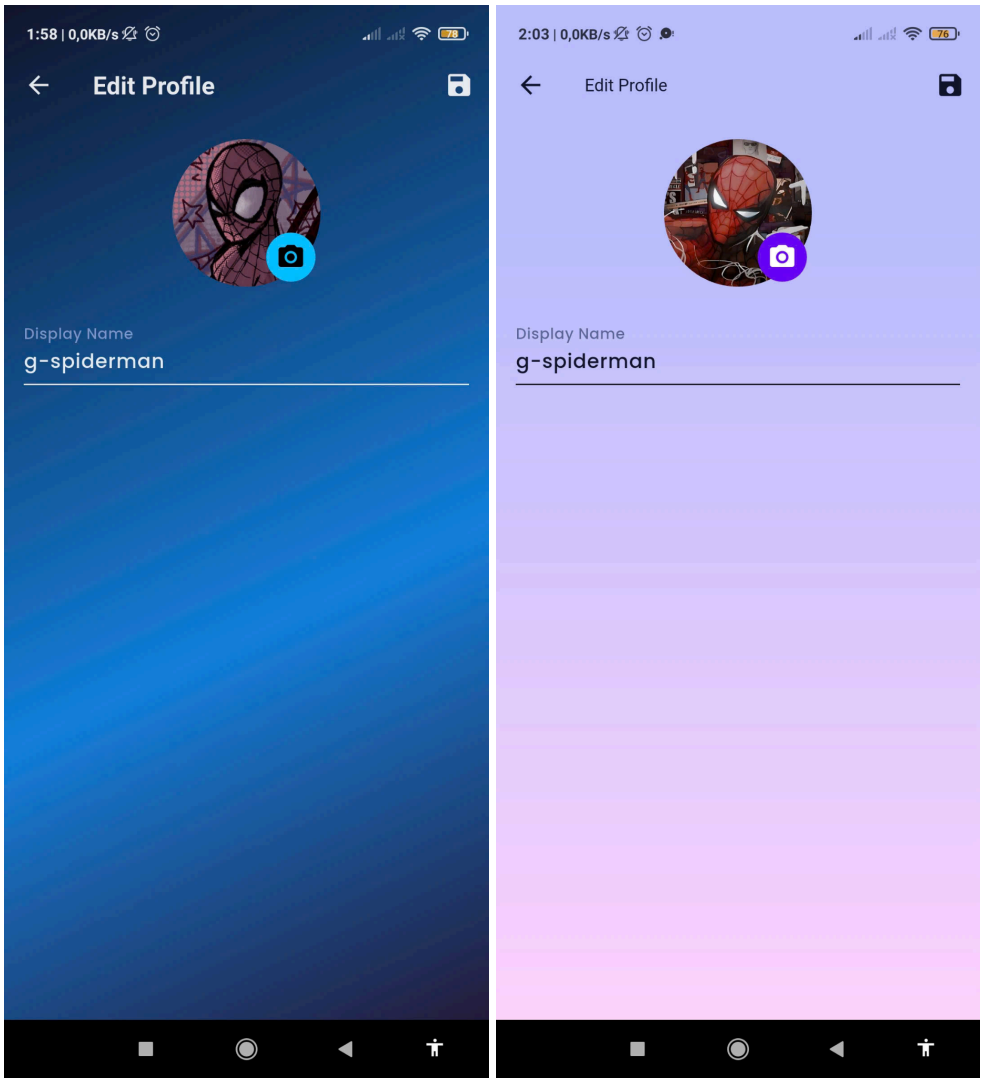
- **Description:**
  Displays the profile information of the currently logged-in user, including avatar, display name, email, and statistics (following count, comment count, review count). Allows navigation to the edit profile screen, logging out, and toggling Dark Mode.
- Screenshots:



- **Implementation details:**
  - **Widgets:** Scaffold, AppBar, ListView, CircleAvatar, Text, ListTile, Divider, ConsumerWidget, Row, Column, SwitchListTile.
  - **Libraries/Plugins:**
    - flutter_riverpod: Provides user profile information via userProfileProvider and manages theme state via themeModeProvider.
    - go_router: Navigates to the edit profile screen (EditProfileScreen).
  - **State Management:** Uses Riverpod:
    - userProfileProvider: StreamProvider (or FutureProvider if real-time is not needed) to fetch the Profile information from ProfileRepository.
    - profileRepositoryProvider: Provides an instance of ProfileRepositoryImpl.
    - themeModeProvider: Manages the application's theme mode (Light/Dark/System).
  - **Data Storage:** Profile information (Profile) and statistics are fetched remotely via the get_user_profile_details RPC on Supabase. Email is obtained from Supabase.instance.client.auth.currentUser.
  - **REST API:** Uses Supabase RPC (get_user_profile_details) via the client library.

Feature 7: Edit Profile Screen

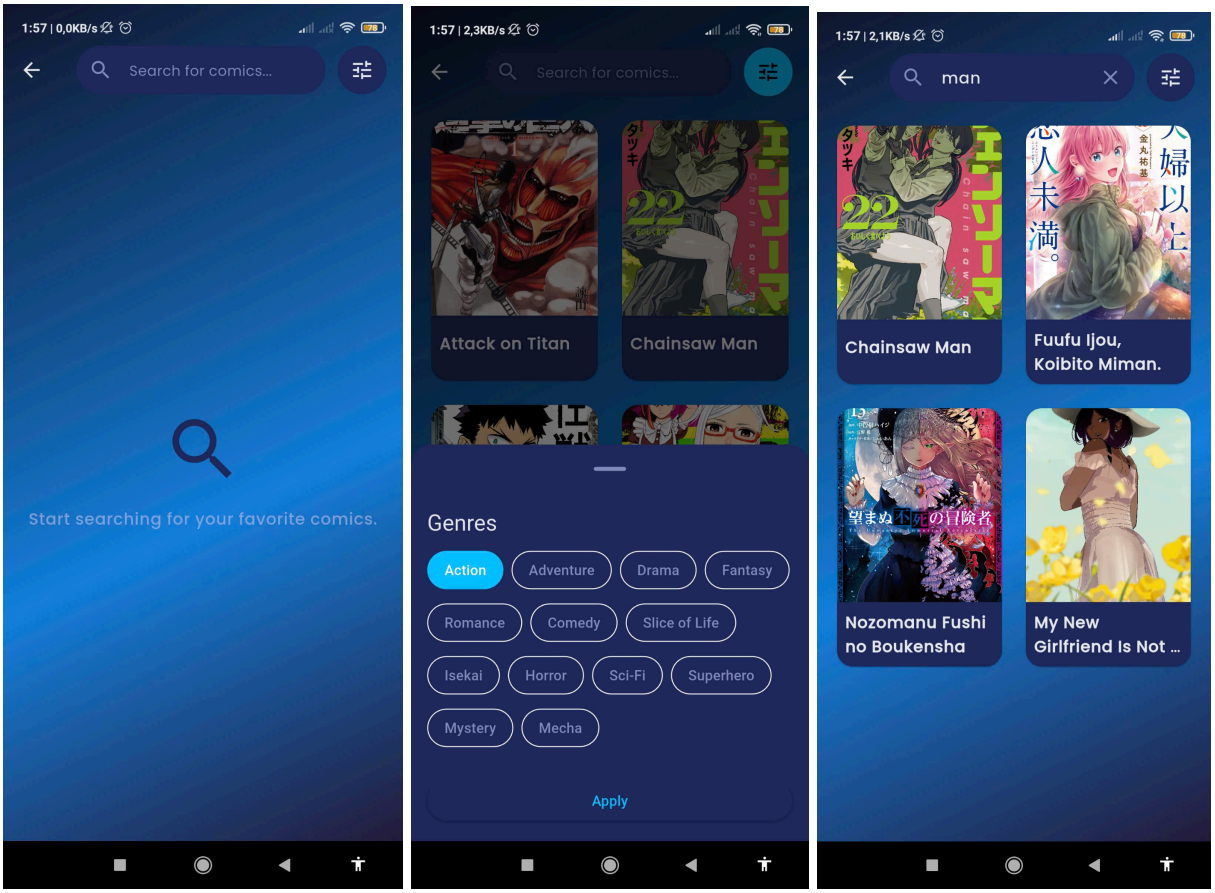
- **Description:**
  Allows users to update their display name and profile picture. The profile picture can be selected from the device's photo library. Success or error notifications are displayed upon saving
- **Screenshots:**



- **Implementation details:**
  - **Widgets:** Scaffold, AppBar, SingleChildScrollView, Form, Column, Stack, CircleAvatar, IconButton.filled, TextFormField, CircularProgressIndicator, ConsumerStatefulWidget, Flushbar.
  - **Libraries/Plugins:**
    - flutter_riverpod: Updates the profile via profileUpdaterProvider and reads initial data from userProfileProvider.
    - image_picker: Selects images from the device library.
    - go_router: Navigates back to the Profile screen after saving.
  - **State Management:** Uses Riverpod:
    - profileUpdaterProvider: StateNotifierProvider to manage the state (loading, error) of the profile update operation. Calls the updateProfile method to perform the update.
    - userProfileProvider: Reads the initial profile data for display. This provider is also invalidated after a successful update to refresh the data on the Profile screen.
    - Uses ConsumerStatefulWidget and setState to manage the local _avatarFile state.
  - **Data Storage:** The display name is updated in the profiles table on Supabase. The avatar image (avatarFile) is uploaded to Supabase Storage (bucket public_avatar), and then the public URL is saved to the avatar_url column in the profiles table via ProfileRepository.
  - **REST API:** None. Interacts with Supabase DB and Storage via the client library.

Feature 8: Search and Filter Screen

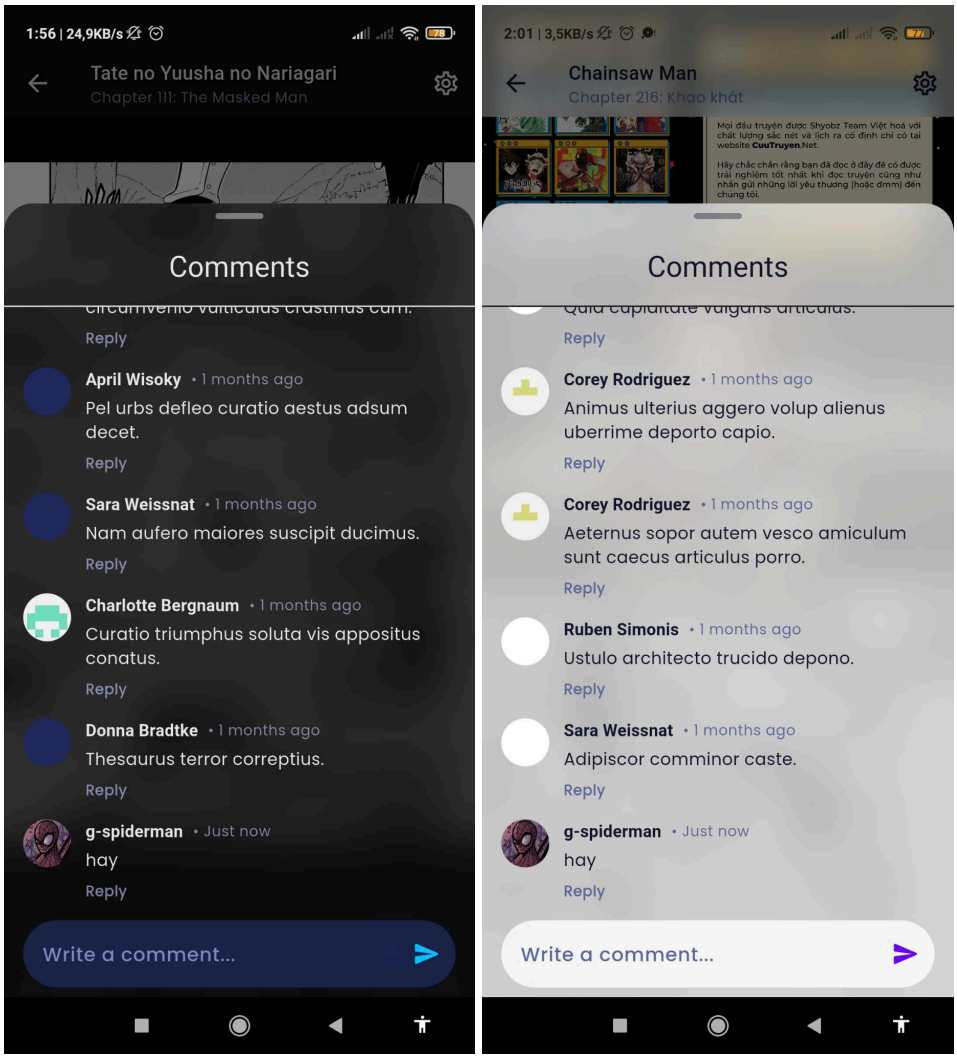
- **Description:**
  Allows users to enter keywords to search for comics by title and filter results by genres. Search results are displayed in a grid with animation. Includes a debounce mechanism to limit API calls while the user is typing.
- **Screenshots:**



- **Implementation details:**
  - **Widgets:** Scaffold, AppBar, TextField, IconButton (to clear query), GridView.builder, StoryCard (custom widget), Center, Text, CircularProgressIndicator, ConsumerWidget, FilterBottomSheet.
  - **Libraries/Plugins:**
    - flutter_riverpod: Manages the search query, filter options, and search results.
    - cached_network_image: Displays cover images in StoryCard.
    - go_router: Navigates to story details when a StoryCard is tapped.
    - flutter_animate: Adds fade-in and slide animations to search results.
  - **State Management:** Uses Riverpod:
    - searchQueryProvider: StateProvider to store the user's input query.
    - filterOptionsProvider: StateNotifierProvider to manage selected genres for filtering.
    - allGenresProvider: FutureProvider to fetch the list of available genres.
    - searchResultsProvider: AsyncNotifierProvider to perform the search. This provider listens to both searchQueryProvider and filterOptionsProvider, uses a Timer to create a debounce (waits 500ms), and calls SearchRepository.
  - **Responsive Design:** Adapts the grid layout based on screen width using MediaQuery. On screens wider than 600px, it displays 4 columns with an adjusted aspect ratio (0.72). On smaller screens, it defaults to 2 columns with a taller aspect ratio (0.65) to fit comic titles better.
  - **Data Storage:** Search results are fetched remotely via SearchRepository. It queries the Story table (using ilike for search) and performs an inner join with the Story_Genre table if filters are applied.
  - **REST API:** None. Interacts with Supabase DB via the client library.

Feature 9: Comments

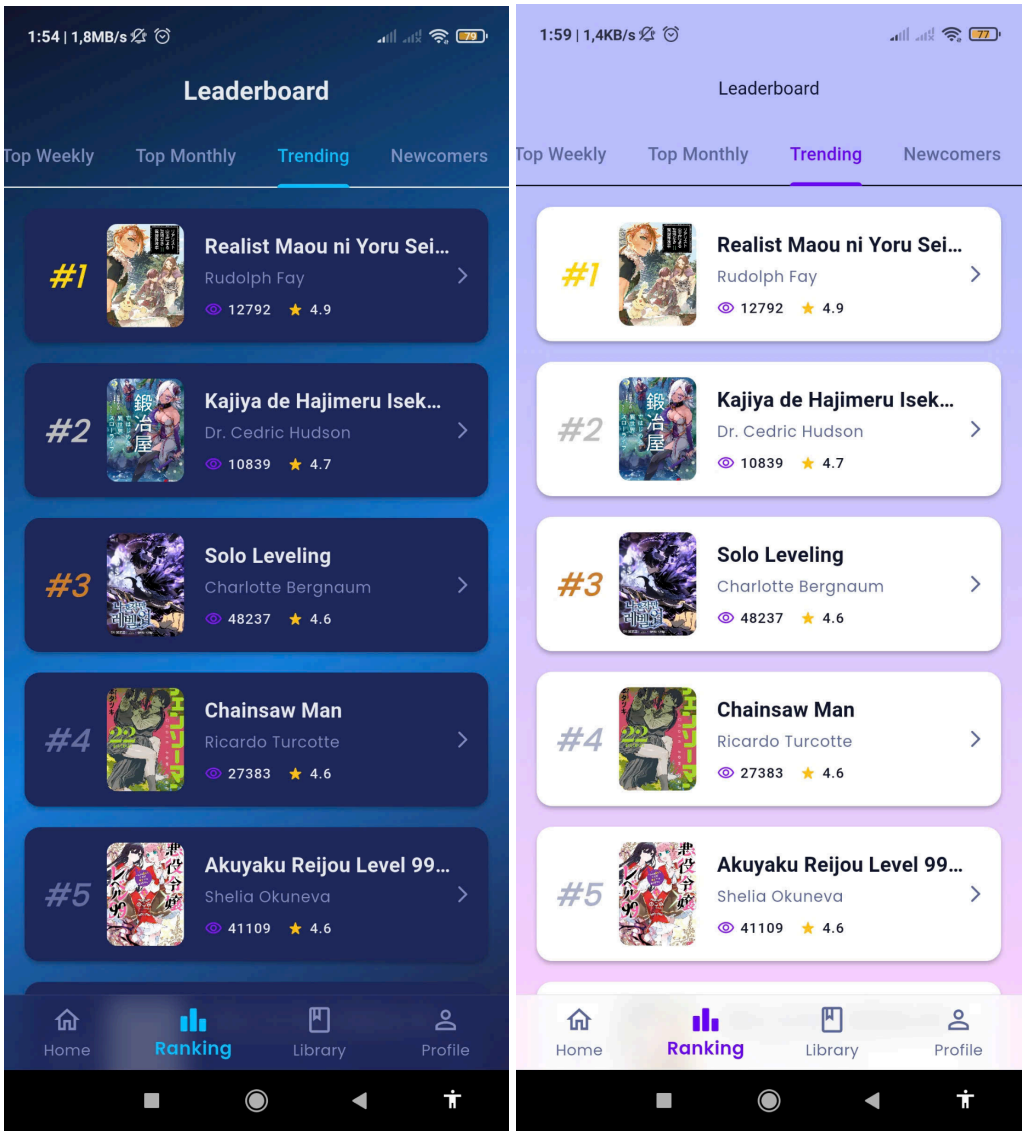
- **Description:**
  Displayed in a BottomSheet on the ReaderScreen. Allows users to view comments for a chapter in a nested tree structure, post new comments, and reply to existing comments. Uses Supabase Realtime for live comment updates.
- **Screenshots:**



- **Implementation details:**
  - **Widgets:** ModalBottomSheet, BackdropFilter, Column, ListView.builder (custom CommentListItem widget), TextField, IconButton, CircleAvatar(CachedNetworkImage), Shimmer (loading effect), Row.
  - **Libraries/Plugins:**
    - flutter_riverpod: Manages the comment list (commentsStreamProvider, nestedCommentsProvider) and comment posting state (commentPostControllerProvider).
    - supabase_flutter: Interacts with Supabase DB (table Comment) and Supabase Realtime.
    - cached_network_image: Displays commenter avatars.
    - shimmer: Loading effect.
  - **State Management:** Uses Riverpod:
    - commentRepositoryProvider: Provides an instance of CommentRepositoryImpl.
    - commentsStreamProvider: StreamProvider.family to listen for real-time changes from the Comment table on Supabase for a specific chapterId.
    - nestedCommentsProvider: Provider.family to transform the flat list of comments from commentsStreamProvider into a tree structure (reply hierarchy).
    - commentPostControllerProvider: StateNotifierProvider to manage the state (loading, error) when posting a new comment or reply.
  - **Data Storage:** Comment data (CommentEntity) is stored and fetched from the Comment table on Supabase. Utilizes Supabase Realtime feature to automatically update the UI when new comments or changes occur.
  - **REST API:** None. Interacts with Supabase DB and Realtime via the client library.

Feature 10: Ranking Screen

- **Description:** Displays leaderboards for comics categorized by different criteria: Top Weekly, Top Monthly, Trending, and Newcomers. Users can switch between these categories using a tab bar. The list highlights the top 3 comics with distinct colors for their rank numbers (Gold, Silver, Bronze) for better visual distinction.
- **Screenshots:**



- **Implementation details:**
    - **Widgets:** Scaffold, AppBar, DefaultTabController, TabBar, Tab, TabBarView, ListView.builder, Card, ListTile, Icon, Text, ConsumerWidget.
    - **Libraries/Plugins:**
        - flutter_riverpod: Used for accessing theme/global state via ConsumerWidget.
        - cached_network_image: Optimizes the loading of story cover images in the list.
        - go_router: Handles navigation to the StoryDetailsScreen when a user selects a story from the list.
    - **State Management:** Uses Riverpod:
        - Uses Flutter's built-in DefaultTabController to manage the state of the selected tab and switch views (Weekly, Monthly, etc.) without needing a complex external state provider.
    - **Data Storage:** Currently utilizes mock data (dummy data) within the UI code to demonstrate the layout and logic for ranking display. Designed to fetch and sort data from the Story table on Supabase in future implementations.
    - **REST API:** None. Interacts with Supabase DB via the client library.