

Task: Implement a Responsive Home Screen for Slash App

Objective

Develop a responsive UI for the Home Screen that adapts seamlessly to both mobile and web platforms, based on the provided Figma design and dummy data. The implementation should follow clean architecture principles and use the MVVM pattern with Bloc or Cubit for state management.

Requirements

- **Architecture:** Follow clean architecture principles and MVVM pattern.
- **State Management:** Use Bloc or Cubit for state management.
- **UI:** Ensure the UI is responsive using appropriate layout widgets.
- **Data:** Fetch and display data from the provided dummy data.
- **Design:** Maintain design consistency as per the Figma design provided.

Provided Resources

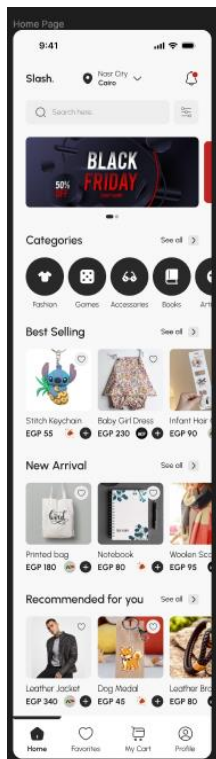
1. **Figma Design:** [Link.](#)
2. **Dummy Data:** [Link.](#)

Screen Details

- **Design References:**

- Mobile Layout: Refer to the mobile layout in the Figma design.
- Web Layout: use the same design for mobile layout in the Figma design.

Note: This task description focuses specifically on the Home Screen and allows for flexibility in adjusting widget placements to meet **web** UI requirements.



Technical Details

- **Responsive Design:**
 - Use `LayoutBuilder`, `MediaQuery`, and other responsive layout widgets.
 - Implement breakpoints for different screen sizes to adjust the layout accordingly.
- **Flutter Widgets:**
 - Mobile layout could use `ListView`, `Column`, or `SingleChildScrollView`.
 - Web layout could use `GridView`, `Row`, or `Expanded`.
- **Architecture:**
 - Follow clean architecture principles to structure the code.
 - Implement the MVVM pattern to separate concerns.
- **State Management:**
 - Use `Bloc` or `Cubit` for managing state across the application.

Implementation Outline

3. **Setup:**
 - Import necessary packages.
 - Set up a basic Flutter app structure.
4. **Project Structure:**
 - **Data Layer:** Define data models and repositories.
 - **Domain Layer:** Define use cases and entities.
 - **Presentation Layer:** Define view models, blocs/cubits, and UI components.
5. **Responsive Home Screen Widget:**
 - Create a `ResponsiveHomeScreen` widget.
 - Use `LayoutBuilder` to switch between mobile and web layouts.
6. **Layout Widget:**
 - Implement the mobile and web version of the Home Screen.
7. **State Management:**
 - Implement `Bloc` or `Cubit` to manage the state of the Home Screen.

Note

- The developer can change the widget placements on the Home Screen to ensure the UI fits web requirements appropriately.

Evaluation Criteria

- **Functionality:** Correctness and functionality of the implemented Home Screen.
- **Design Consistency:** Adherence to the design specifications in Figma.
- **Responsiveness:** Adaptability of the UI across different screen sizes.
- **Code Quality:** Use of best practices, clean architecture, and maintainable code.
- **State Management:** Proper implementation of Bloc or Cubit for state management.

Submission

- **GitHub Repository:**
 - Submit the complete Flutter project in a public GitHub repository.
 - Ensure the repository is accessible and includes all necessary files and documentation.
 - Provide the GitHub repository link upon submission.
- **Instructions:**
 - Ensure the app runs without errors on both mobile and web platforms.
 - Include a README file with instructions on how to set up and run the project
- Submit your task through [google form Link](#).
- Deadline is 5 days from receiving that task by max hours **11:59 pm**