



FLUTTER AND PWA LAB

BY ISHWARI DATIR

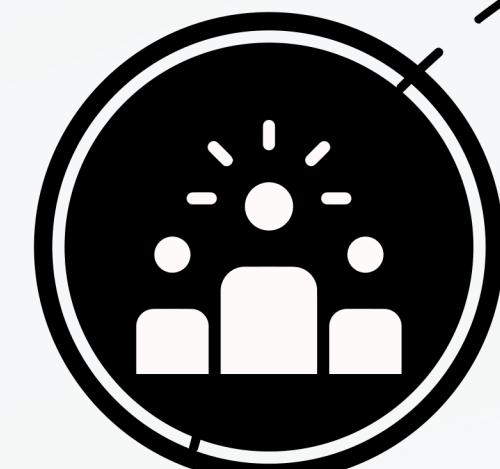
FLUTTER SECTION

- 01** OVERVIEW-FURNITURE APP
- 02** FUNCTIONALITIES
- 03** AUTHENTICATION
- 04** HOME SCREEN
- 05** PRODUCT DESCRIPTION
- 06** BUILD GUIDE
- 07** FURNITURE QUIZ
- 08** LITERATURE SURVEY

FUNCTIONALITIES

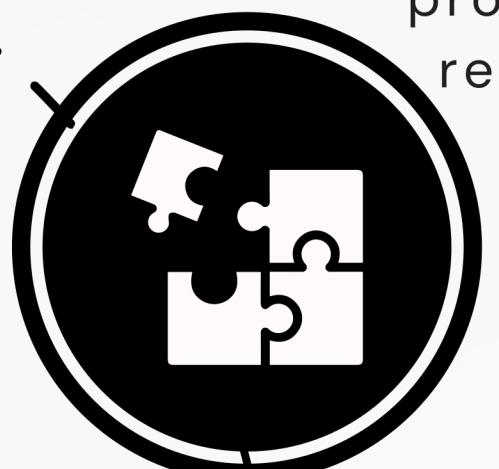
Furniture Display

Users can browse through a collection of furniture items showcased in the app, allowing them to view details, images, and descriptions of each piece.



Build Guide

A build guide section is included in the app, offering users step-by-step instructions on how to assemble or build specific furniture pieces featured in the app. This adds value by providing practical guidance to users interested in DIY projects.



Fun Quiz

The app includes a fun and interactive quiz comprising three questions related to furniture preferences and style. Based on the user's responses, the quiz determines their furniture style, providing personalized recommendations or insights.



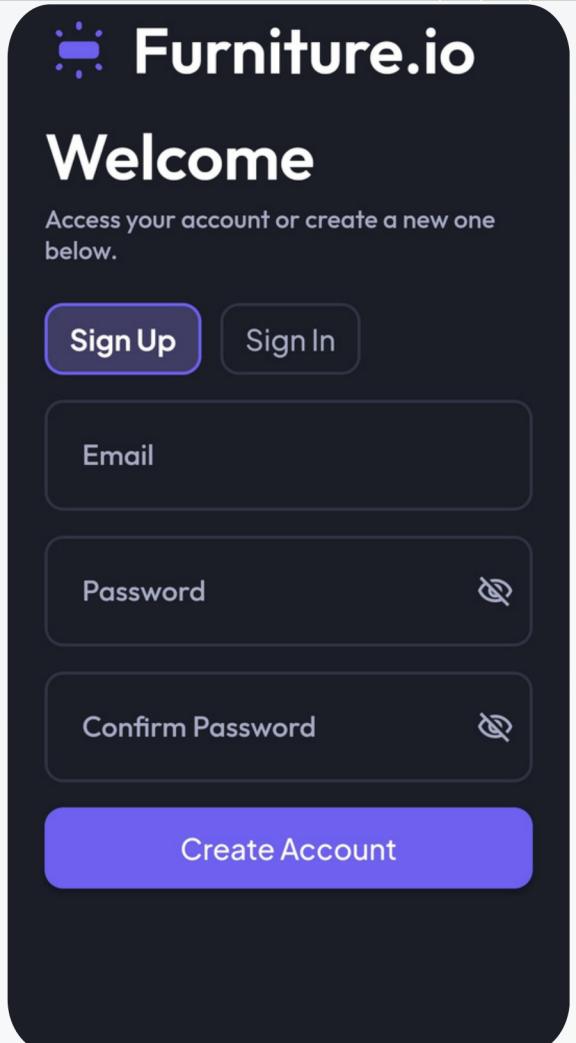
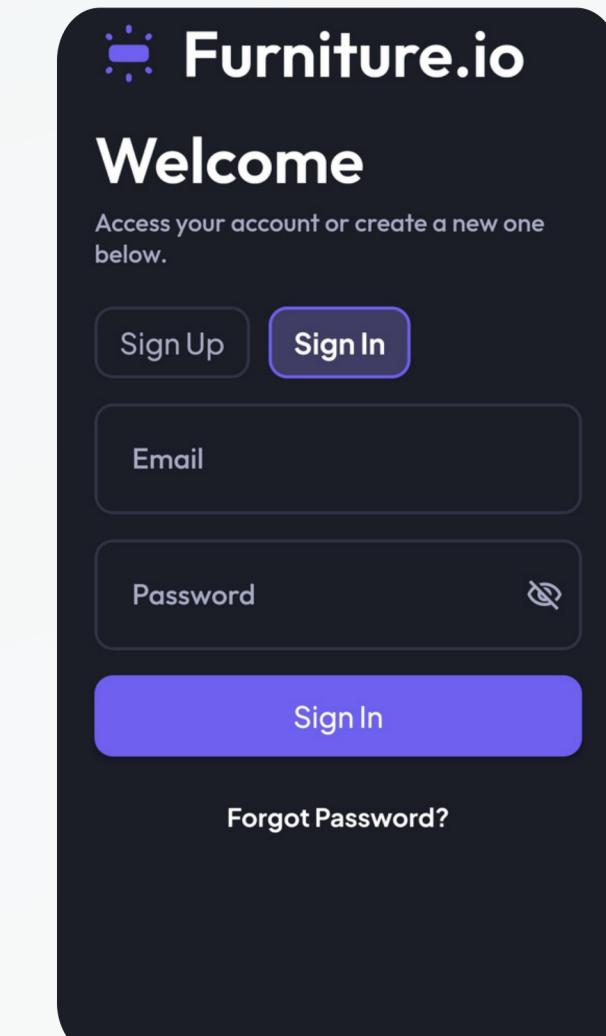
1. AUTHENTICATION

SignUp

- Signup Process: Users who do not have an account can sign up by providing their full name, phone number, email, and password on the signup page. This information is essential for creating a personalized account and accessing the app's features.

- Login Process: Once registered, users can log in using their email and password on the login screen. This authentication step ensures secure access to the app's functionalities and personalized content.

Login



HOME SCREEN

1. Staggered Grid View:

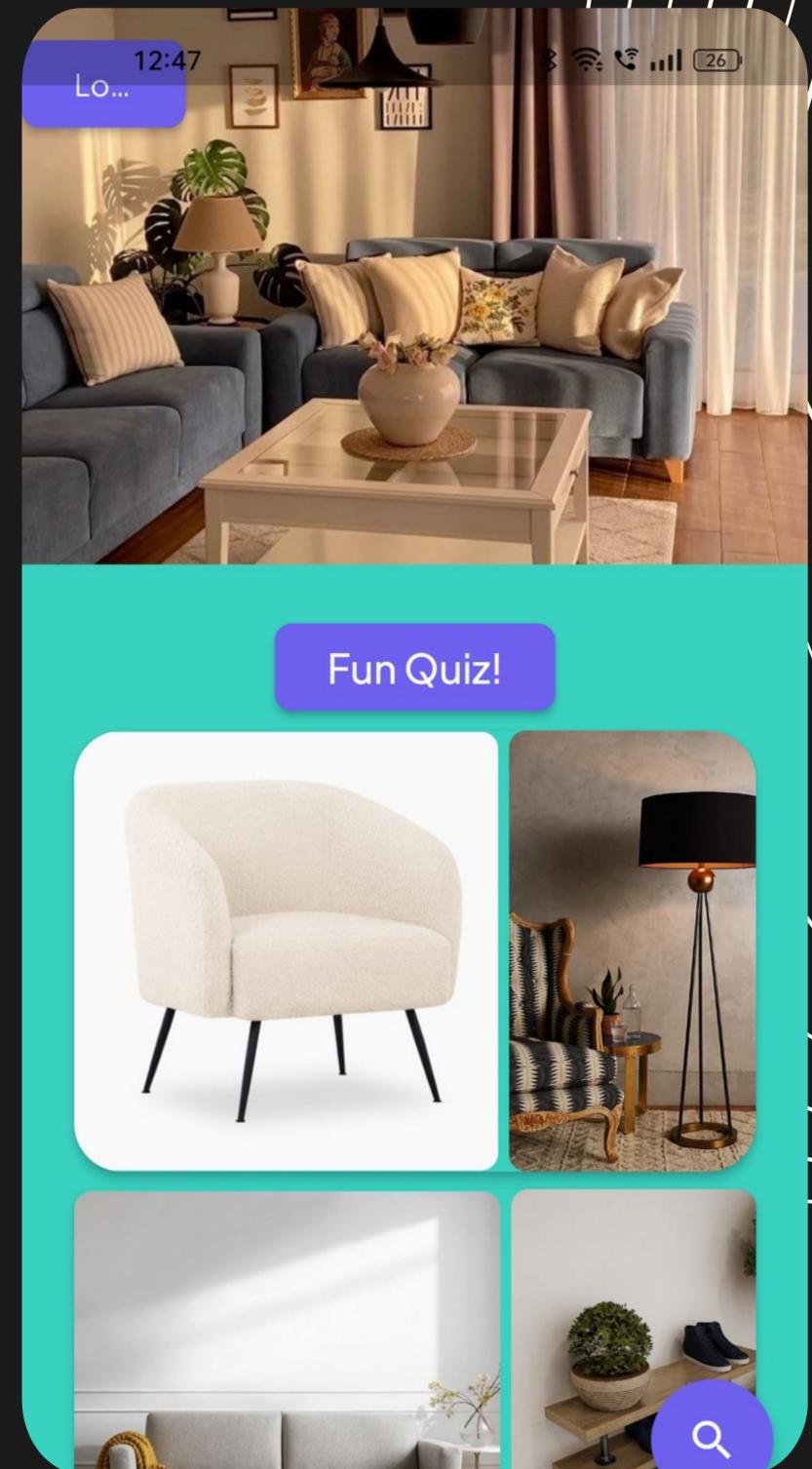
- Displays all furniture items in a visually appealing grid layout on the home screen.

2. Bottom Navbar:

- Toggles between different screens, including the all users page and the user's profile page.

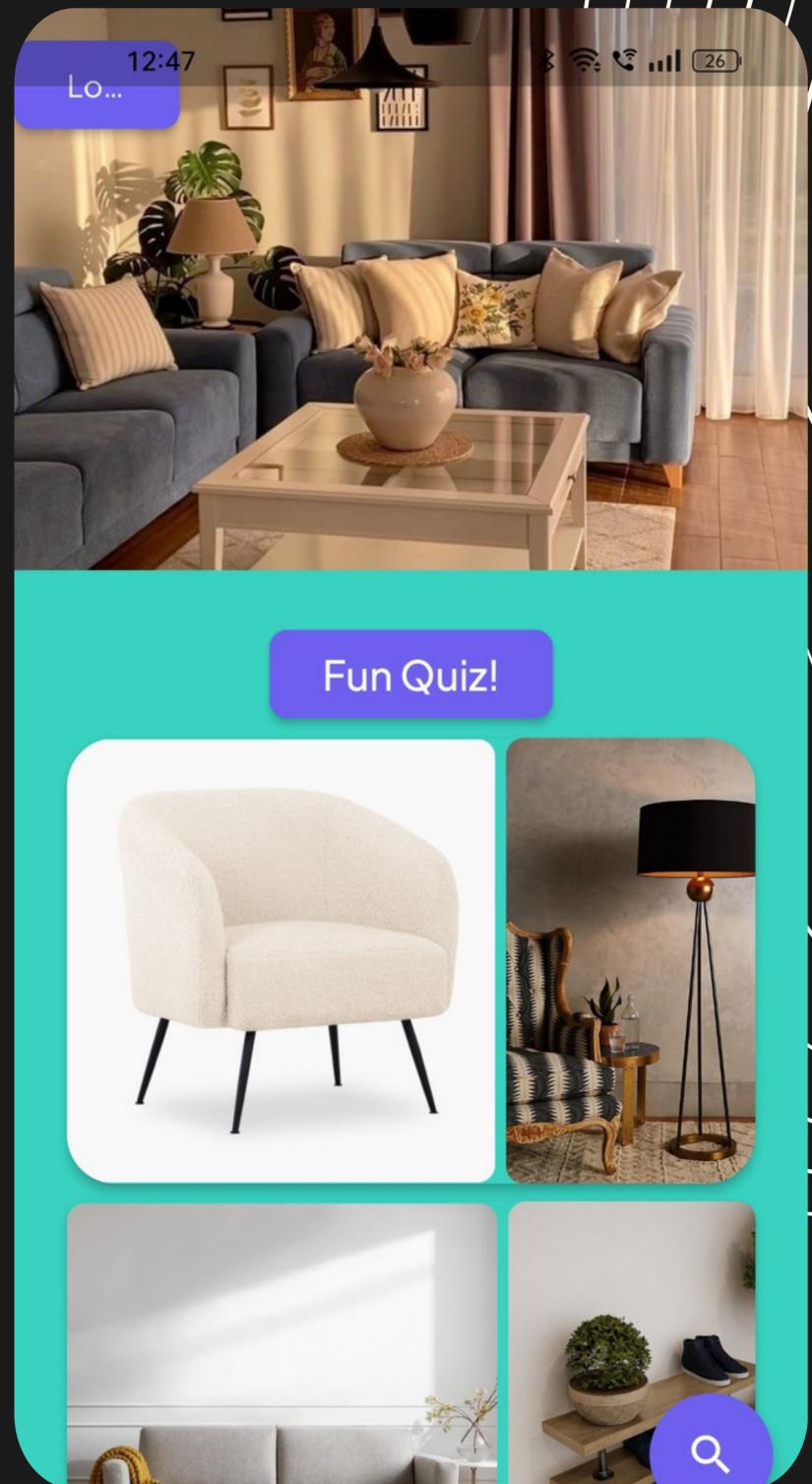
3. Top Buttons:

- "View List" Button: Allows users to switch to a list view of furniture items for easier browsing.
- "Take Quiz" Button: Redirects users to the furniture style quiz mentioned earlier.
- "Logout" Button: Enables users to log out of their account and exit the app securely.



SEARCH

- Implemented search functionality for furniture items, allowing users to input item names for search.
- Displays search results in a user-friendly list view format for easy browsing.
- Enables users to click on a specific item from the list to navigate to a dedicated page or screen showing detailed information about the selected furniture piece.
- Enhances user experience by providing quick and convenient access to relevant item details based on search queries.



PRODUCT DESCRIPTION

1. Horizontally Scrollable Grid:

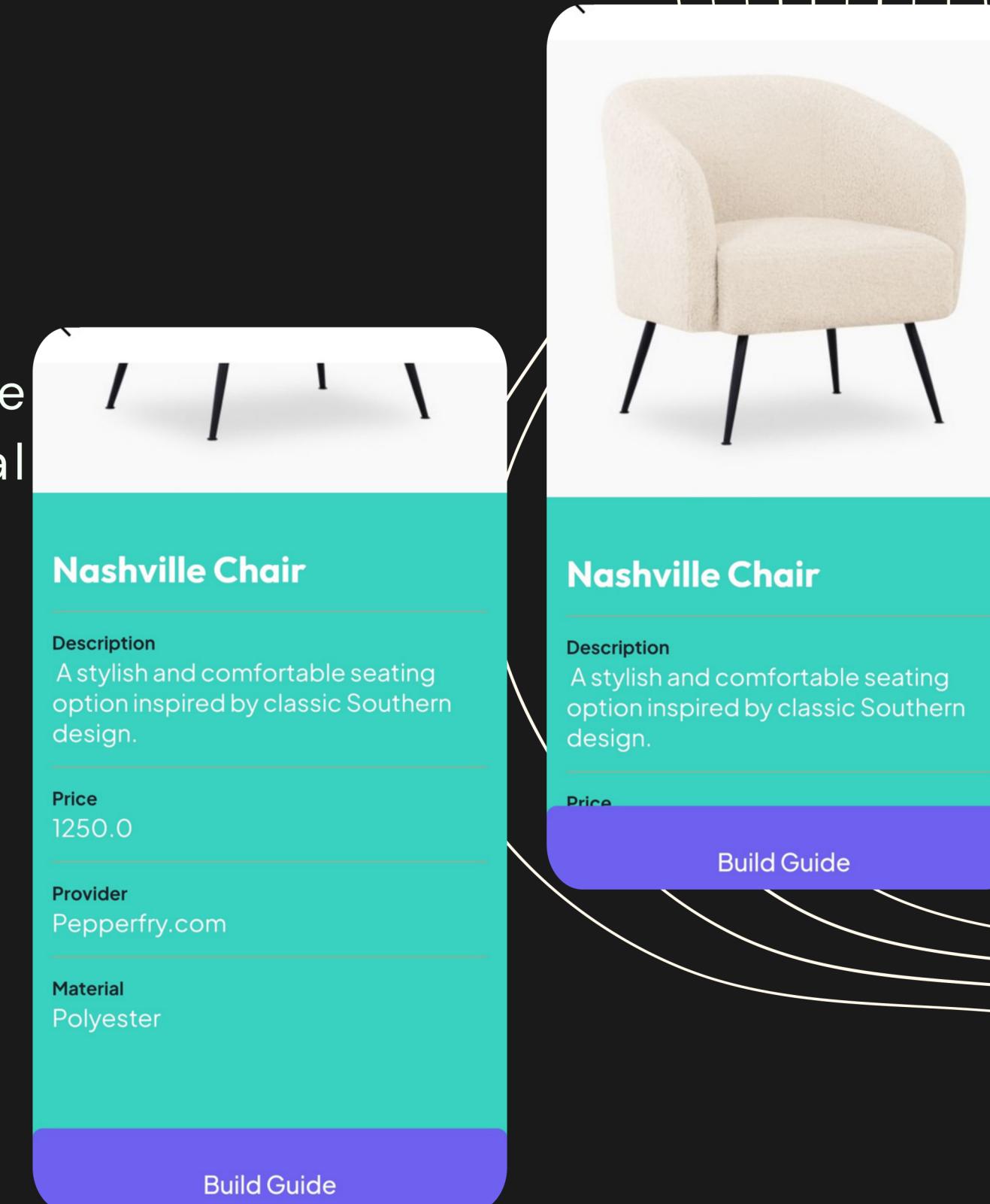
- Displays all furniture items horizontally, allowing users to scroll through and view different products.

2. Grid for Different Angles:

- Below the scrollable grid, there's another grid showcasing the current furniture piece from various angles, enhancing the visual experience.

3. Product Details Section:

- Name: Clearly states the name of the furniture item for easy identification.
- Price: Indicates the price of the product, helping users make purchase decisions.
- Rating in Stars: Users can give ratings to the product using a star-based rating system, providing feedback on product satisfaction.
- Short Description: Offers a brief overview or description of the furniture item, highlighting key features or characteristics.



BUILD GUIDE

Build Guide Section:

- Specific to Each Furniture Item: The build guide is tailored to the furniture item being viewed, ensuring relevance and accuracy in assembly instructions.
- Detailed Assembly Instructions: Provides step-by-step guidance on how to assemble the furniture piece, including diagrams or illustrations for clarity.
- Usage Instructions: Along with assembly instructions, the build guide also provides the approximate time taken to assemble, the level of difficulty to assemble and the rating for satisfaction.
- User-Friendly Format: Presented in a user-friendly format, the build guide enhances the user experience by making it easy to follow and understand the assembly process.

Build Guide



Nashville Chair

A sleek and modern lighting solution featuring a tripod base for stability and contemporary style.

How to Build:

- Lay out all the chair components and tools needed for assembly.
- Start by attaching the backrest to the seat frame using the provided screws.
- Next, attach the armrests to the sides of the seat frame, ensuring they are aligned properly.
- Secure the legs to the bottom of the seat frame, making sure they are evenly positioned for stability.
- Once all components are attached, tighten all screws to ensure a sturdy build.
- Flip the chair upright and test for stability and comfort before use.

Statistics

3-4 hrs Medium 3/5

FURNITURE QUIZ

1. Quiz Questions:

- Preferred Color: Users are asked to choose their preferred color from options like red, blue, and green.
- Preferred Material: Users select their preferred material for furniture, such as wood, metal, or plastic.
- Style Preference: Users indicate their style preference, such as minimalist, classic, or eclectic.

2. Quiz Result:

- Based on the user's responses to these questions, the quiz algorithm determines their recommended furniture type.
- For example, if a user prefers blue, wood, and classic style, the quiz might recommend a classic wooden table in blue.

What is your
preferred colour
in furniture?

- Blue
- Red
- Green
- Black

Next Question

Your
Recommended
Furniture is :

Lava Lamp!

Go to Home

LITERATURE SURVEY

Name of the paper	Author/Year	Focus/Relevance
Decal App: An Augmented Reality-based Flutter Application	2023/ IJRASET	Building an AR furniture app with Flutter
Augmented Reality based Furniture Application	2014/ Sai Gubala et al.	3D room planning app for adaptable furniture
Decal App: An Augmented Reality-based Flutter Application	2011/Various	Comparison of cross-platform mobile app frameworks

PWA SECTION

- 01** OVERVIEW-FURNITURE APP
- 02** HOME PAGE
- 03** MANIFEST.JSON
- 04** SERVICE WORKER
- 05** PUSH NOTIFICATIONS
- 06** FETCH EVENTS
- 07** LIGHTHOUSE TOOL

MY PWA PROJECT: CLOTHING ECOMMERCE

Overview:



The Clothing Ecommerce PWA is a modern web app tailored for shopping clothing items. With offline support and push notifications, it ensures seamless browsing and timely updates for users. Its responsive design and manifest file enable easy installation and access from any device, delivering a smooth and engaging shopping experience.

HOME PAGE

Welcome to Our Ecommerce Store

[Home](#) [Shop](#) [Contact Us](#)

Browsing Categories

Shirts

Pants

Featured Products





MANIFEST.JSON

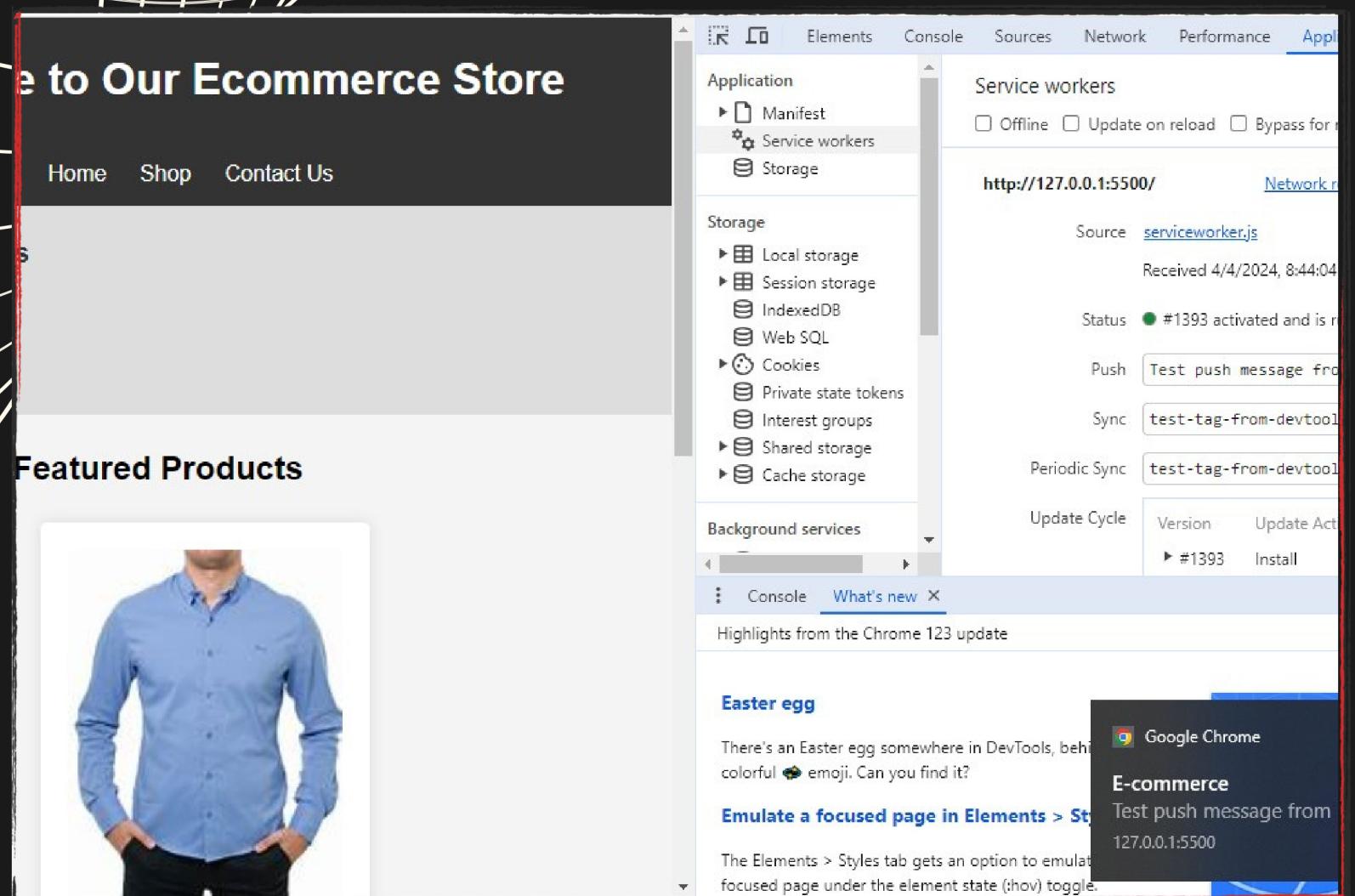
The `manifest.json` file plays a crucial role in defining the behavior and appearance of Progressive Web Apps (PWAs). It contains essential metadata such as the app's name, short name, description, icons in various sizes and formats, start URL, display mode (fullscreen, standalone, browser), theme colors for styling, background color during loading, default orientation (portrait or landscape), navigation scope, default language and text direction, and manifest version. This file ensures a consistent and engaging user experience across different devices and platforms by providing crucial information about the app's identity, functionality, and visual presentation. Developers use the `manifest.json` file to optimize PWAs for discoverability, accessibility, and usability, whether accessed through web browsers or installed as standalone applications on users' devices.



SERVICE WORKER

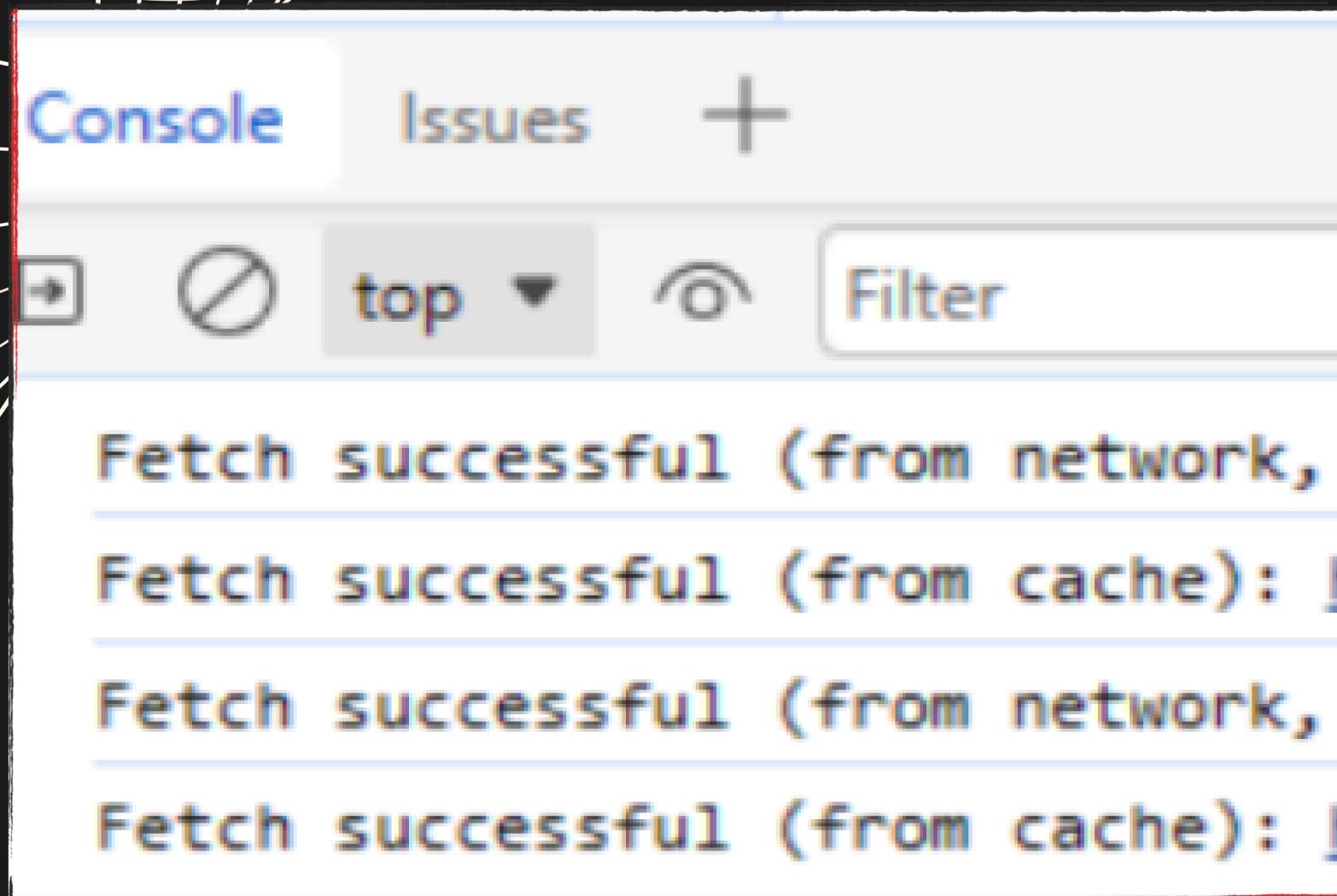
The service worker is a critical component of Progressive Web Apps (PWAs) responsible for enabling offline functionality, caching resources, and managing push notifications. It is a JavaScript file that runs separately from the main browser thread, allowing PWAs to work reliably even with limited or no internet connectivity. The service worker intercepts network requests, enabling the app to serve cached content when offline and reducing load times by fetching resources from the cache when available. Additionally, service workers enable push notifications, allowing apps to engage users with timely updates and notifications even when the app is not actively open in the browser. By leveraging the service worker, PWAs can deliver a seamless and responsive user experience, bridging the gap between web and native applications while offering enhanced performance and functionality across various devices and network conditions.

PUSH NOTIFICATIONS



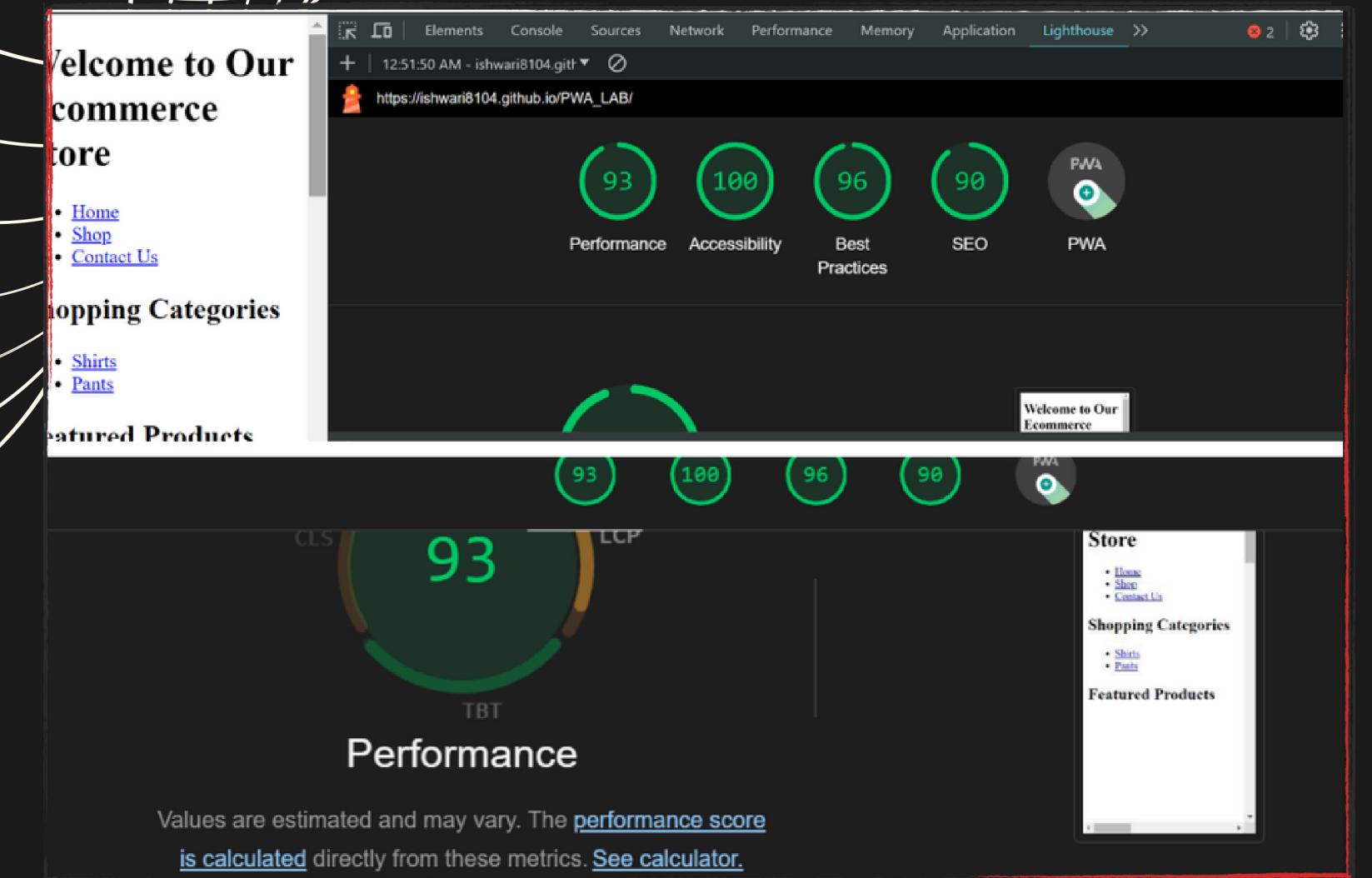
In Progressive Web Apps (PWAs), the `sync` event is a crucial feature that allows background synchronization of data when the device reconnects to the internet. It enables PWAs to perform tasks such as sending queued requests, updating cached data, or syncing user data with the server, even when the app is not actively in use. By leveraging the `sync` event, PWAs can provide a seamless offline experience and ensure that important tasks are executed reliably, regardless of network availability. Developers can register sync event listeners in service workers to handle specific synchronization tasks, ensuring that data remains up-to-date and consistent across devices.

FETCH EVENTS



The `fetch` event in PWAs is triggered whenever the browser fetches a resource like HTML, CSS, JavaScript, or other files. It allows developers to intercept network requests and define custom responses, enabling powerful features like caching, offline support, and dynamic content delivery. By handling the `fetch` event in service workers, PWAs can serve cached resources when offline, prefetch assets for faster loading, or implement custom logic for network requests. This event plays a crucial role in optimizing performance and providing a reliable user experience across various network conditions in PWAs.

LIGHTHOUSE



Lighthouse is an analysis tool provided by Google that evaluates web pages and web applications, including Progressive Web Apps (PWAs), against a set of best practices and performance metrics. For PWAs specifically, Lighthouse assesses various aspects such as performance, accessibility, best practices, SEO, and Progressive Web App features. It provides developers with insights and recommendations to improve the quality and user experience of their PWAs. By running Lighthouse audits, developers can identify areas for optimization, ensure compliance with PWA standards, and enhance the overall performance and usability of their applications.

THANK YOU!!