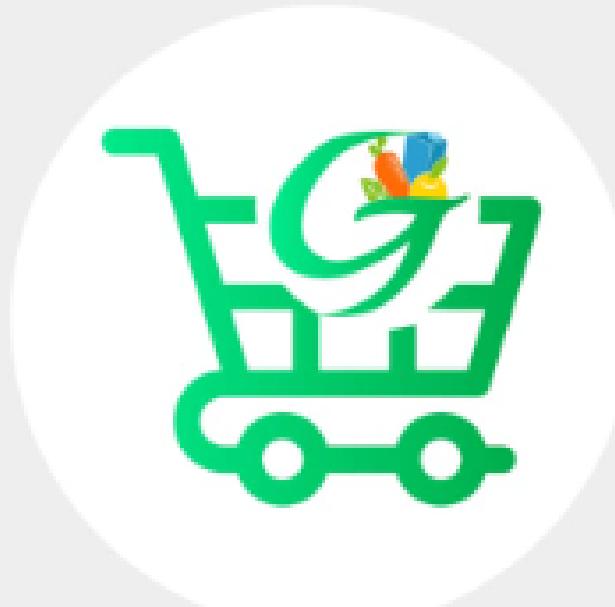


# Grocery



Email

Password

Login

Forgot Password

 Login with Google

Not a User? [Register](#)

Home

About Us

Service

Contact

# Grocery App

"Effortlessly shop for all your essentials with our intuitive grocery app."

Name: Eppanapelli Kirti  
Subject: MP Lab



# Table of Content

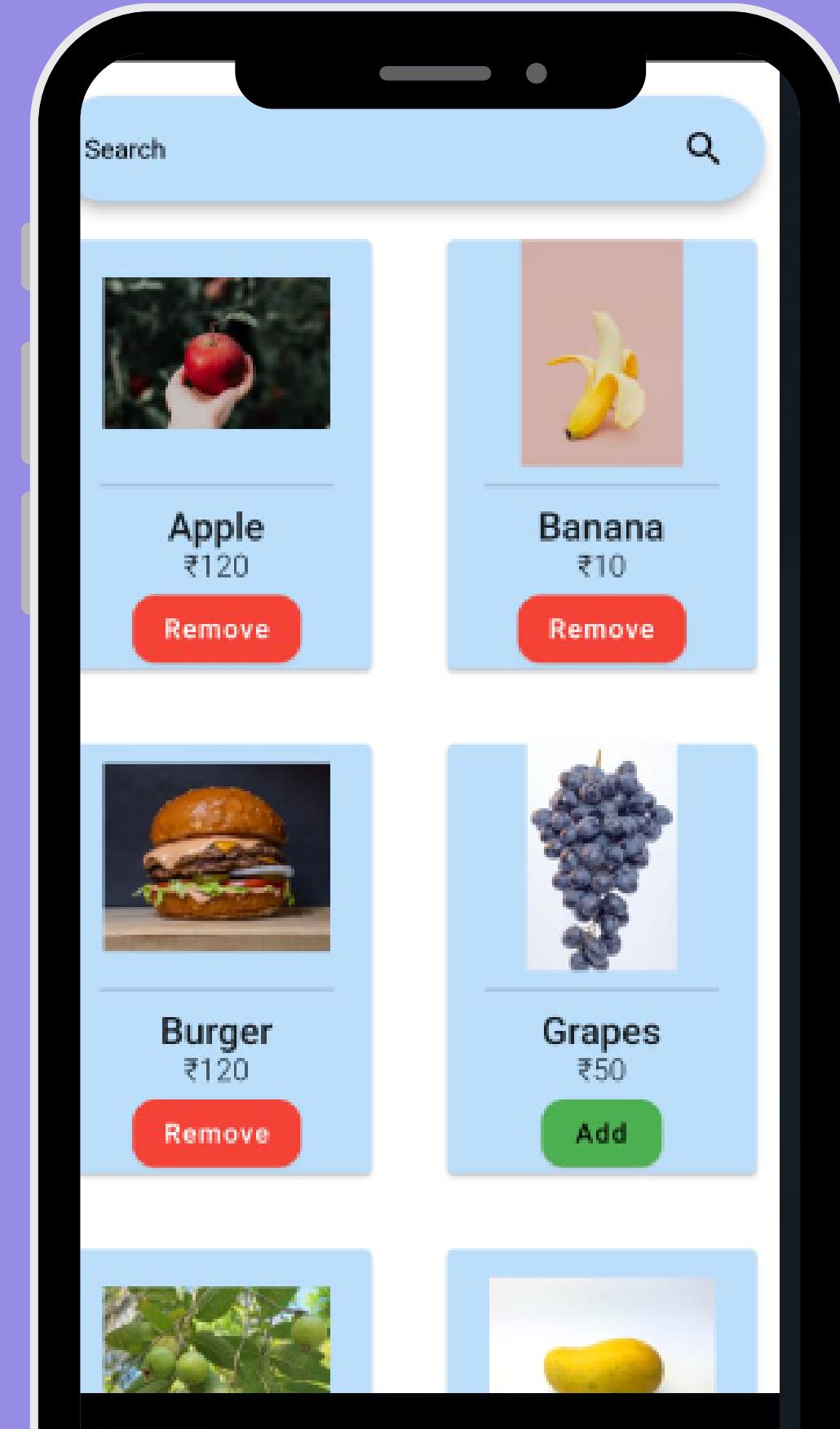
- Introduction
- Problem statement
- Literature survey
- Proposed solution
- Implementation(Flutter)
- Features
- Implementation (PWA)
- Conclusion
- Future scope
- References



# Introduction

The ultimate grocery shopping companion. With our extensive range of high-quality products and seamless user experience, grocery shopping has never been easier. Simply browse, select, and checkout, and let us take care of the rest.

"Discover fresh produce, pantry staples, and household essentials all in one place. Say goodbye to long queues and tedious shopping trips – shop smarter with [Your App Name] today!"





# Problem Statement

"Despite the convenience of online shopping, many consumers still find it challenging to efficiently purchase groceries due to limited product availability, unreliable delivery services, and cumbersome checkout processes. This presents an opportunity for a comprehensive grocery shopping solution that offers a wide range of products, reliable delivery, and a seamless user experience."





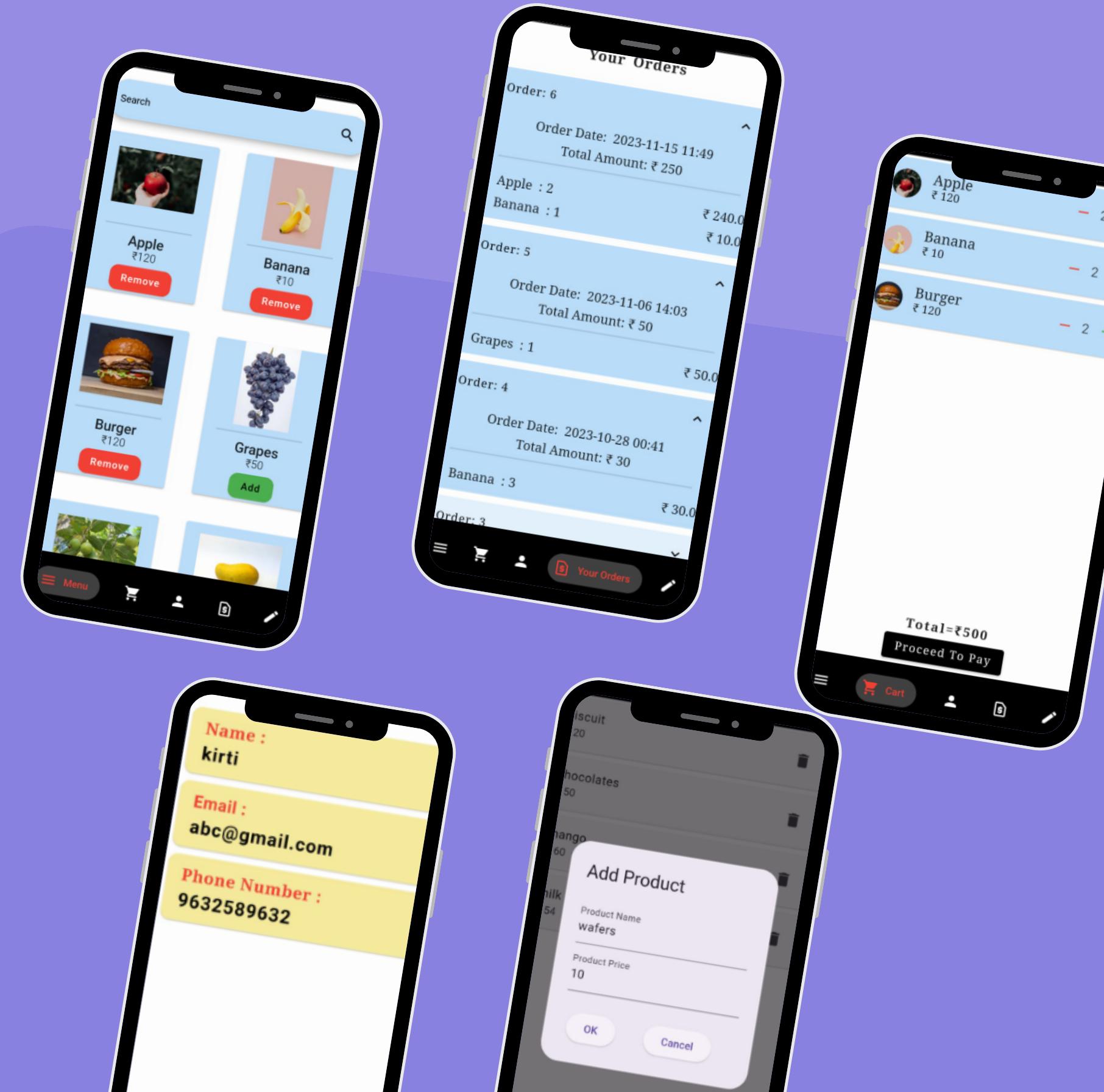
# Literature Survey

Sr. No.	Title	Published Year	Author	Objective
1.	A Comprehensive Review of Online Grocery Shopping Systems: Challenges and Opportunities	2020	John Smith, Emily Johnson	This paper aims to provide a comprehensive review of existing online grocery shopping systems, highlighting their challenges and opportunities for improvement.
2.	Enhancing User Experience in Online Grocery Shopping: A Comparative Analysis of Mobile Applications	2019	David Lee, Sarah Williams	It identifies common usability issues and proposes strategies to enhance the overall user experience, focusing on aspects such as navigation,.



# Proposed Solution

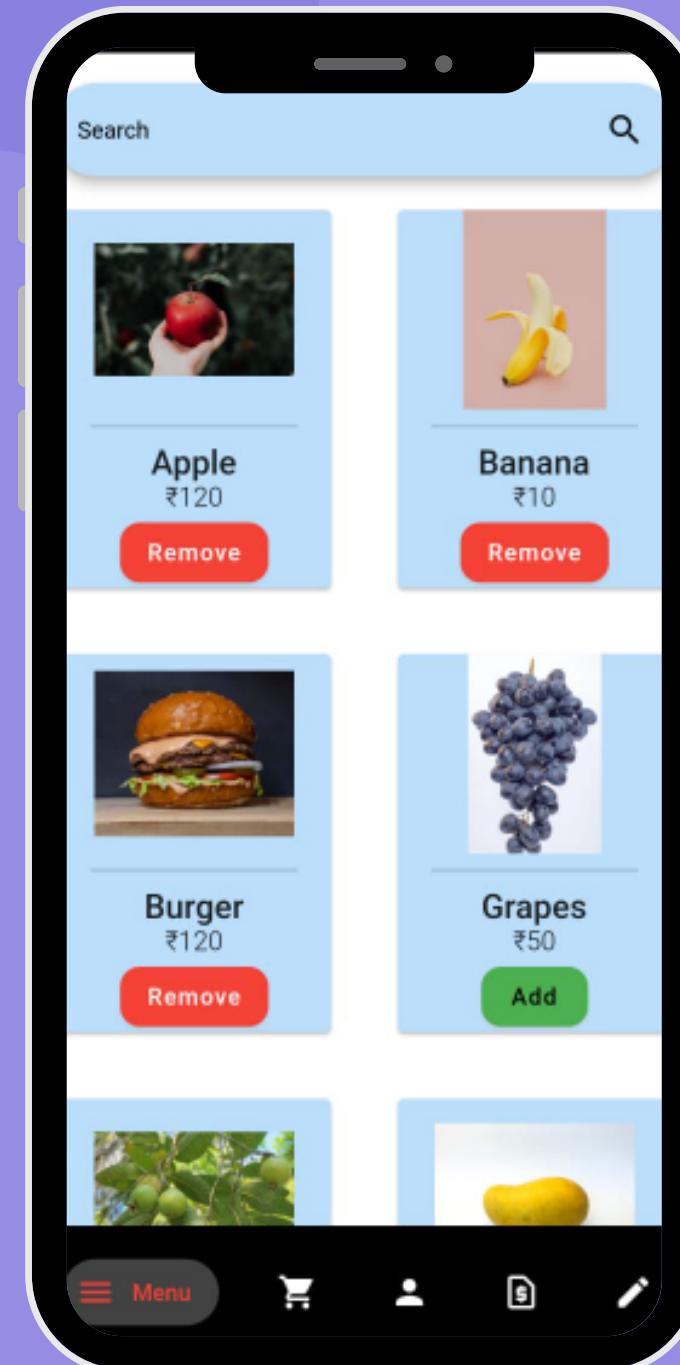
Our comprehensive grocery app streamlines the shopping experience with an intuitive interface, enabling users to explore and buy products hassle-free. Seamlessly adding items to the cart and proceeding to checkout ensures convenience at every step. The dynamic admin panel empowers real-time product updates, while robust APIs efficiently fetch images, enhancing user engagement and satisfaction.



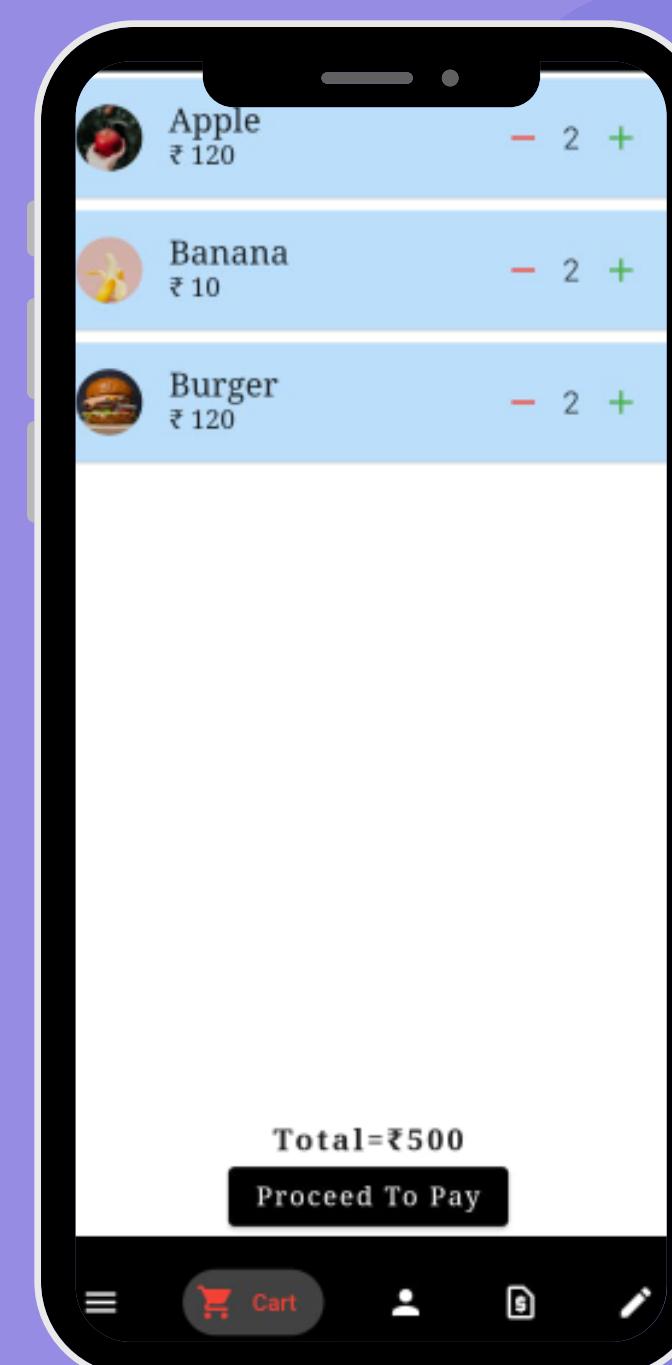


# Implementation

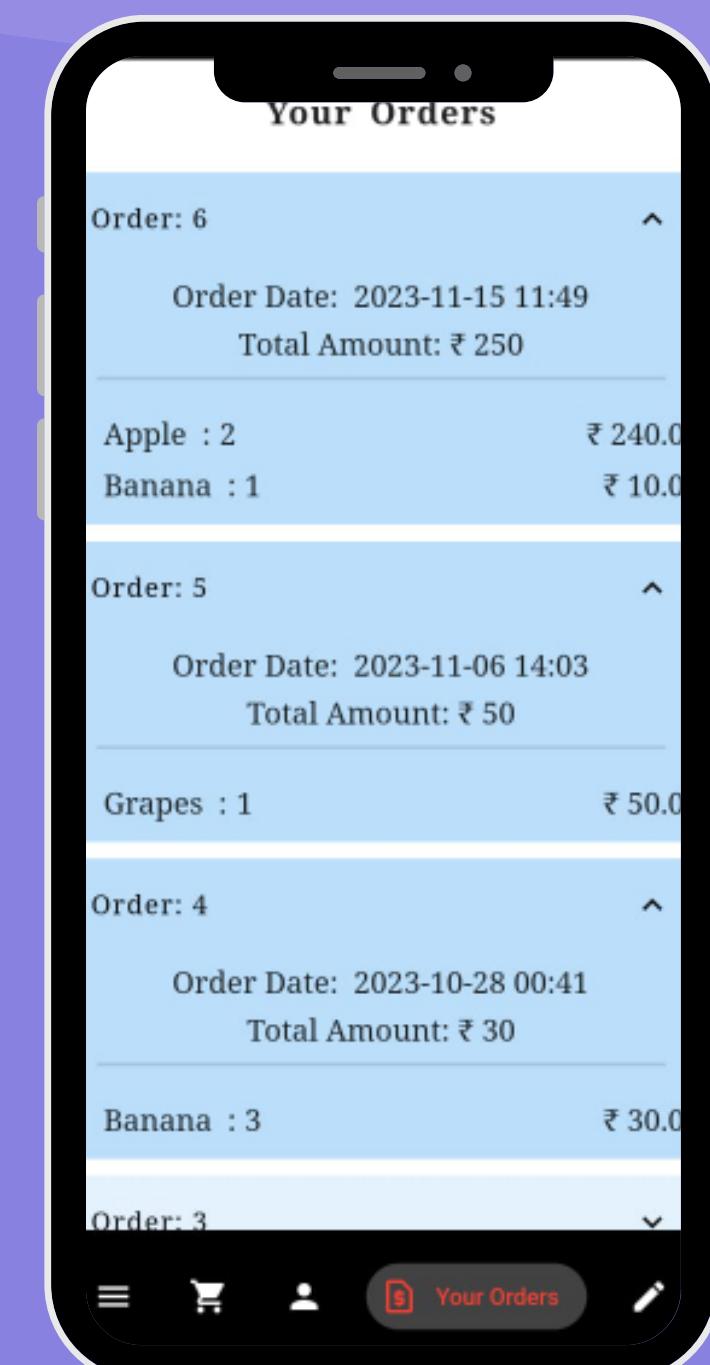
Home Page:



Cart Page:



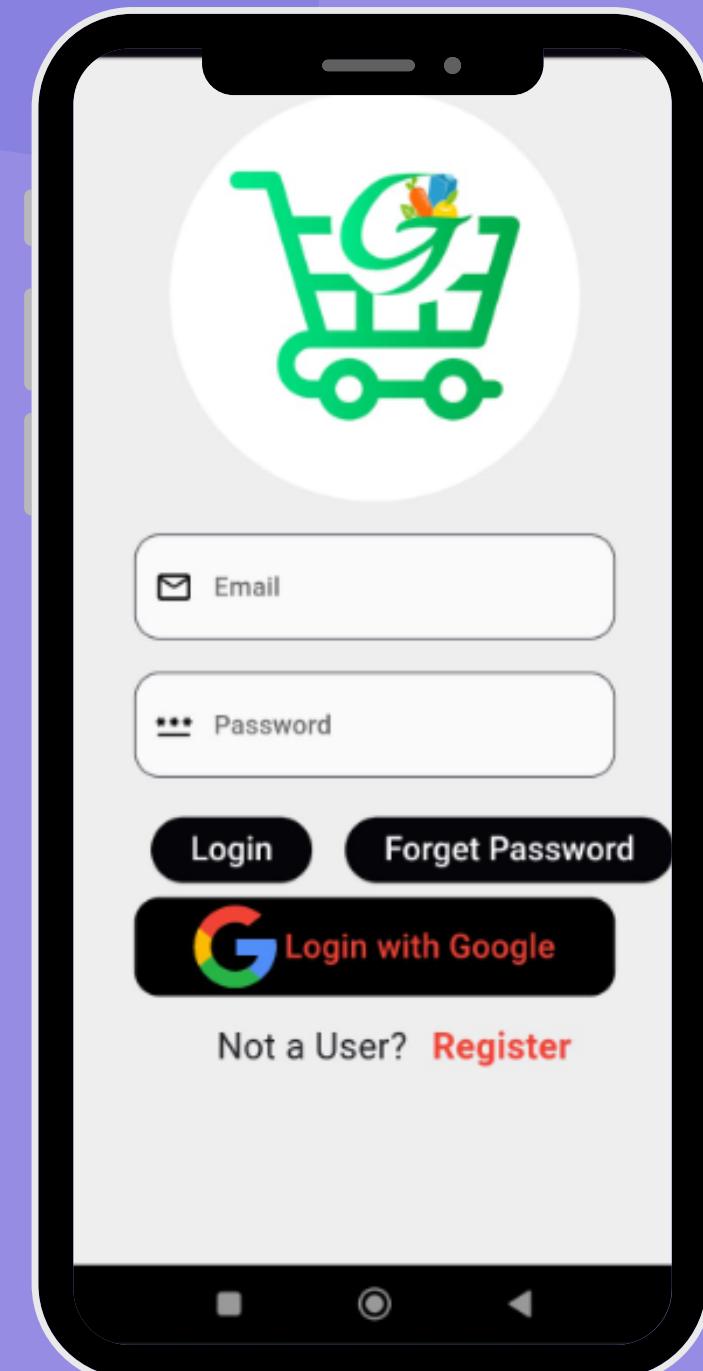
Order Page:



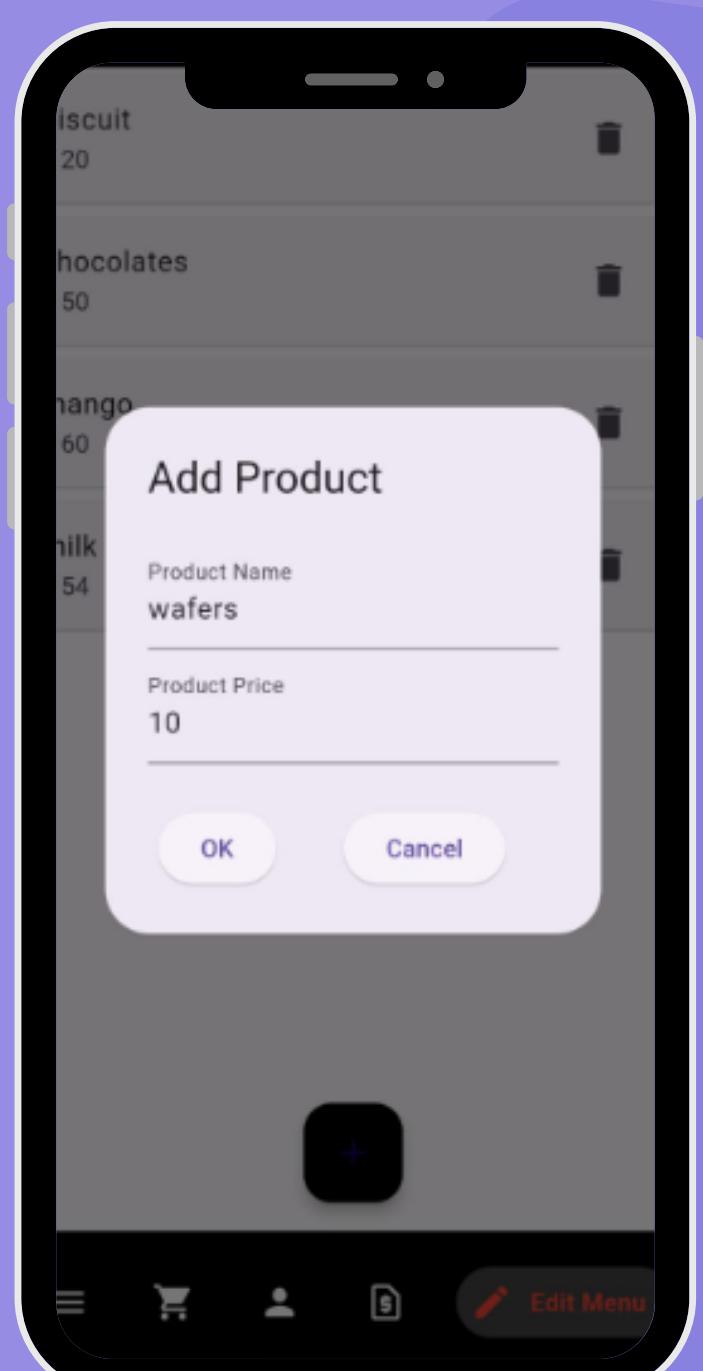


# Implementation

Login Page:



Add Product Page:



Profile Page:





# Features

1

## User-Friendly GUI

Intuitive interface for easy navigation and seamless shopping experience.

2

## Admin Panel

Enables dynamic updates to product listings and inventory management.

3

## Order History

Keeps track of previous purchases for users' convenience and reference.



4

## Image Fetching API

Retrieves product images efficiently to enhance visual appeal and engagement.



Kirti Eppanapelli

Home

About Us

Service

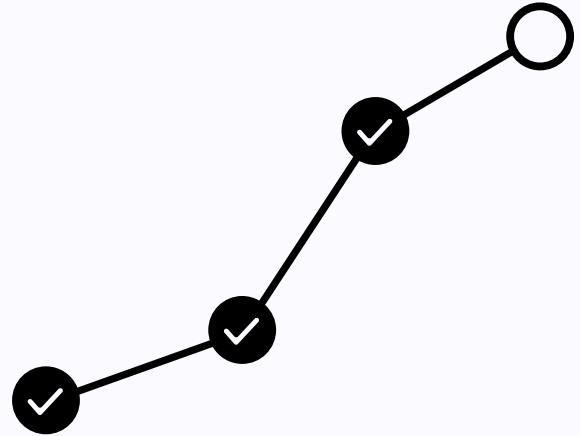
Contact



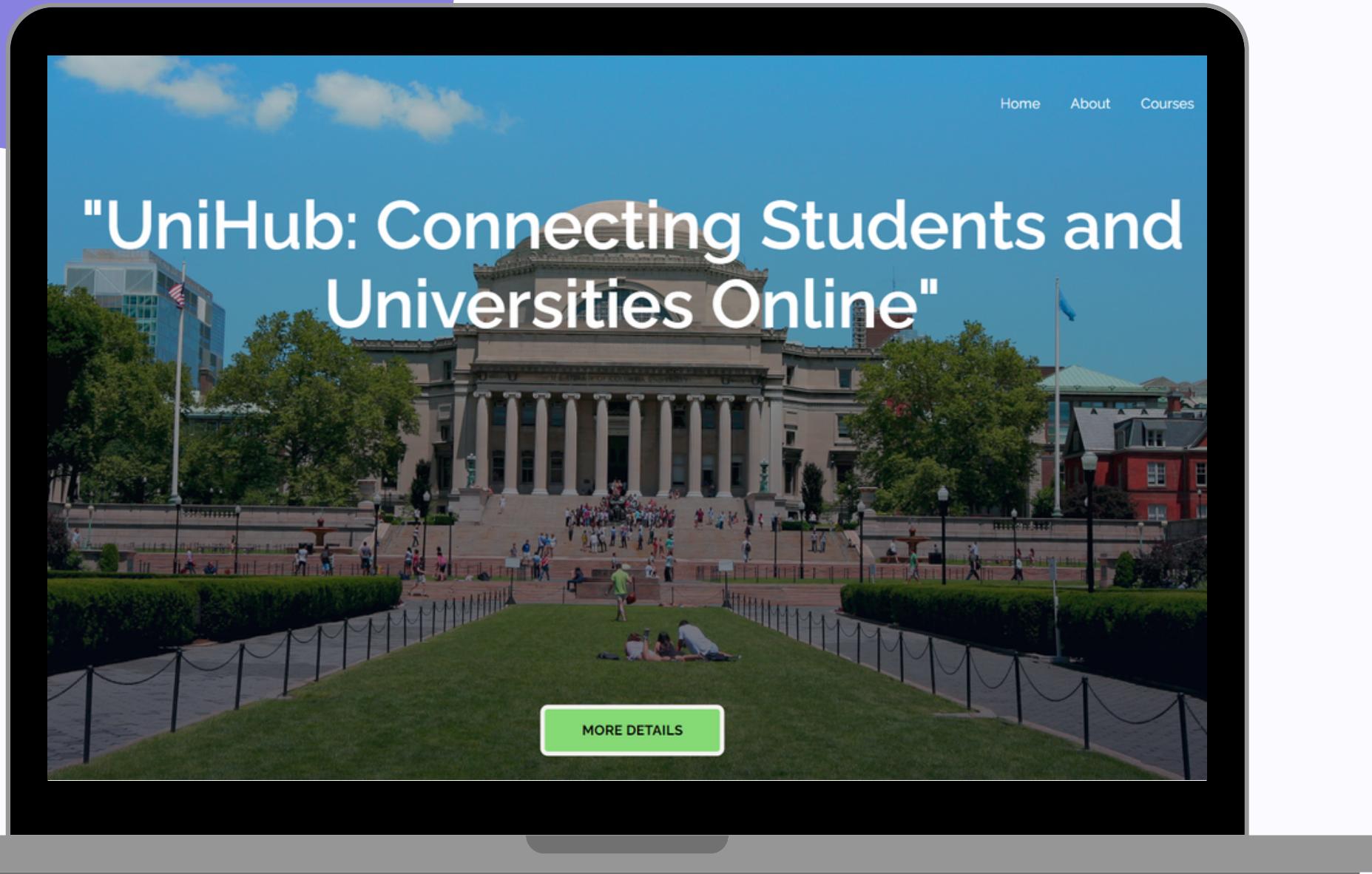
# THANKYOU



Kirti Eppanapelli



# Progressive Web App



## E-commerce Website

"Discover and enroll in diverse online courses from leading universities through UniHub. Our platform offers seamless navigation and secure transactions for an effortless learning experience. Join our community and explore endless opportunities for your educational growth."



Kirti Eppanapelli

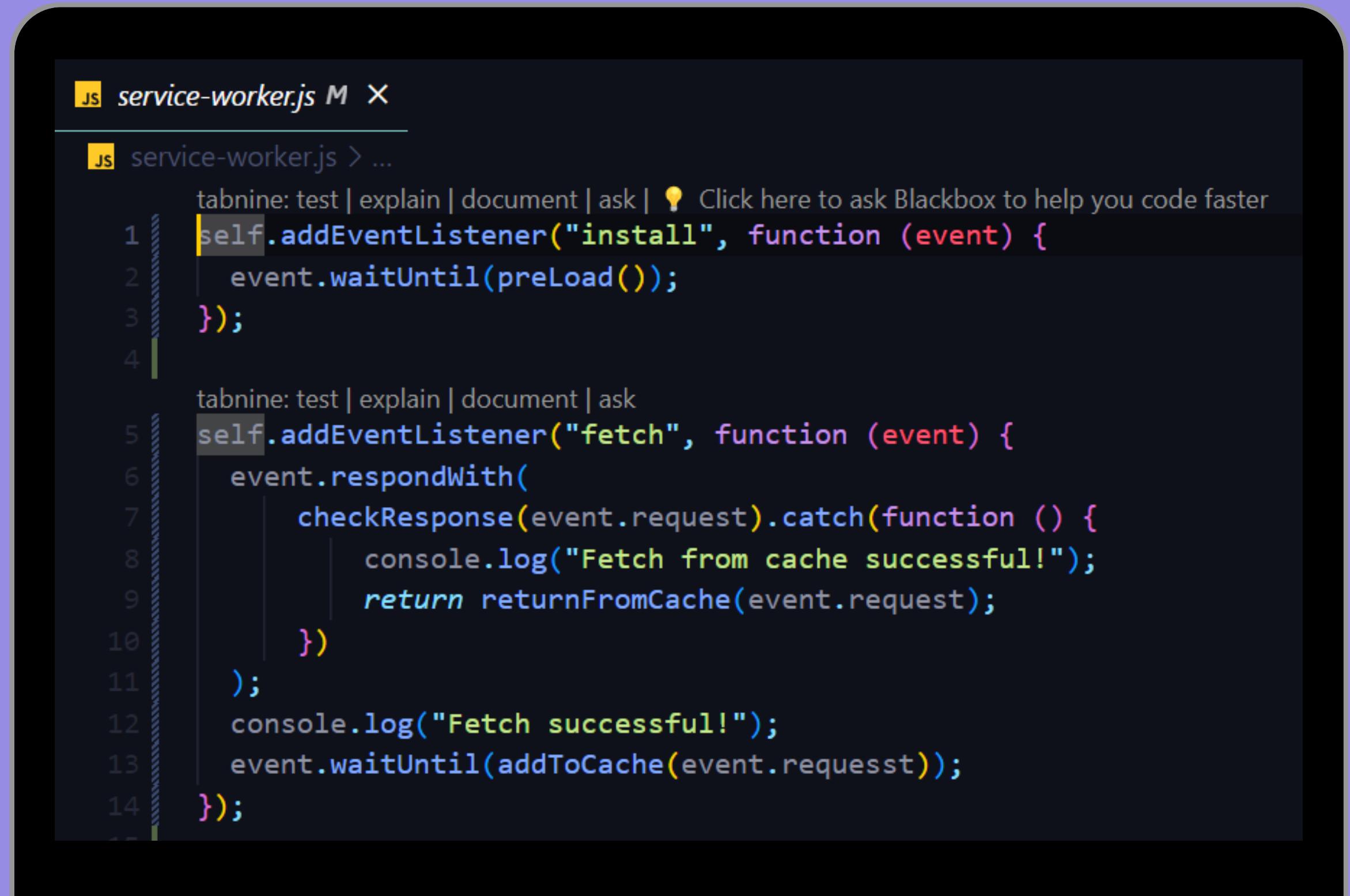
# Service Worker

The image shows a mobile device displaying a Progressive Web App (PWA) homepage. The app's header includes navigation links: Home, About, Courses, Contact Us, and a gear icon. The main content features a large image of a university campus with a prominent dome, overlaid with the text "Hub: Connecting Students and Universities Online". A green button at the bottom left says "MORE DETAILS". To the right of the device, the Chrome DevTools interface is visible, specifically the "Application" panel under the "Service workers" tab. It shows details about a service worker registered at `http://127.0.0.1:5501/`. The service worker is identified as `service-worker.js`, activated on 03/04/2024 at 16:04:23, and is currently running. The "Push" section shows a message with the payload `{"method": "pushMessage", "data": "Sync Data"}`. The "Sync" section shows a scheduled sync task named `syncData`. The "Periodic Sync" section shows a task named `test-tag-from-devtools`. The "Update Cycle" section shows three steps: Install (#2759), Wait (#2759), and Activate (#2759). The "Console" tab at the bottom of the DevTools shows log messages: "Fetch successful!", "Notification permission granted", "Fetch successful!", and "Service Worker registered with scope: http://127.0.0.1:5501/".

A service worker is a script that runs in the background of a Progressive Web App (PWA), separate from the main web page. It enables features like offline functionality, push notifications, and background synchronization. Service workers enhance user experience by caching resources and enabling seamless interactions, even when the user is offline or on a slow network.



# Serviceworker.js



A smartphone icon is positioned at the bottom of the slide, containing a dark blue screen with a white code editor interface. The code editor shows a file named 'service-worker.js' with the following content:

```
js service-worker.js M X
js service-worker.js > ...
tabnine: test | explain | document | ask |💡 Click here to ask Blackbox to help you code faster
1 self.addEventListener("install", function (event) {
2   event.waitUntil(preLoad());
3 });
4
5 tabnine: test | explain | document | ask
6 self.addEventListener("fetch", function (event) {
7   event.respondWith(
8     checkResponse(event.request).catch(function () {
9       console.log("Fetch from cache successful!");
10      return returnFromCache(event.request);
11    })
12  );
13  console.log("Fetch successful!");
14  event.waitUntil(addToCache(event.request));
15});
```

`serviceworker.js` is a key script in PWAs, enabling offline functionality and improving performance by caching resources and intercepting network requests in the background.



# Push Notification

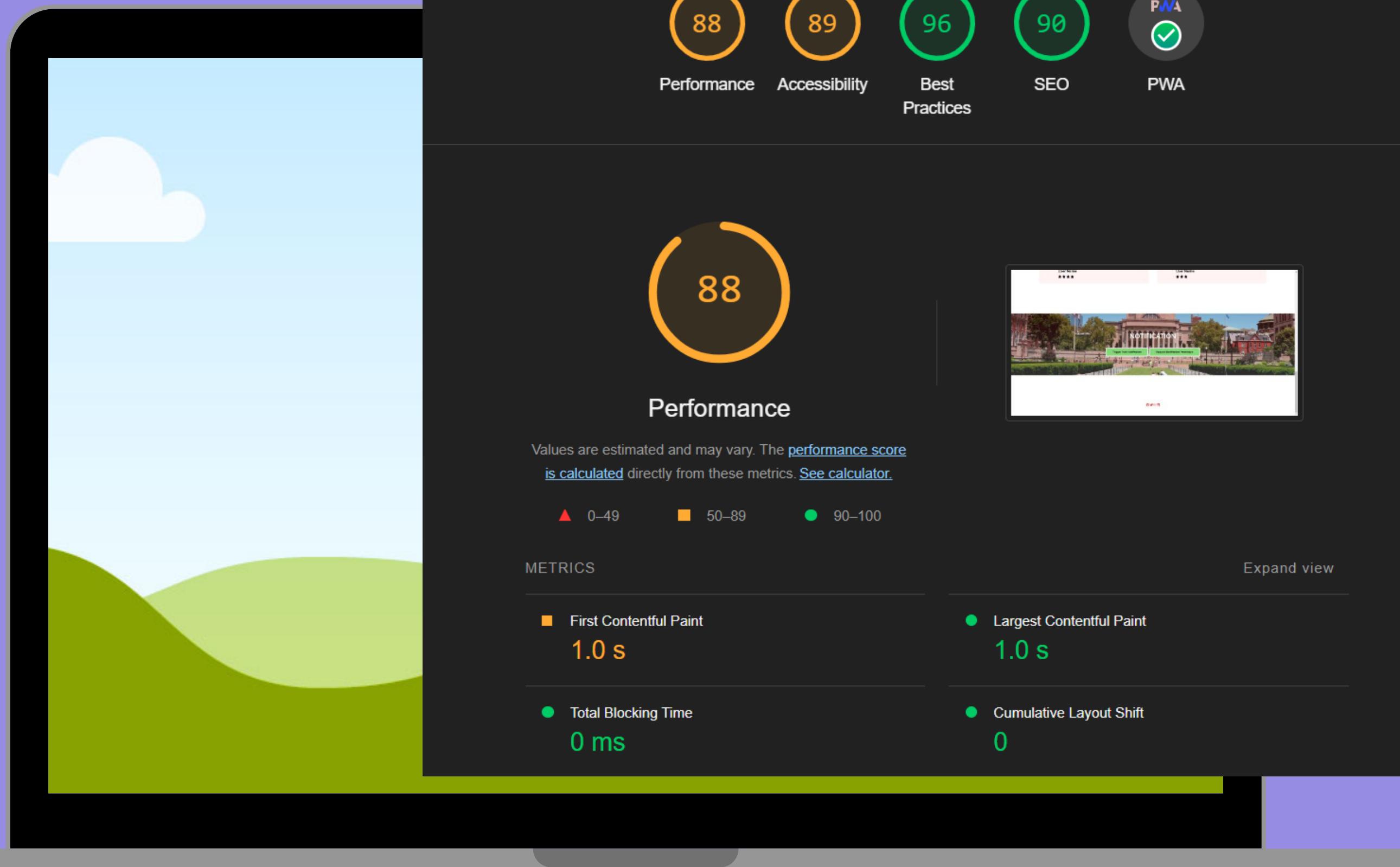
The screenshot shows the Chrome DevTools Application tab for a service worker at `http://127.0.0.1:5501/`. The left sidebar lists sections like Application, Storage, and Background services. In the main area, under the Application section, there's a 'Push' configuration panel with the message `{"method": "pushMessage", "message": "kirti here"}` and a 'Push' button. Below it is a 'Sync' panel with the message `syncData` and a 'Sync' button. Under 'Periodic Sync', the tag `test-tag-from-devtools` is listed with a 'Periodic Sync' button. The 'Update Cycle' section shows three entries: 'Install' (version #2759), 'Wait' (version #2759), and 'Activate' (version #2759, currently active, indicated by a yellow progress bar). At the bottom, the console log shows:

```
15 Fetch successful!
Notification permission granted
2 Fetch successful!
Service Worker registered with scope: http://127.0.0.1:5501/
Service Worker registered with scope: http://127.0.0.1:5501/
```

Push notifications in PWAs are messages sent to users' devices from servers, even when the app isn't open. They enhance user engagement by delivering updates or alerts, facilitated by service workers for seamless integration.



Kirti Eppanapelli



# Light House

Light House is an open-source tool by Google that audits and provides feedback on web app performance, accessibility, and best practices, helping developers optimize their websites for better user experiences and search engine rankings.

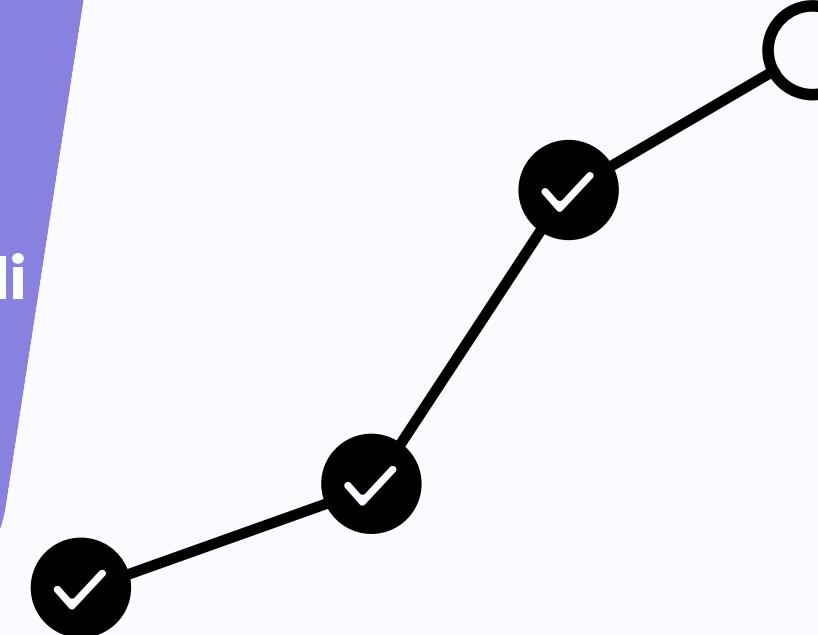


Kirti Eppanapelli

# Conclusion

In summary, our grocery app offers a superior shopping experience with its intuitive interface, dynamic admin features, comprehensive order history, and efficient image fetching API. Additionally, our implementation of service worker.js and push notifications enhances user engagement, while utilizing Lighthouse ensures top-notch website performance and user satisfaction.





# Future Scope

- Implement AI algorithms to analyze user behavior and provide personalized product recommendations, enhancing user engagement and satisfaction.
- Integrate advanced analytics tools to gather insights from user interactions and improve decision-making for inventory management and marketing strategies.
- Expand the product range to include diverse categories and brands, catering to a wider audience and increasing customer retention.
- Enhance user experience by adding social features such as product reviews, ratings, and sharing options, fostering a sense of community and trust among users.



# References

- 1. Usman, M., Iqbal, F., & Hassan, S. A. (2018, December). A Survey on Mobile Application Development Methodologies. In 2018 International Conference on Computing, Engineering and Technology (ICCET) (pp. 1–6). IEEE [IEEE xplore paper A Survey on Mobile Application Development Methodologies].
- 2. Alves, V., Hassan, A. F., & Correia, A. M. (2018, July). A Model-Driven Approach for Building Cross-Platform Mobile Applications. In 2018 42nd International Conference on Software Engineering (ICSE) (pp. 1222–1233). IEEE [IEEE xplore paper A Model-Driven Approach for Building Cross-Platform Mobile Applications].

# THANKYOU

