Thursday, 4th April, 2024

Information Technology,
VESIT

ADVISOR

Dr. Ravita Mishra

STUDENT

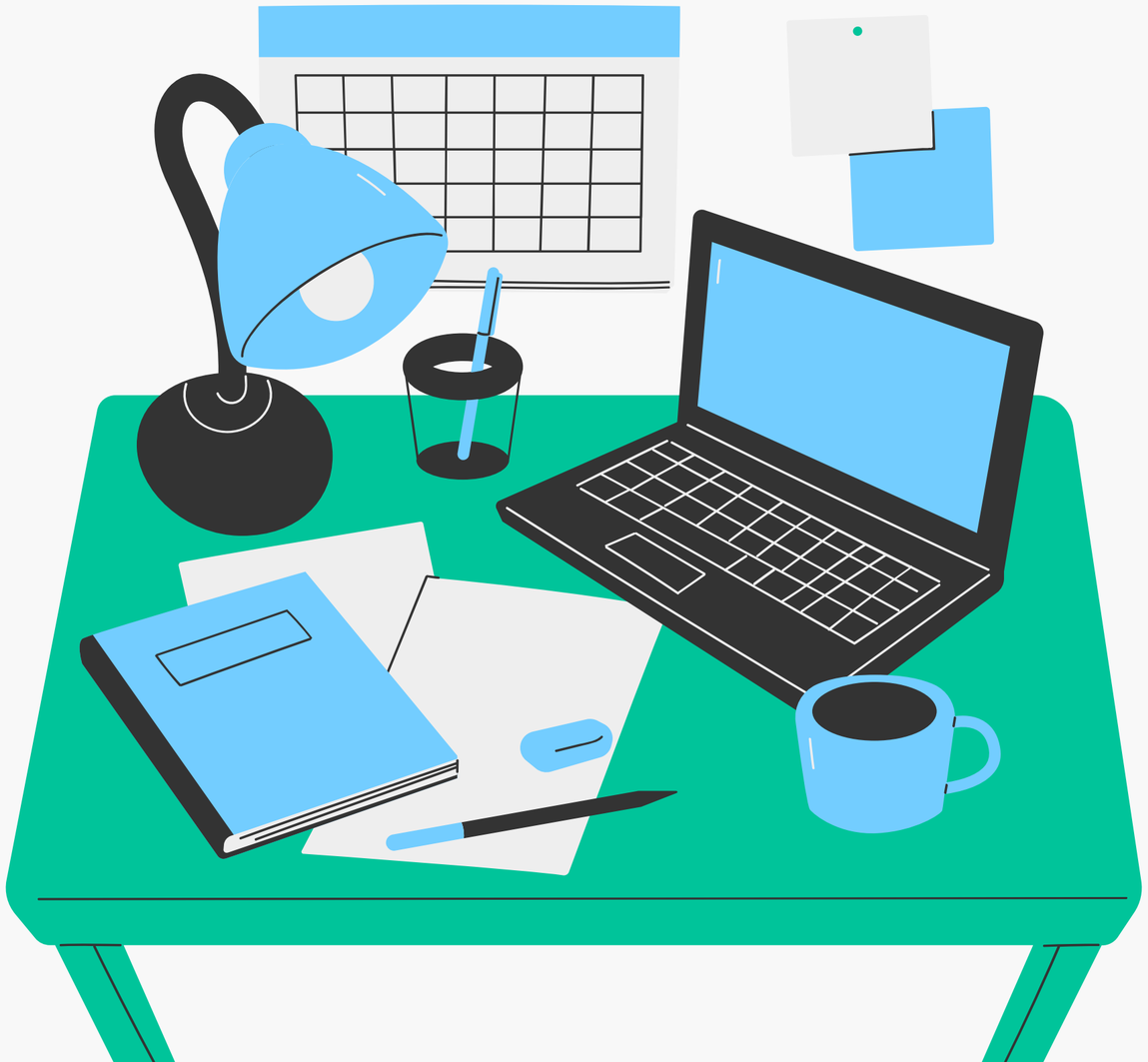
Nishtha Joshi
D15B 25

Flutter Presentation

Table of Contents

I Introduction and Motivation

- Students need efficient expense management to balance finances and academics effectively.
- Our Flutter Expense Tracker project aims to provide a user-friendly solution tailored to student needs.
- Student expenses range from tuition to leisure activities, demanding a streamlined tracking approach.
- Conventional methods are often cumbersome for busy students.
- Our project's motivation lies in offering students a convenient tool for monitoring spending habits and fostering financial literacy.



II Objectives

1. Expense Categorization:
- Allow users to categorize expenses for a clear overview.
- Customize categories to fit individual preferences.

2. Spending Visualization:
- Display spending patterns over daily, weekly, monthly, and yearly periods.
- Enable interactive graphs for detailed analysis.

3. Top Spending Insights:
- Provide insights into top spending categories.
- Allow viewing of top spendings across different time frames.

4. User-Friendly Design:
- Design an intuitive interface for easy navigation.
- Incorporate user feedback for continuous improvement.

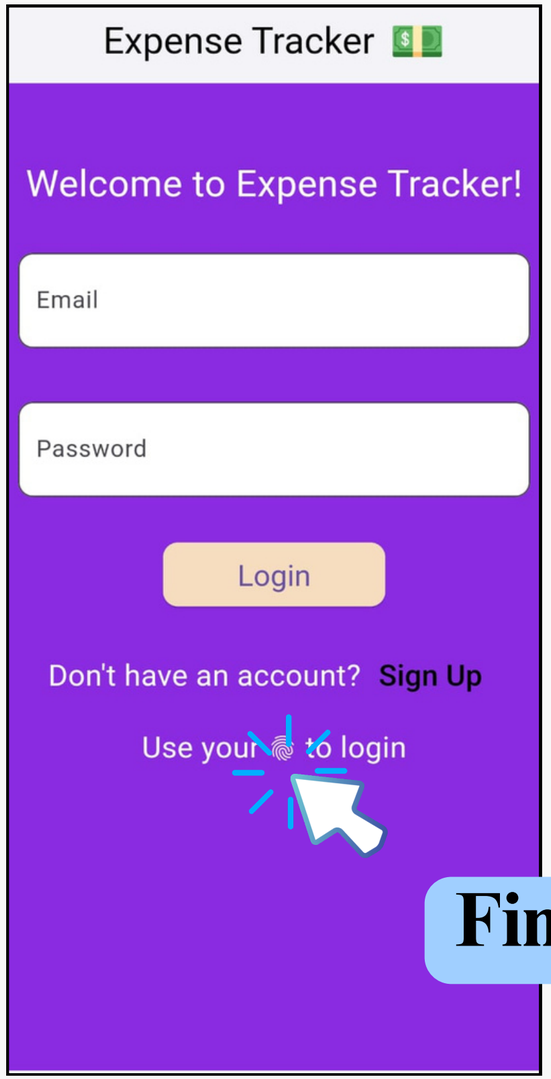
III Hardware and Software Requirements

- **Hardware:**
- Device: Compatible with Android and iOS devices (smartphones and tablets).
- Processor: Standard processors found in modern smartphones and tablets.
- Memory: Recommended minimum RAM of 2GB for smooth performance.
- Storage: Adequate storage space for the application installation and data storage.

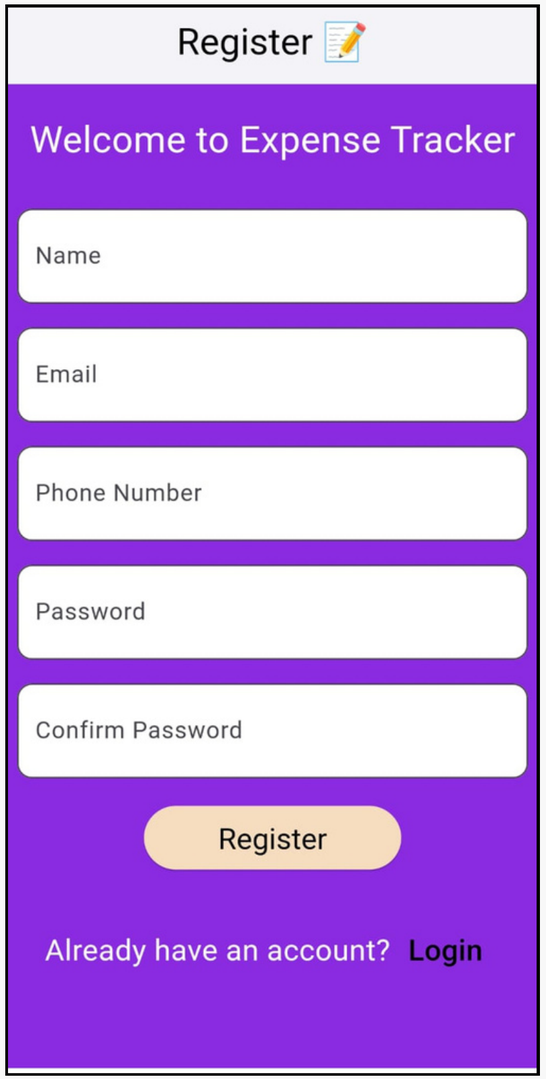
- **Software**
- Operating System: Android 5.0 (Lollipop) or later for Android devices, iOS 10.0 or later for iOS devices.
- Development Platform: Latest stable versions of Flutter SDK and Dart SDK.
- Integrated Development Environment (IDE): Visual Studio Code or Android Studio with Flutter and Dart plugins.
- Firebase Integration: Firebase Authentication SDK for Dart, configured Firebase project with Authentication enabled.

IV Implementation

Login and Register Page

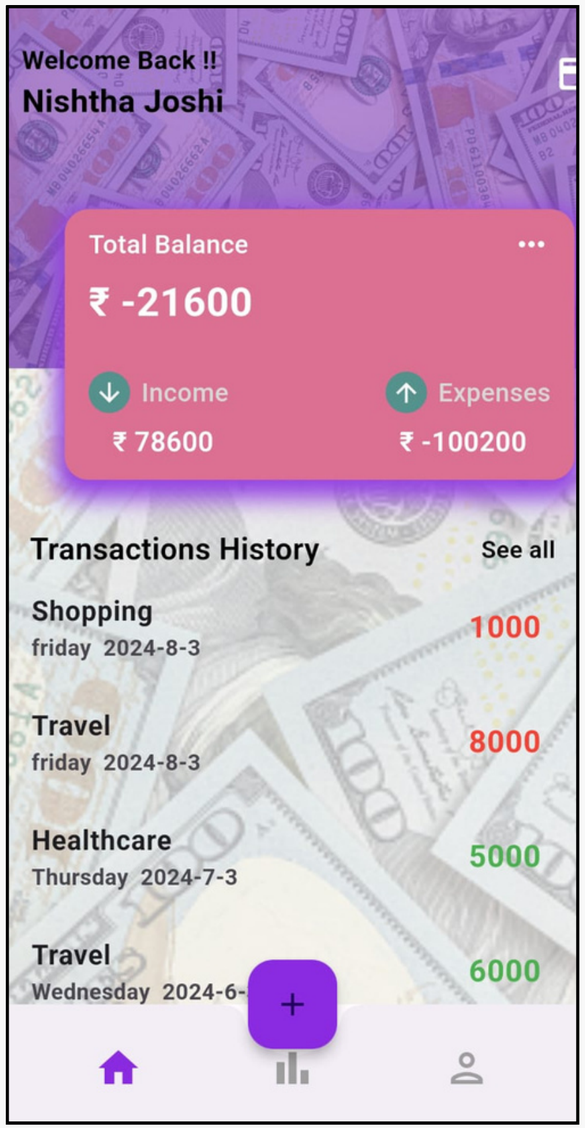


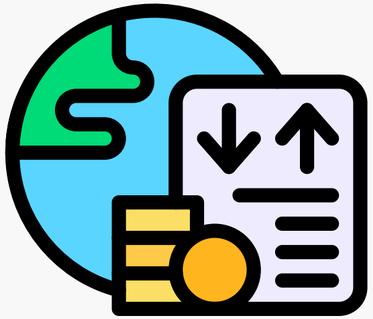
Fingerprint sensor



IV Implementation

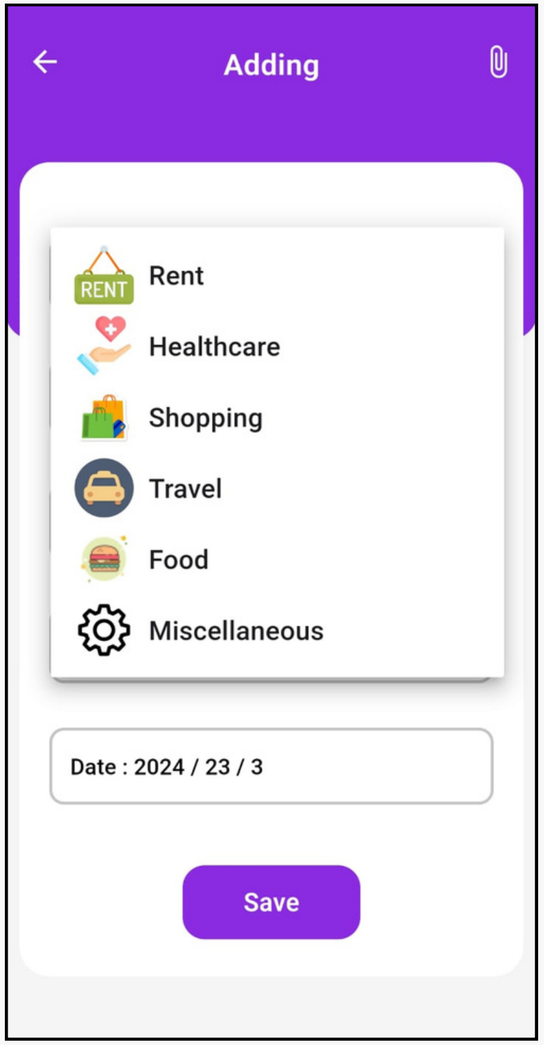
Home Page





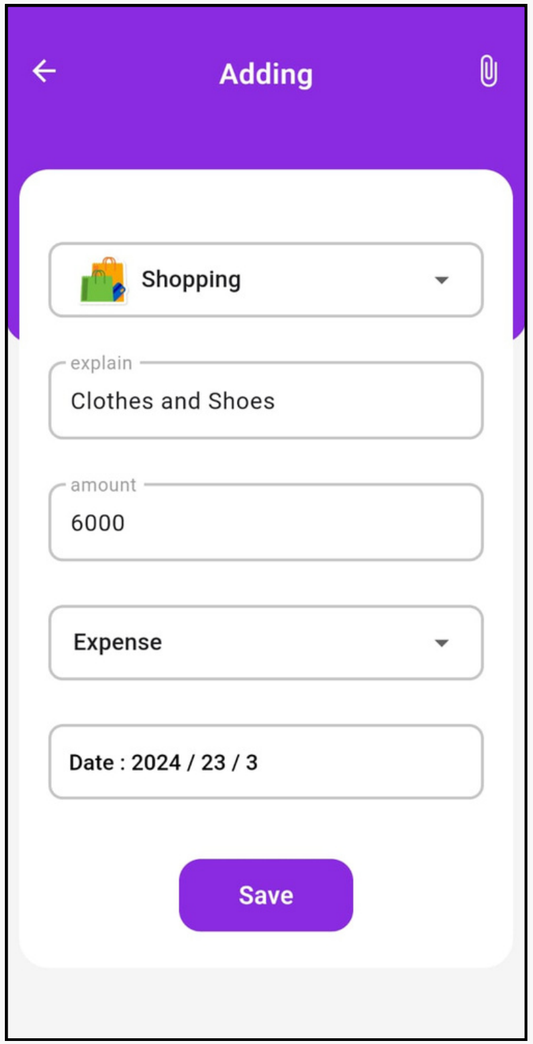
IV Implementation

Spending / Income Categories

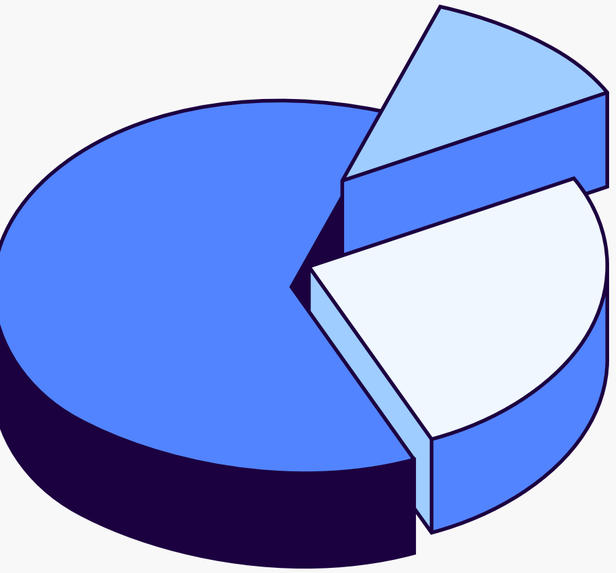


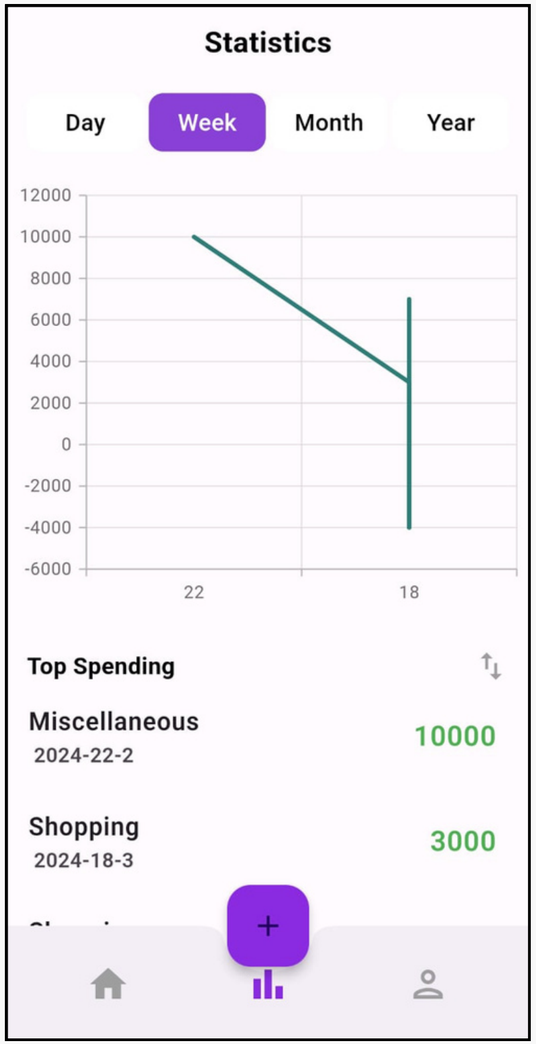
IV Implementation

Spending / Income Categories

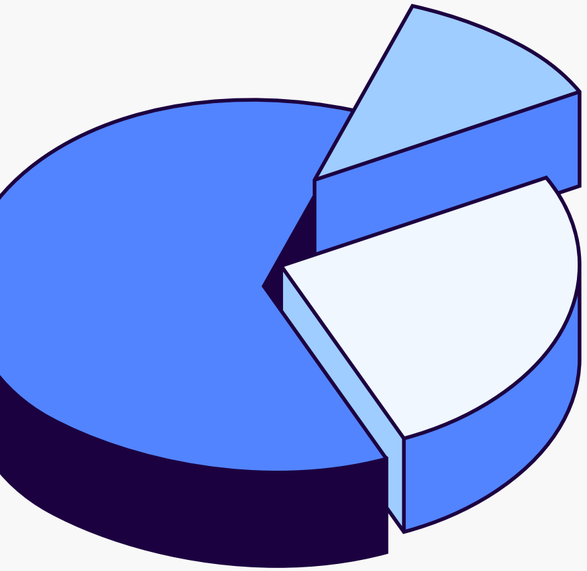


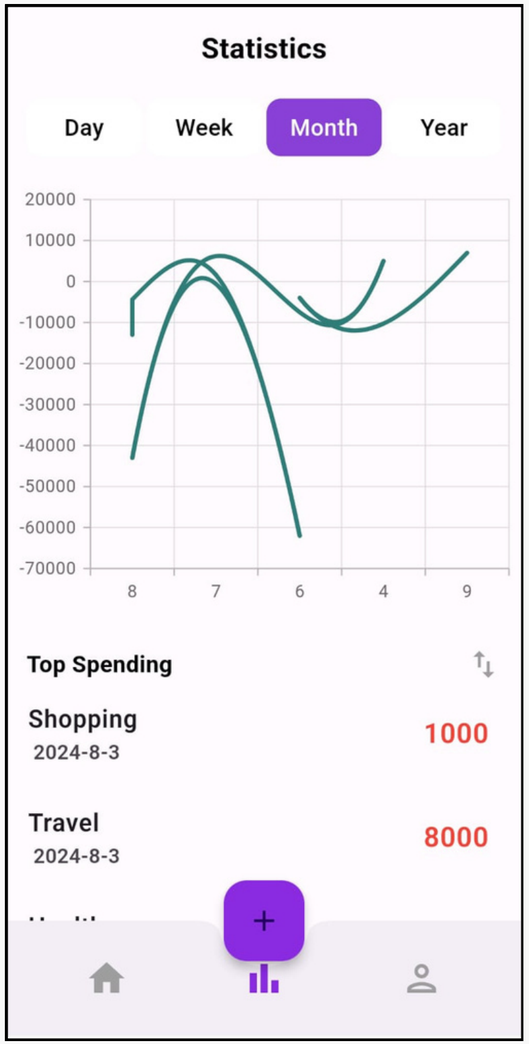
IV Statistics



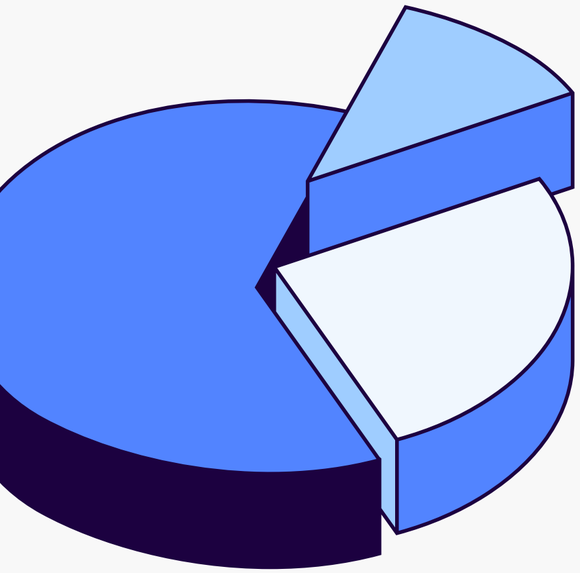


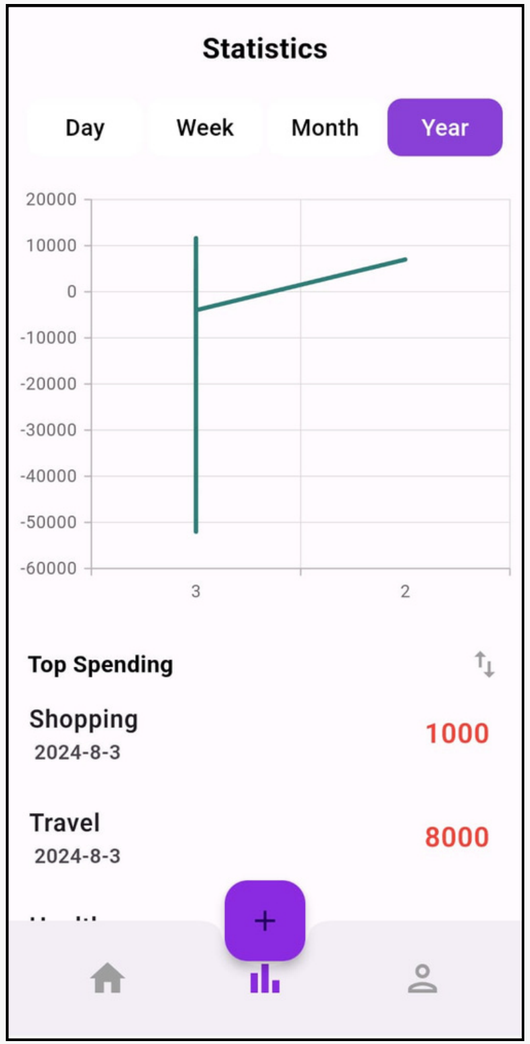
IV Statistics



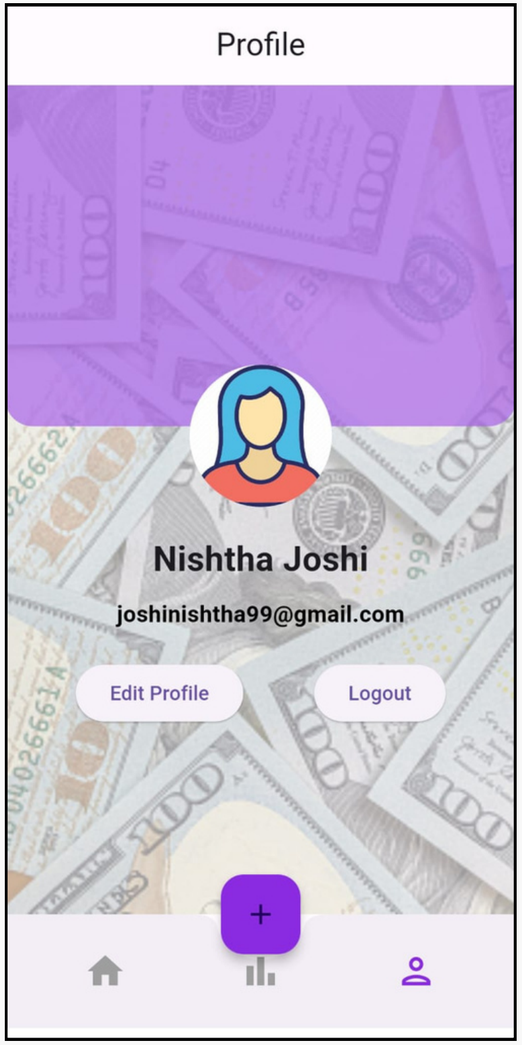


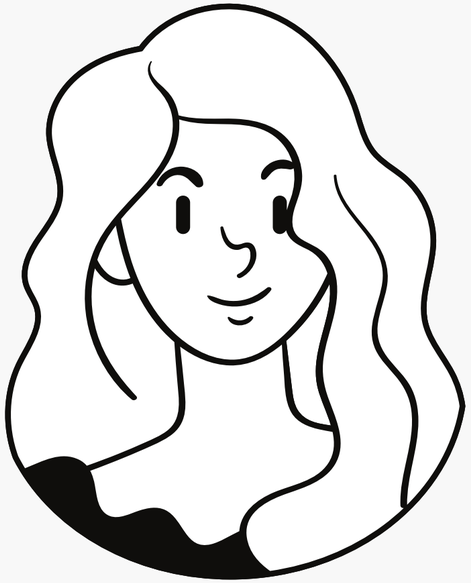
IV Statistics

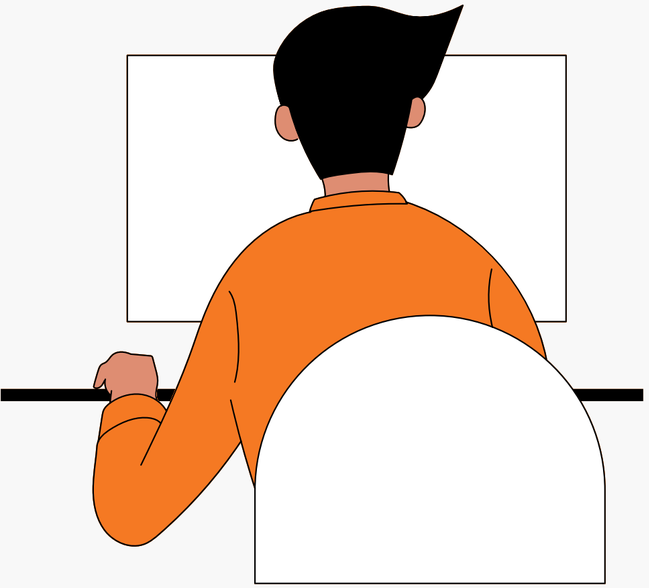




IV Profile Page







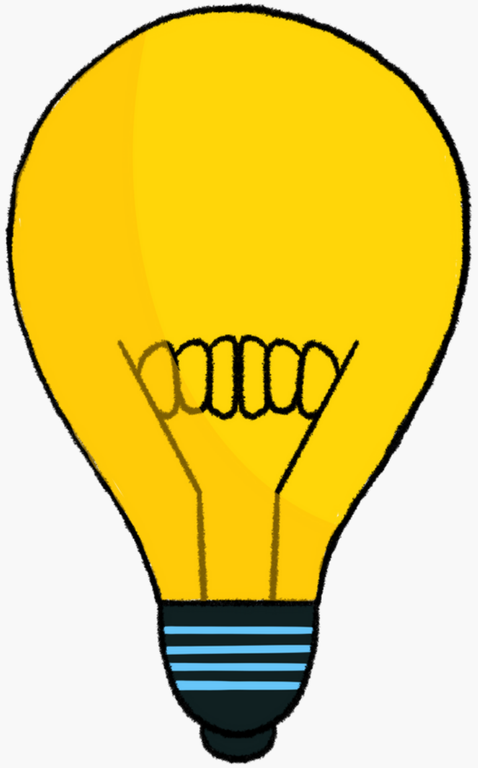
V Conclusion

1. Our Flutter Expense Tracker project offers a tailored solution for students to manage their finances effectively, addressing the need for simplified expense tracking in a busy academic environment.

2. By implementing features such as expense categorization, spending visualization, and top spending insights, our application empowers users with valuable tools for budgeting and financial awareness.

3. Leveraging the Dart programming language and Firebase authentication, we've ensured a seamless and secure user experience, enabling students to trust our application with their sensitive financial data.

4. As we conclude this project, we envision our Expense Tracker not just as a tool for managing expenses, but as a catalyst for promoting financial literacy and responsibility among students, fostering lifelong financial habits that extend beyond their academic years.



VI References

eExpense: A Smart Approach to Track Everyday Expense

S. A. Sabab, S. S. Islam, M. J. Rana and M. Hossain, "eExpense: A Smart Approach to Track Everyday Expense," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 2018, pp. 136-141, doi: 10.1109/CEEICT.2018.8628070. keywords: {Optical character recognition software;History;Smart phones;Character recognition;Calculators;Data mining;Cameras;smartphone;expenditure;track expense;OCR;SMS tracking},

Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart

Thomas Bailey; Alessandro Biessek; Trevor Wills, Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter 2.5 and Dart , Packt Publishing, 2021.

Wednesday, 4th March

Information Technology,
VESIT

ADVISOR

Dr. Ravita Mishra

STUDENT

Nishtha Josh
D15B 25

Thank you for listening!

E-Commerce PWA

Nishtha Joshi

2024 April 4

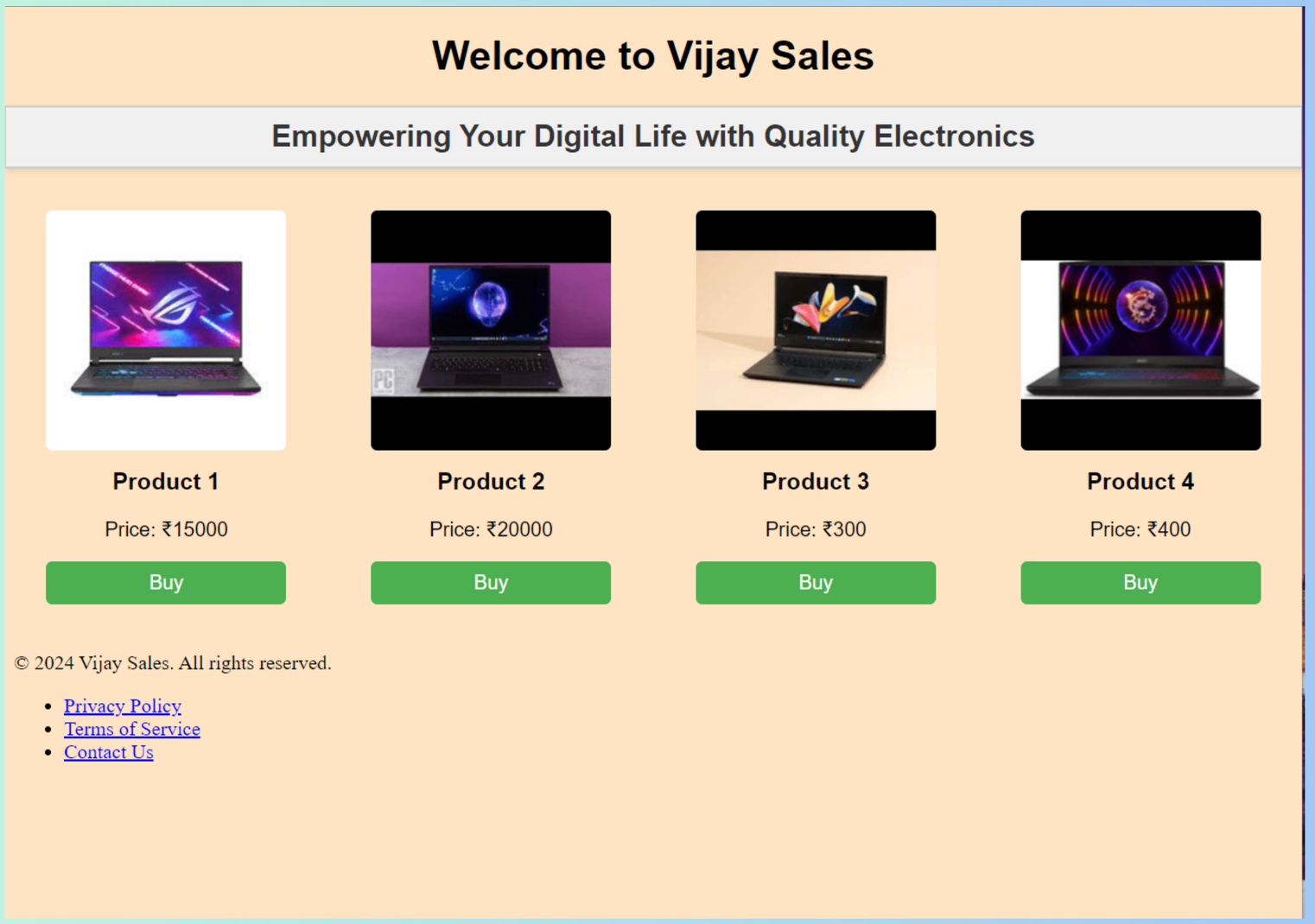
What is PWA?

PWA stands for Progressive Web App. It's a type of application software delivered through the web, built using common web technologies like HTML, CSS, and JavaScript. PWAs are designed to work on any platform that uses a standards-compliant browser, including both desktop and mobile devices.

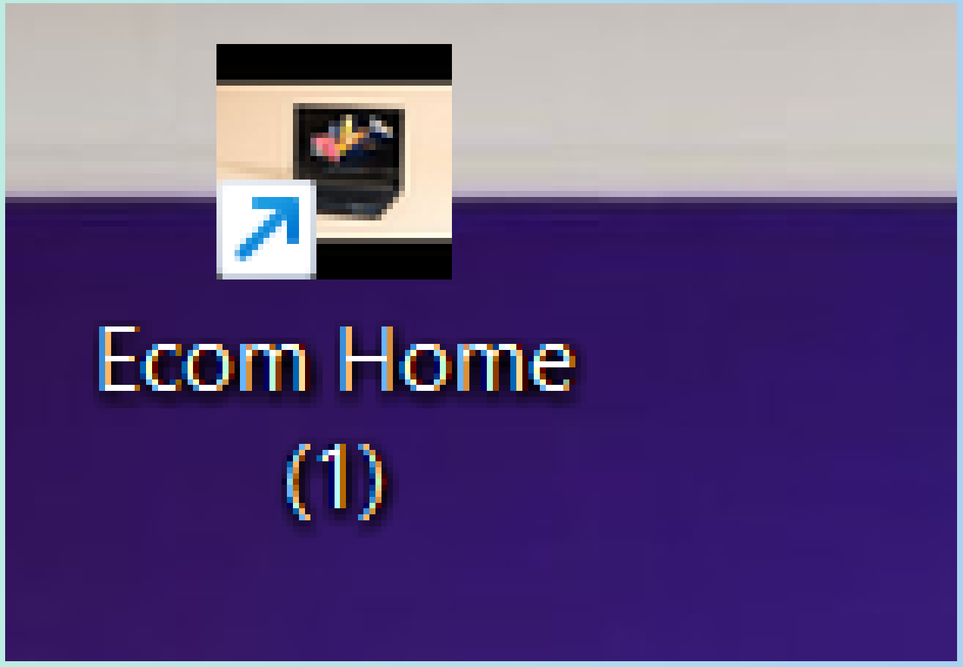
Installation of PWA

Installing a Progressive Web App (PWA) is crucial as it ensures the website's availability regardless of network connectivity, offering users instant access anytime, anywhere. By allowing installation, PWAs integrate seamlessly into users' devices, fostering increased engagement and retention rates. Additionally, PWAs boast optimized performance, delivering fast loading times and responsive navigation, thereby enhancing user satisfaction. Moreover, the simplified distribution process of PWAs eliminates the need for app store approval, enabling developers to reach a wider audience more efficiently. Overall, installing a PWA enhances accessibility, improves user engagement, streamlines distribution, and contributes to a seamless user experience.

Installation of PWA

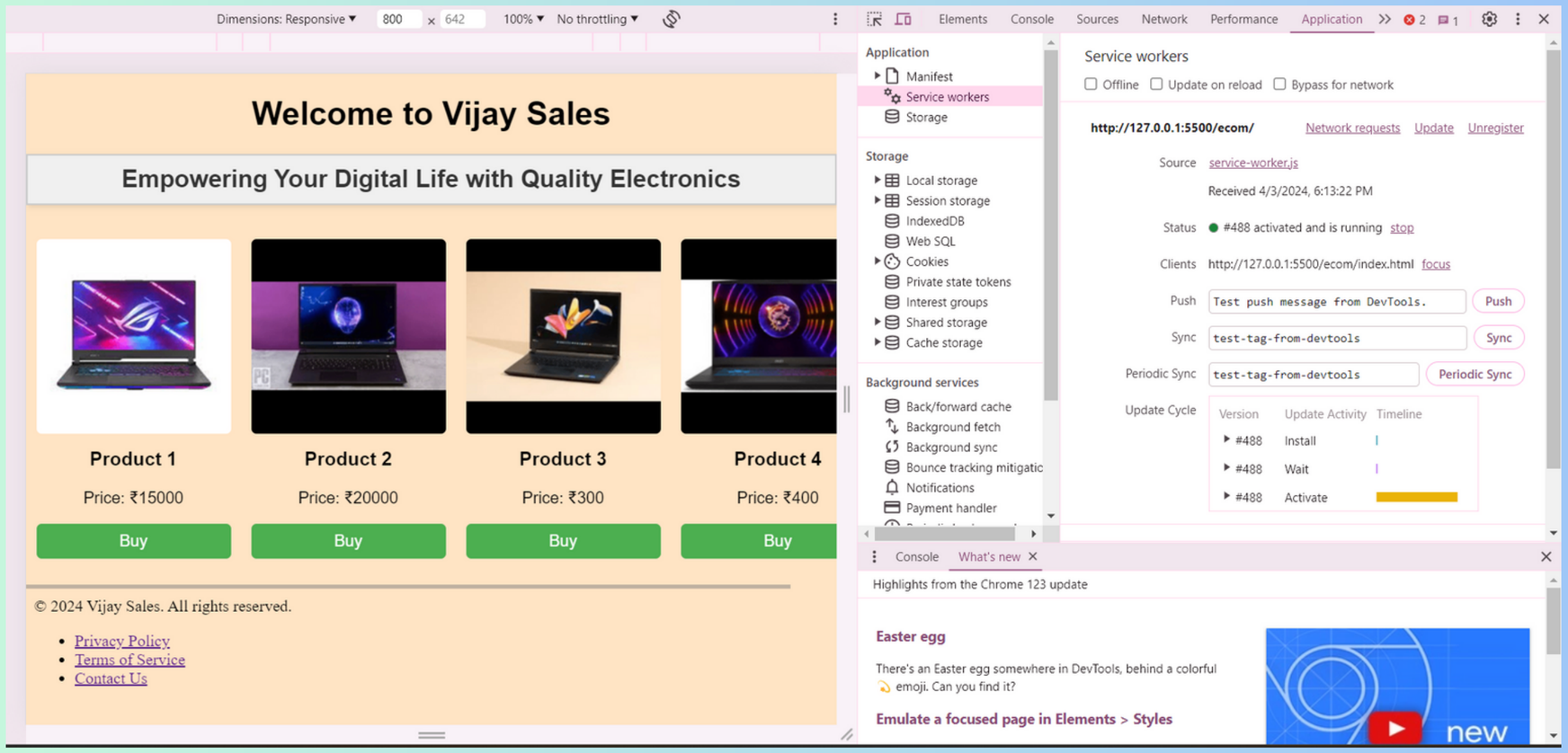


Installation of PWA



SERVICE WORKERS IN PWA

Service Worker is a script that works on browser background without user interaction independently. Also, It resembles a proxy that works on the user side. With this script, you can track network traffic of the page, manage push notifications and develop “offline first” web applications with Cache API.



Implementation of Events

In Progressive Web Apps (PWAs), fetch, push, and sync events play crucial roles in enhancing functionality and user experience.

The fetch event allows PWAs to intercept network requests, enabling them to cache resources for offline use, reduce load times, and improve performance.

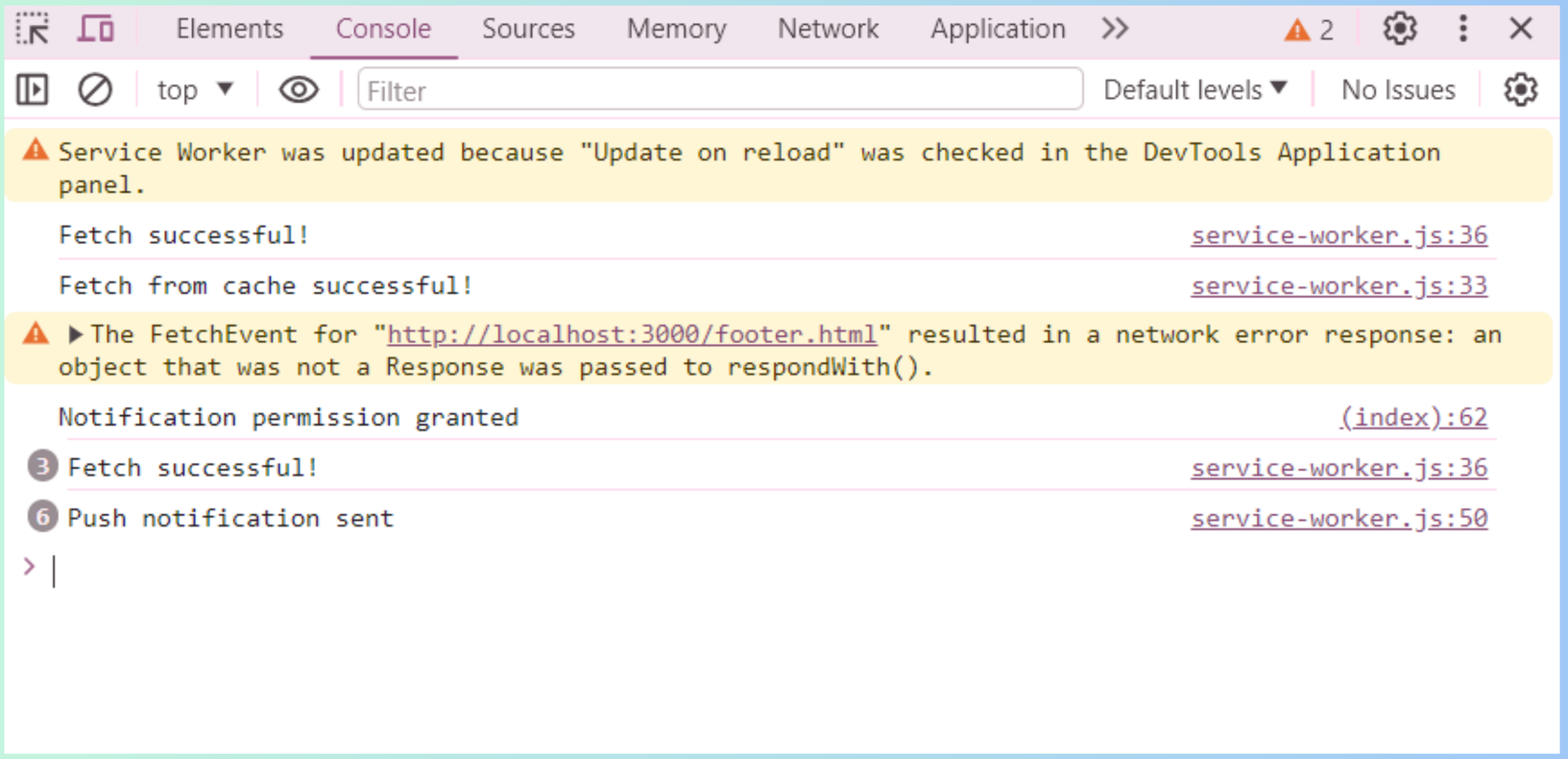
Push events empower PWAs to send notifications to users even when the app is not actively running, facilitating real-time updates and engagement.

Sync events enable background synchronization of data, ensuring that the app remains up-to-date with the latest information, regardless of network availability.

Together, these events contribute to the seamless operation and enhanced functionality of PWAs, enriching the user experience across various devices and network conditions.

PUSH EVENT

Usiing Application Tab from Chrome Developer Tools for testing push notification.



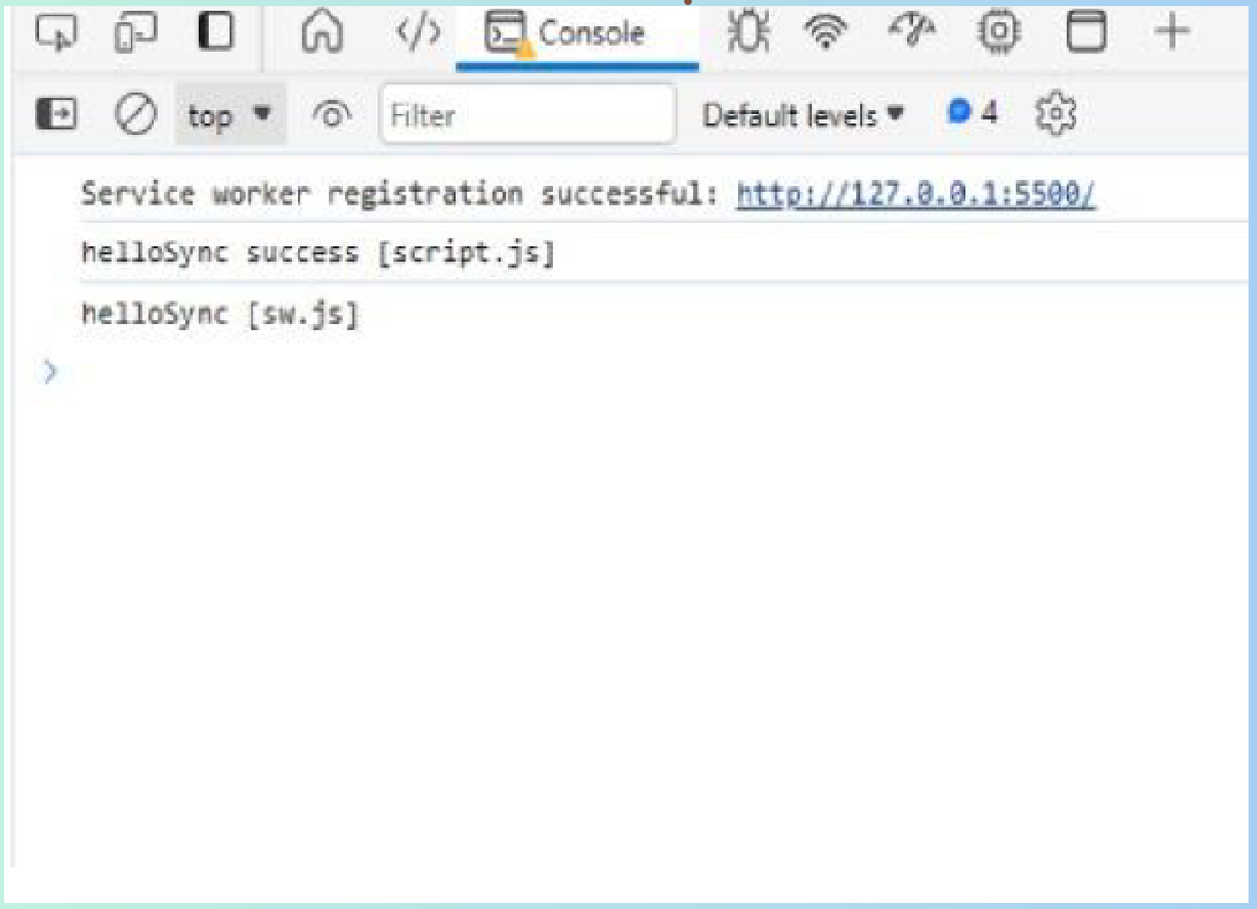
FETCH EVENT

You can track and manage page network traffic with this event. You can check existing cache, manage “cache first” and “network first” requests and return a response that you want.

.

SYNC EVENT

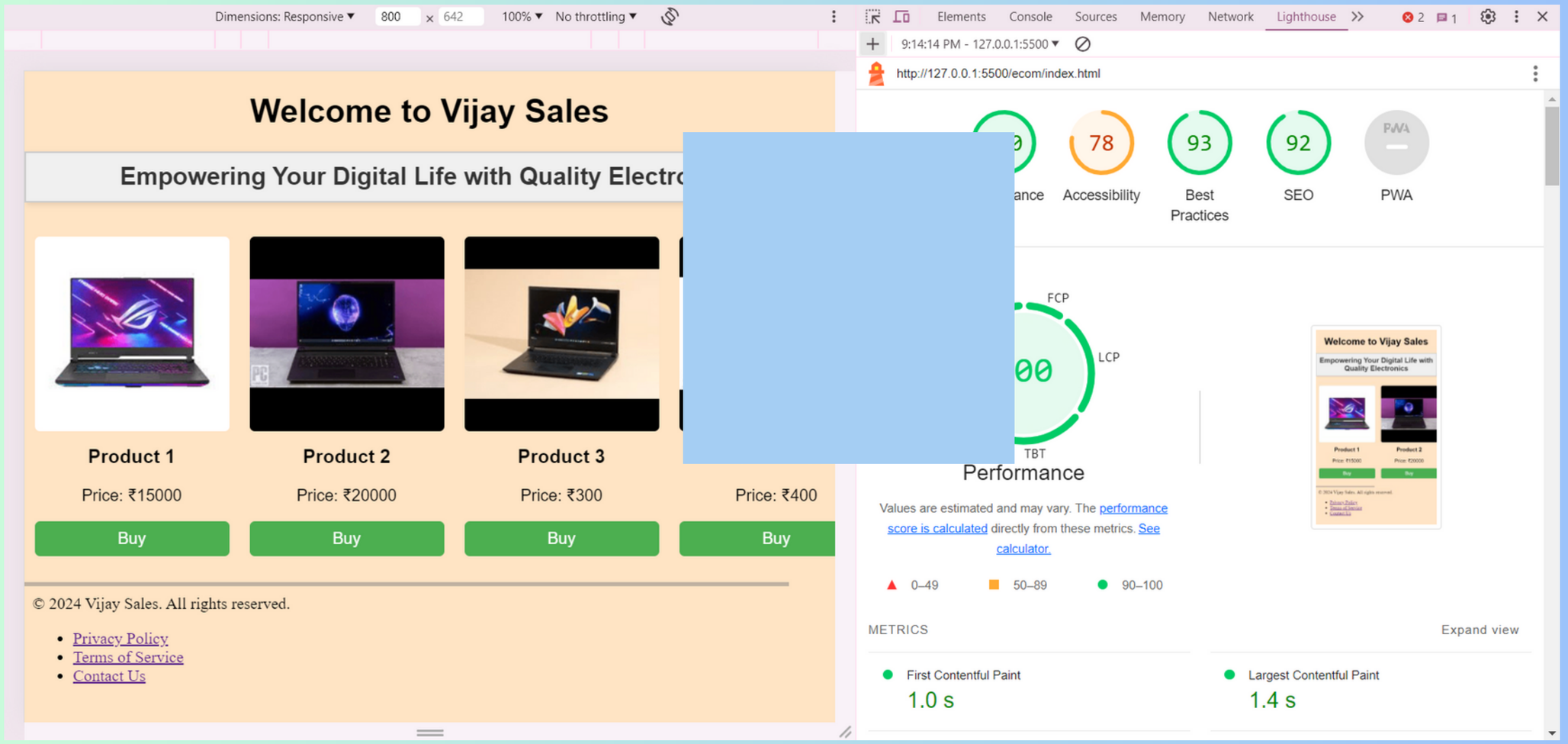
Background Sync is a Web API that is used to delay a process until the Internet connection is stable. We can adapt this definition to the real world; there is an e-mail client application that works on the browser and we want to send an email with this tool.



Performance Analysis with Google Lighthouse

Google Lighthouse is an open-source tool developed by Google that helps developers improve the quality and performance of web pages and web applications. It is commonly used to audit and analyze various aspects of a website, including performance, accessibility, best practices, SEO (Search Engine Optimization), and Progressive Web App (PWA) functionality.

Performance Analysis with Google Lighthouse



Conclusion

In conclusion, our journey in building the E-commerce Progressive Web Application (PWA) has been marked by significant achievements. From the successful installation and service worker registration to the seamless implementation of fetch, push, and sync events, followed by deployment and thorough performance analysis using Google Lighthouse, we have demonstrated the power and versatility of PWAs in modern web development. The importance of PWAs for enhancing user experience cannot be overstated, and as we look to the future, there are ample opportunities for further enhancements and innovations in our E-commerce PWA, ensuring continued growth and success in the ever-evolving digital landscape.

Thank you for listening!

Nishtha Joshi

2024 April 4