



FLUTTER PRESENTATION

“ORDER EASY”

Name :
Sanika Rane (D15B/57)

Guided by -
Dr.Ravita Mishra





Order Easy

Table of C O N T E N T S

01

Introduction

02

Objectives

03

Software and
Hardware
Requirements

04

Implementation

05

Conclusion

06

Discussion

07

Future Scope

08

References



Introduction

“ORDER EASY” is Flutter app, dedicated to developing a specialized Food Ordering Application tailored for a specific restaurant, designed to simplify your dining journey and enhance overall satisfaction.

Objectives



- **Efficiency Enhancement :**
Streamlining the traditional food ordering process by leveraging technology to minimize waiting times and improve overall efficiency.

- **Revolutionizing Ordering Methods :**
Introducing innovative approaches to food ordering that break away from traditional methods, thereby providing a seamless and time-saving experience for customers.

- **Enhanced User Flexibility :**
Provide users with the flexibility to modify their orders within a specified time frame, catering to evolving preferences or last-minute additions.

Software Requirements

- **Flutter SDK:** Install Flutter for cross-platform app development.
- **Dart Programming Language:** Utilize Dart for coding the application logic.
- **IDE:** Use preferred IDEs such as Visual Studio Code or IntelliJ IDEA for coding and debugging.
- **Flutter Packages:** provider: For state management within the application.
- **http:** For making HTTP requests to fetch data from remote servers.
- **path_provider:** For accessing filesystem paths.
- **intl:** For internationalization and localization support.

Firebase Integration:

- **Firebase Authentication :** Firebase authentication service for user authentication
- **Firestore :** Real-time data base to store customer's orders and menu

Hardware Requirements

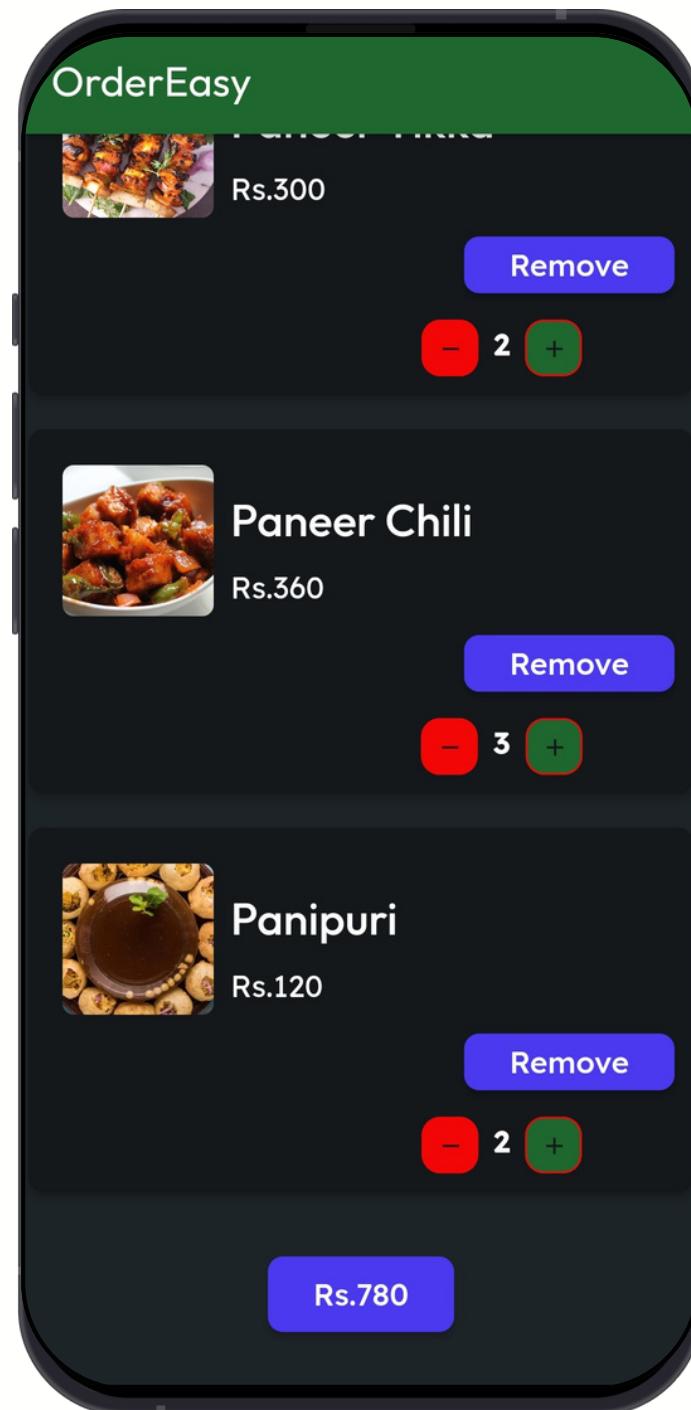
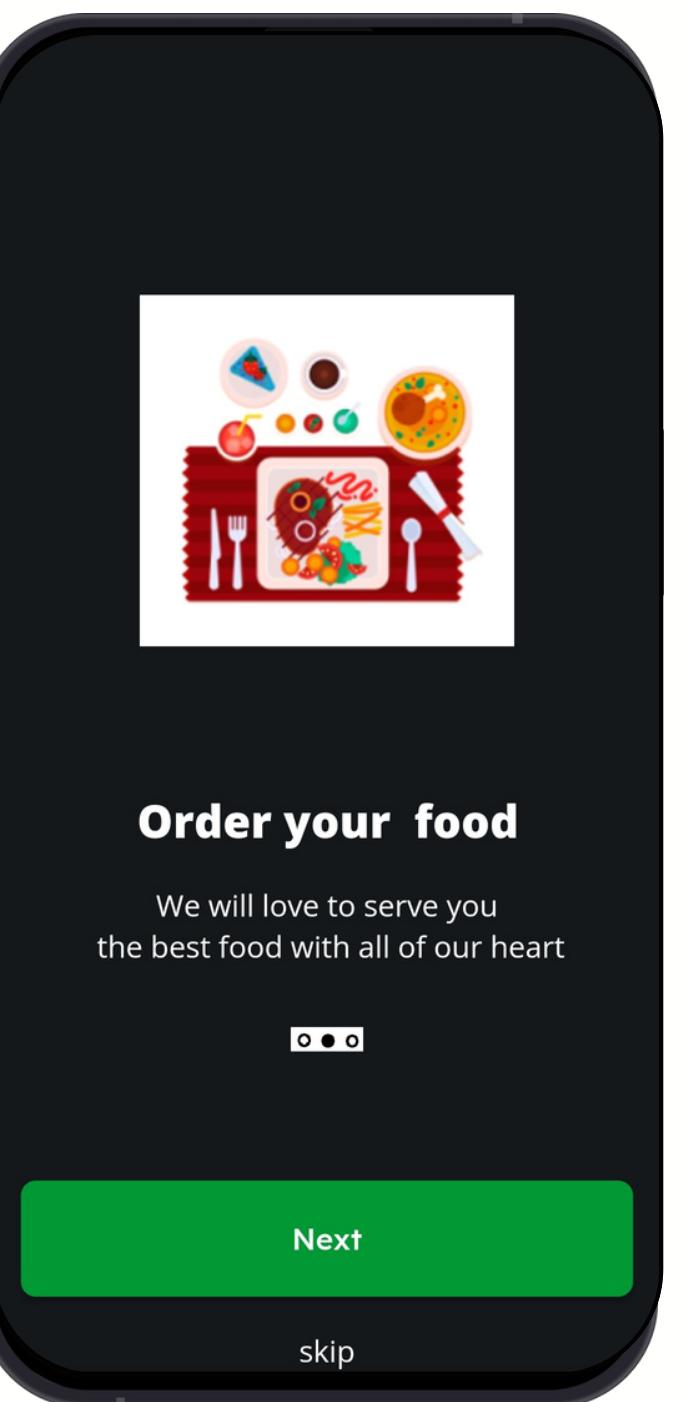
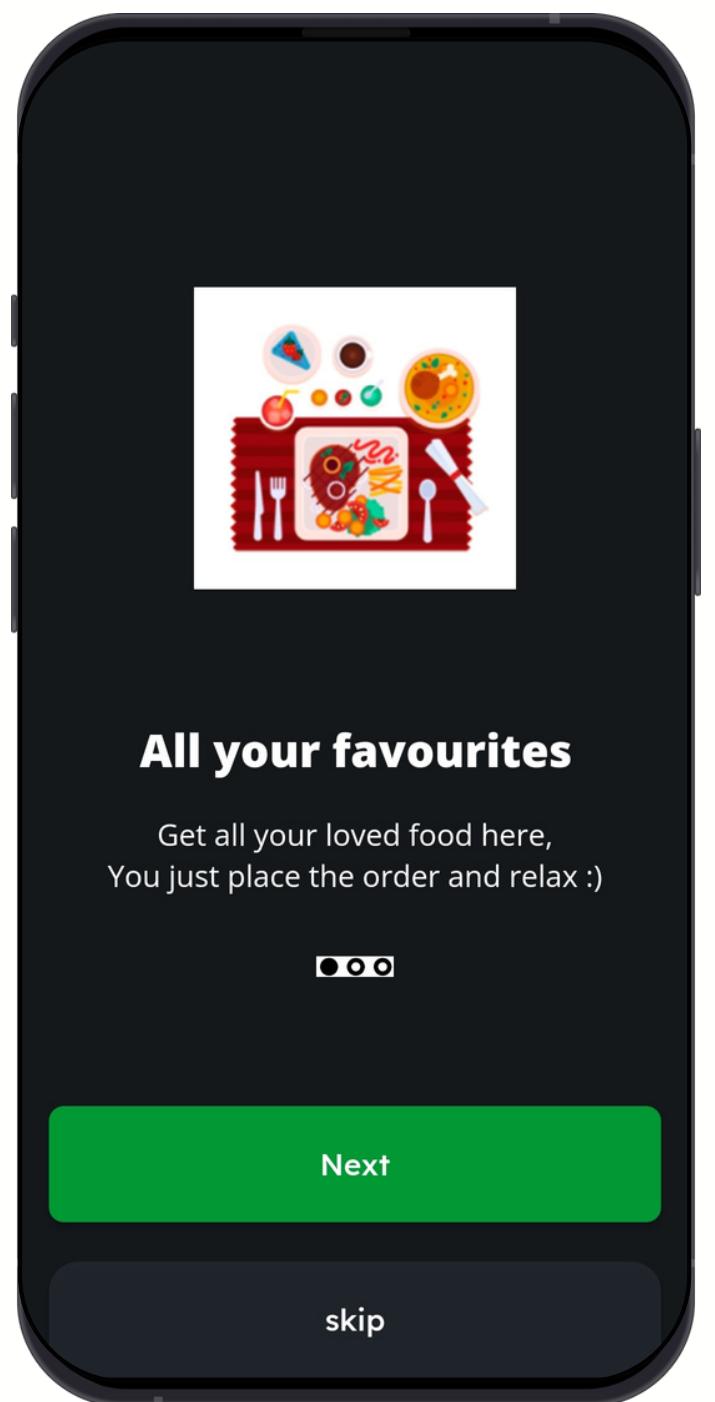
Computer

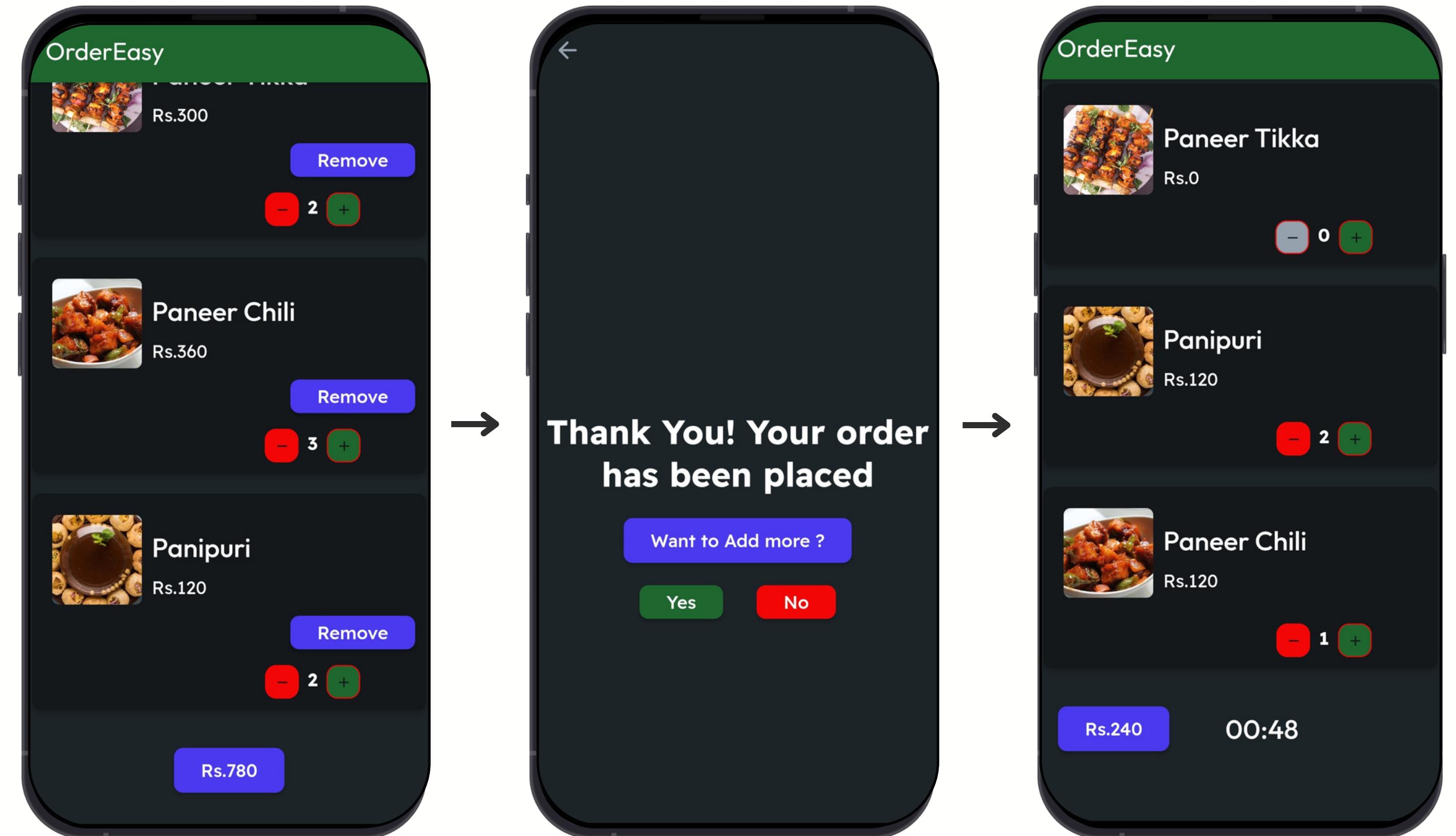
- **Processor:** Intel i3/i5/i7/i9 or equivalent AMD Processor.
- **Memory:** 4GB RAM minimum, 8 GB or more recommended.
- **Disk Space:** Minimum of 5 GB free disk space, SSD preferred.
- **Operating System:** Windows (7 SP1 or later), macOS, or Linux.

Mobile Devices (for testing)

- **Android :** Android phone or tablet with a recent Android version.

Implementation





Dynamic

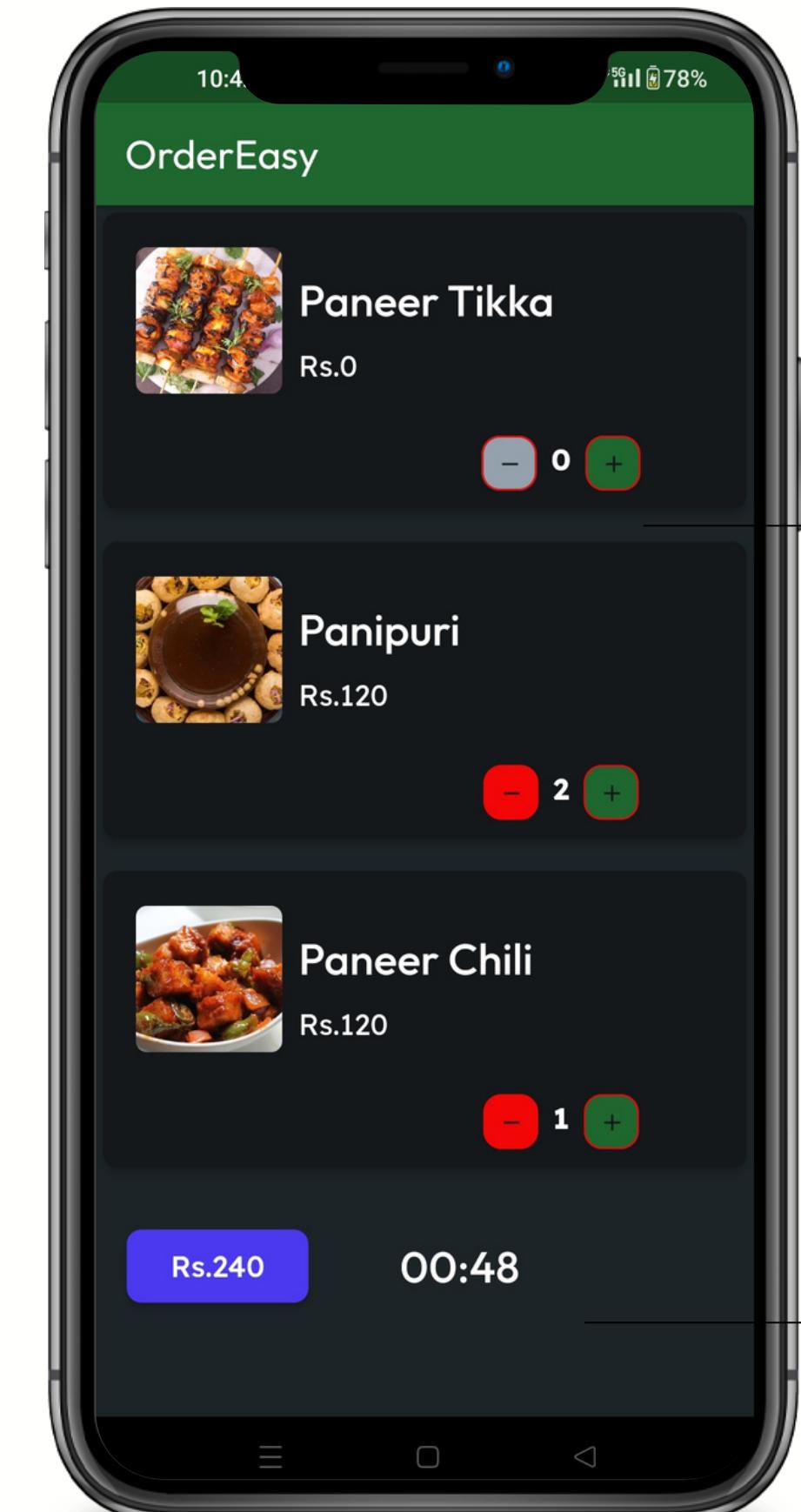
Features

Cart Functionality

- Can add foodtimes to cart

Timer

- Can increase quantity , as a last time add-ons
- Added items will get reflected in the database



● Cart functionality

Fetching data from firestore and storing customers's cart

● Timer

Features

Firebase Authentication

The screenshot shows the Firebase Authentication console for a project named "orderEasy1". The left sidebar includes links for Project Overview, Release Monitor (NEW), Authentication (selected), Realtime Database, Firestore Database, Storage, Product categories, and Build. The main area is titled "Authentication" and contains tabs for Users, Sign-in method, Templates, Usage, Settings, and Extensions. The "Users" tab is active, displaying a table with two rows of user data. The columns are Identifier, Providers, Created, Signed In, and User UID. The first user is tanvisaraf@gmail.com, created on Apr 5, 2024, signed in on Apr 5, 2024, and has a User UID of W52UYA1hosMLY0noPEaxIPU... The second user is ranesanika3421@gmail.com, also created on Apr 5, 2024, signed in on Apr 5, 2024, and has a User UID of XEJkZrl348hpPSe1mtLn2Hchj... A search bar at the top allows searching by email address, phone number, or user UID. A blue "Add user" button is located in the top right corner of the table header.

Identifier	Providers	Created	Signed In	User UID
tanvisaraf@gmail.com	✉	Apr 5, 2024	Apr 5, 2024	W52UYA1hosMLY0noPEaxIPU...
ranelsanika3421@gmail...	✉	Apr 5, 2024	Apr 5, 2024	XEJkZrl348hpPSe1mtLn2Hchj...

FireStore

The screenshot shows the Firebase Cloud Firestore interface for a project named "orderEasy1". The left sidebar contains navigation links for Project Overview, Release Monitor, Authentication, Realtime Database, Firestore Database (which is selected), Storage, Product categories, Build, Release & Monitor, Analytics, Engage, and All products. At the bottom, there are "Spark" and "Upgrade" buttons. The main area displays a document in the "Menuitems" collection with the ID "46lQF3Ux3ZqbcZ6kKd5W". The document structure is as follows:

```
menuitems
  46lQF3Ux3ZqbcZ6kKd5W
    category: "Main Course"
    id: "6a"
    image: "https://firebasestorage.googleapis.com/v0/b/ordereeasy1-fceff.appspot.com/o/cms_uploads%2FMenuitems%2F171227121230900?alt=media&token=400e8802-d5b0-4053-a03d-41ffc5fe7d38"
    item: "Paneer Tikka"
    price: 240
```

Firebase Storage

The screenshot shows the Firebase Storage interface for the project 'orderEasy1'. The left sidebar contains navigation links for Project Overview, Release Monitor, Authentication, Realtime Database, Firestore Database, and Storage, with Storage being the active tab. Below these are sections for Product categories (Build, Release & Monitor, Analytics, Engage) and All products. At the bottom, there are links for Spark (No-cost \$0/month) and Upgrade.

The main area is titled 'Storage' and shows a file tree under the path `gs://ordereeasy1-fceff.appspot.com > cms_uploads > Menuitems`. The tree lists seven folder entries:

Name	Size	Type	Last modified
1712270969204000/	—	Folder	—
1712271016844000/	—	Folder	—
1712271057117000/	—	Folder	—
1712271127063000/	—	Folder	—
1712271180264000/	—	Folder	—
1712271212309000/	—	Folder	—

At the top right of the main area, there are buttons for 'Upload file' (blue), '+', and three dots for more options. A banner at the top right encourages protecting storage resources from abuse with 'Configure App Check'.

Conclusion

In conclusion, the development of the Food Ordering Application represents a significant leap forward in revolutionizing the traditional methods of food ordering. Through meticulous planning, innovative technology integration, and a steadfast commitment to enhancing user experience, this project has successfully addressed key challenges associated with time-consuming ordering processes.

REFERENCES

01

Flutter documentation

<https://docs.flutter.dev/>

02

**“LITERATURE REVIEW ON CONSUMER PERCEPTION
TOWARDS ONLINE FOOD DELIVERY APPS”**

-By Gaurav Gawde

03

Firebase documentation

<https://firebase.google.com/docs>



PWA

PRESNTATION

Furniture E-commerce

A screenshot of a Progressive Web Application (PWA) for a furniture and home decor store named "Woodex". The interface includes a header with the store name, a search bar, and user account icons. Below the header, there are five product cards displayed in a grid. The products are: "Art Deco Home Decoration" (two wooden vases), "Helen Chair" (a modern wooden armchair with grey cushion), "Vase Of Flowers" (a textured grey vase), "Wood Eggs" (three wooden egg-shaped bowls), and "Table Wood Pine Furniture" (a wooden side table with orange and grey drawers). At the bottom right of the screen, the text "Woodex Store" is written in a cursive script font.

RedStore Website - Red Store | E-commerce Website

Search Anything...

Woodex

User icons: profile, heart (0), cart (0), menu

Product cards:

- Art Deco Home Decoration
- Helen Chair
- Vase Of Flowers
- Wood Eggs
- Table Wood Pine Furniture

Woodex Store

Name :
Sanika Rane (D15B/57)

Guided by -
Dr.Ravita Mishra



Features of PWA

1. Add to Home Screen:

This feature allows users to save a PWA to their device's home screen with an icon, similar to native apps. Once added, the PWA can be launched directly from the home screen, providing a full-screen experience without the browser UI, thereby enhancing user engagement and ease of access.

2. Service Worker Events: Fetch, Sync, Push:

- Fetch: Intercept network requests to cache or retrieve assets, enabling offline use or faster load times.
- Sync: Delay actions until a stable internet connection is available, useful for posts or updates made offline.
- Push: Receive push notifications from a server, even when the browser is closed, keeping users engaged with timely content or alerts.

3. Offline Storage:

Allows data to be stored locally on the user's device for offline access and improved performance.

Implementation

- Cache storage

The screenshot shows a browser window with developer tools open. The main content area displays a web page titled "Woodex" featuring a "Art Deco Home" section with a "Decoration" category and two wooden vases with flowers. The developer tools sidebar has the "Application" tab selected, showing storage details for the origin `http://127.0.0.1:5500`. It lists Local storage, Session storage, IndexedDB, Web SQL, Cookies, Private state tokens, Interest groups, Shared storage, and Cache storage. The Cache storage section shows a table with 26 entries, including files like CSS, fonts, and icons. The Network tab at the bottom shows log entries for successful fetches from both cache and network, along with an uncaught error message.

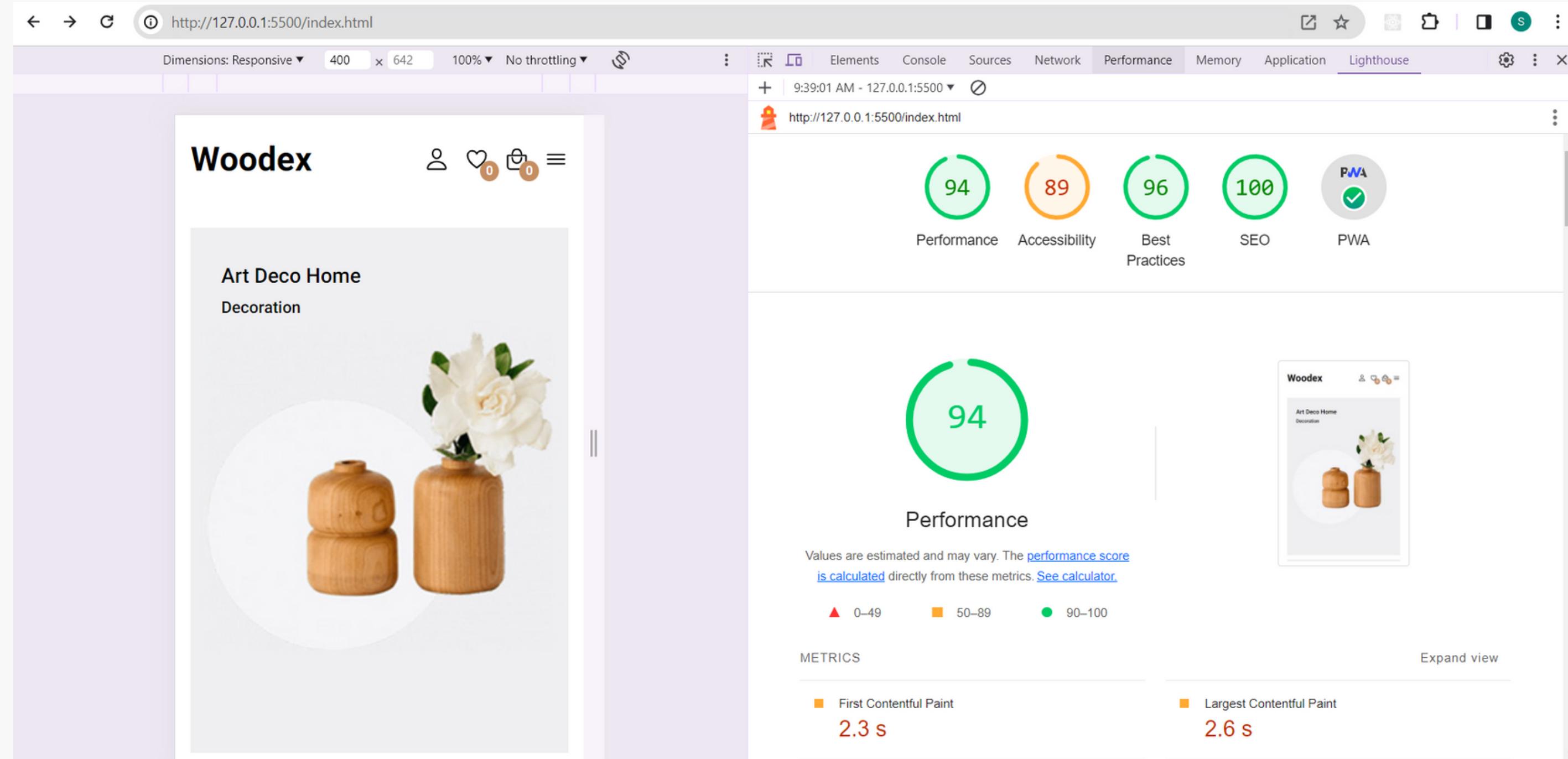
#	Name	Respons...	Content...	Content...	Time Ca...	Vary He...
0	/css?family=Mr+De+Haviland&family=Rob...	opaque	text/css	0	4/4/202...	
1	/css?family=Poppins:wght@300;400;500;60...	opaque	text/css	0	4/4/202...	
2	/s/mrdehaviland/v14/OpNVnoohJj96FdB73...	cors	font/wo...	17,084	4/4/202...	
3	/s/roboto/v30/KFOICnqEu92Fr1MmEU9fBBc...	cors	font/wo...	15,920	4/4/202...	
4	/s/roboto/v30/KFOICnqEu92Fr1MmWUIfBBc...	cors	font/wo...	15,860	4/4/202...	
5	/s/roboto/v30/KFOICnqEu92Fr1Mu4mxKw...	cors	font/wo...	15,744	4/4/202...	
6	/font-awesome/4.7.0/css/font-awesome.min...	opaque	text/css	0	4/4/202...	Accept...
7	/ionicons@5.5.2/dist/ionicons/ionicons.esm.js	cors	application...	0	4/4/202...	Accept...

Total entries: 26

Console

```
Fetch successful (from cache): http://127.0.0.1:5500/index.html
Fetch successful (from network, cached): http://127.0.0.1:5500/manifest.json
Fetch successful (from cache): http://127.0.0.1:5500/manifest.json
Fetch successful (from network, cached): http://127.0.0.1:5500/assets/css/style.css
Fetch successful (from cache): http://127.0.0.1:5500/assets/css/style.css
Uncaught (in promise) TypeError: Failed to execute 'clone' on 'Response': Response body is already used
at sw.js:91:72
```

- Lighthouse (Performance analysis)



THANK YOU!!