



VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY
(AUTONOMOUS INSTITUTE AFFILIATED TO UNIVERSITY OF MUMBAI, APPROVED BY AICTE
& RECOGNIZED BY GOVT. OF MAHARASHTRA, NAAC ACCREDITED WITH 'A' GRADE)

DEPARTMENT OF INFORMATION TECHNOLOGY

FLUTTER presentation

ArtistryIndia

Name: Vaishnavi Chaudhary

D15B / 09

Guided by-
Dr. Ravita Mishra



ArtistryIndia

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

ArtistryIndia is Flutter app, your gateway to the vibrant world of Indian art forms! Immerse yourself in the rich tapestry of India's diverse artistic traditions as we guide you through a journey of discovery. Our platform invites you to join a global community where artists from around the world come together to share their creativity, collaborate, and celebrate the beauty of Indian art. Experience the magic of centuries-old techniques and modern interpretations as we explore the captivating landscapes of Indian creativity.



OBJECTIVES

Promote Cultural Appreciation

ArtistryIndia aims to promote cultural appreciation by providing users with a gateway to explore and immerse themselves in the rich tapestry of Indian art forms, fostering a deeper understanding and appreciation for India's diverse artistic traditions.

Facilitate Global Community Engagement

The platform encourages global community engagement by inviting artists from around the world to connect, share their creativity, collaborate, and celebrate the beauty of Indian art, creating a vibrant and inclusive space for artistic expression and exchange.

Blend Tradition with Modernity

ArtistryIndia offers a blend of centuries-old techniques and modern interpretations, allowing users to experience the magic of Indian creativity across various art forms, from classical to contemporary, ensuring a captivating journey of discovery for enthusiasts of all backgrounds.



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.
 - sqflite: For local storage and database operations.
 - http: For making HTTP requests to fetch data from remote servers.
 - path_provider: For accessing filesystem paths.
 - intl: For internationalization and localization support.
 - shared_preferences: For storing simple data in key-value pairs persistently.
 - flutter_spinkit: For displaying loading spinners or indicators.
- **Firebase Integration:**
 - Firebase Account: Set up a Firebase account to utilize Firebase services for various functionalities like real-time database, authentication, cloud messaging, etc.

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.

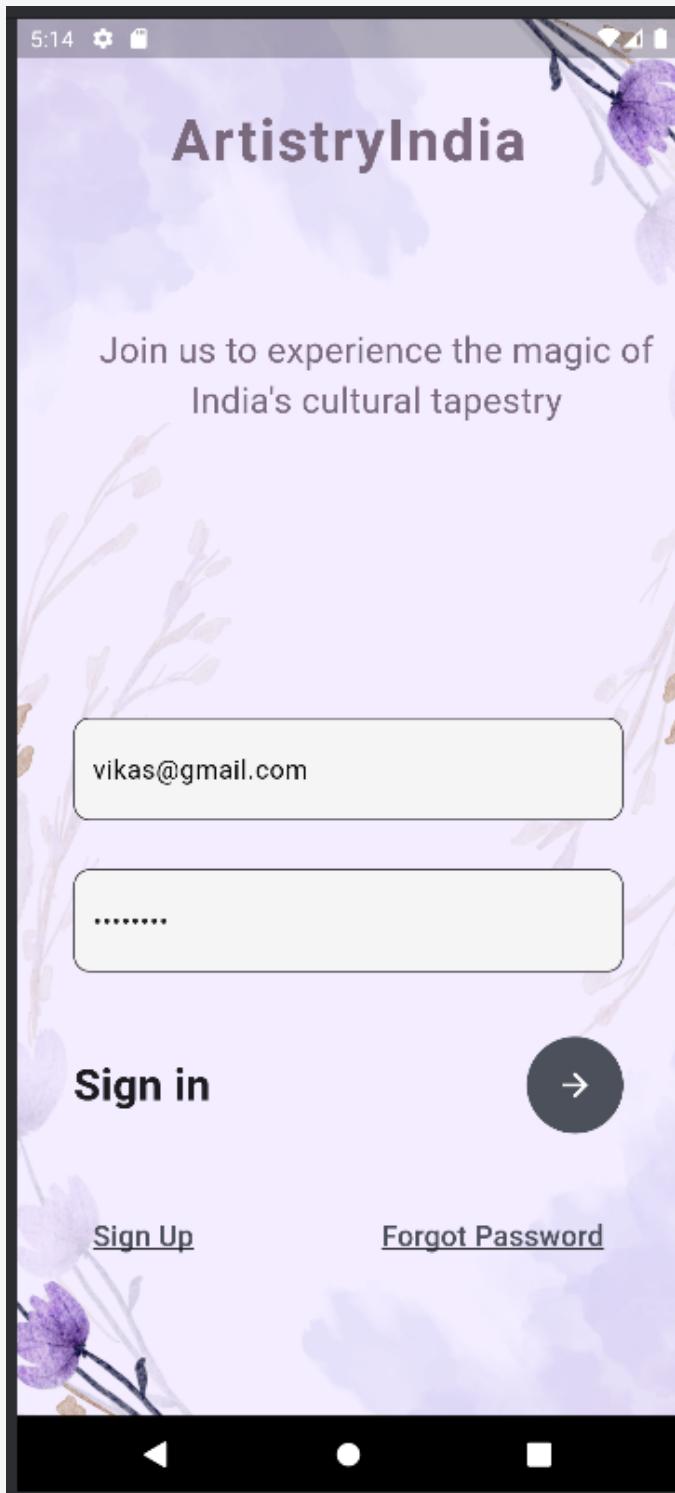
Mobile Devices (for testing)

- iOS: iPhone or iPad running the latest iOS version.
- Android: Android phone or tablet with a recent Android version.

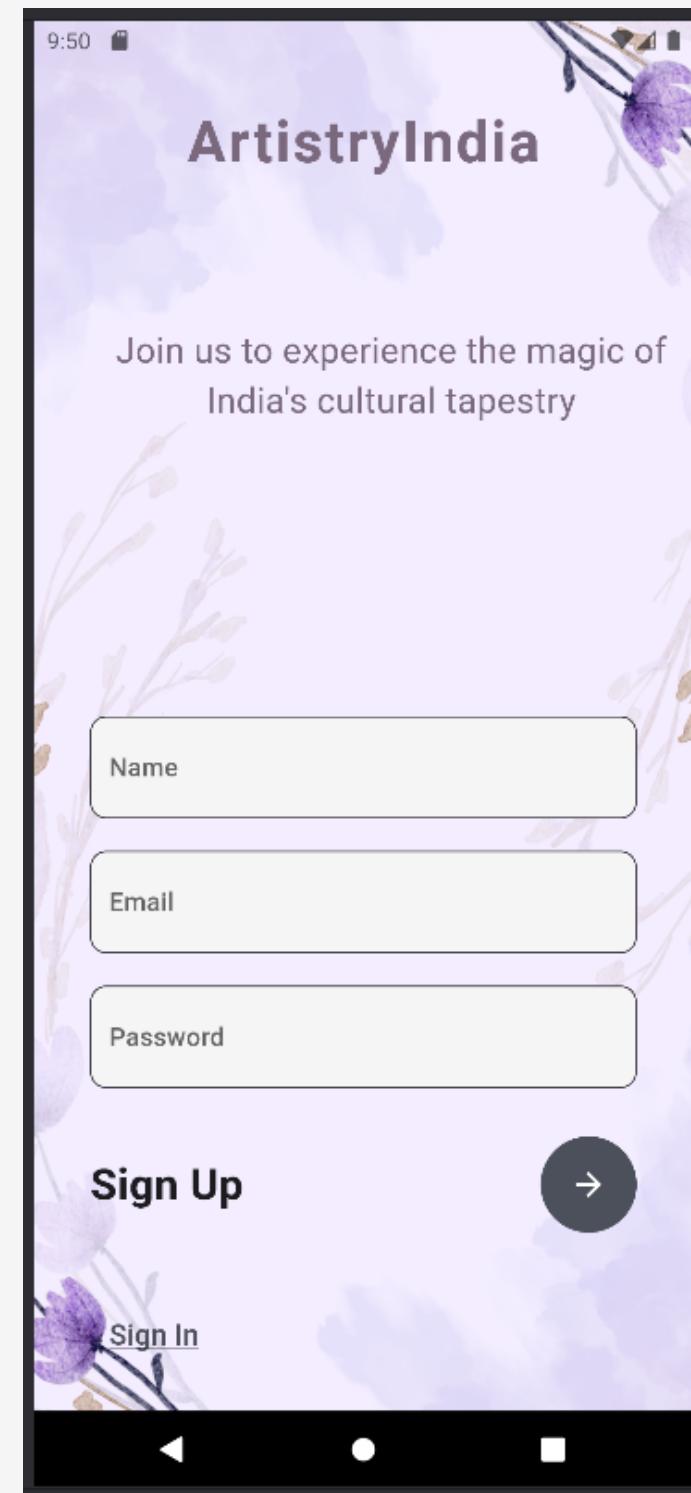


IMPLEMENTATION

LOGIN :



SIGNUP:



FIREBASE AUTHENTICATION:

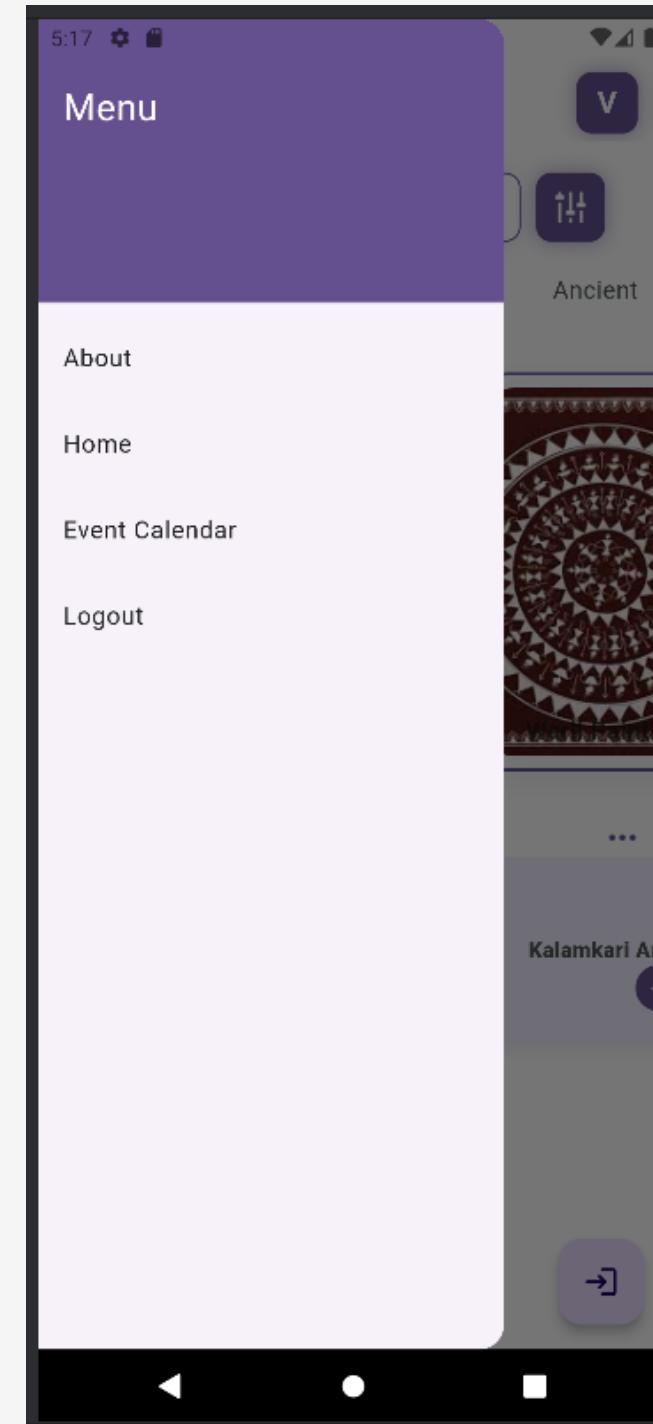
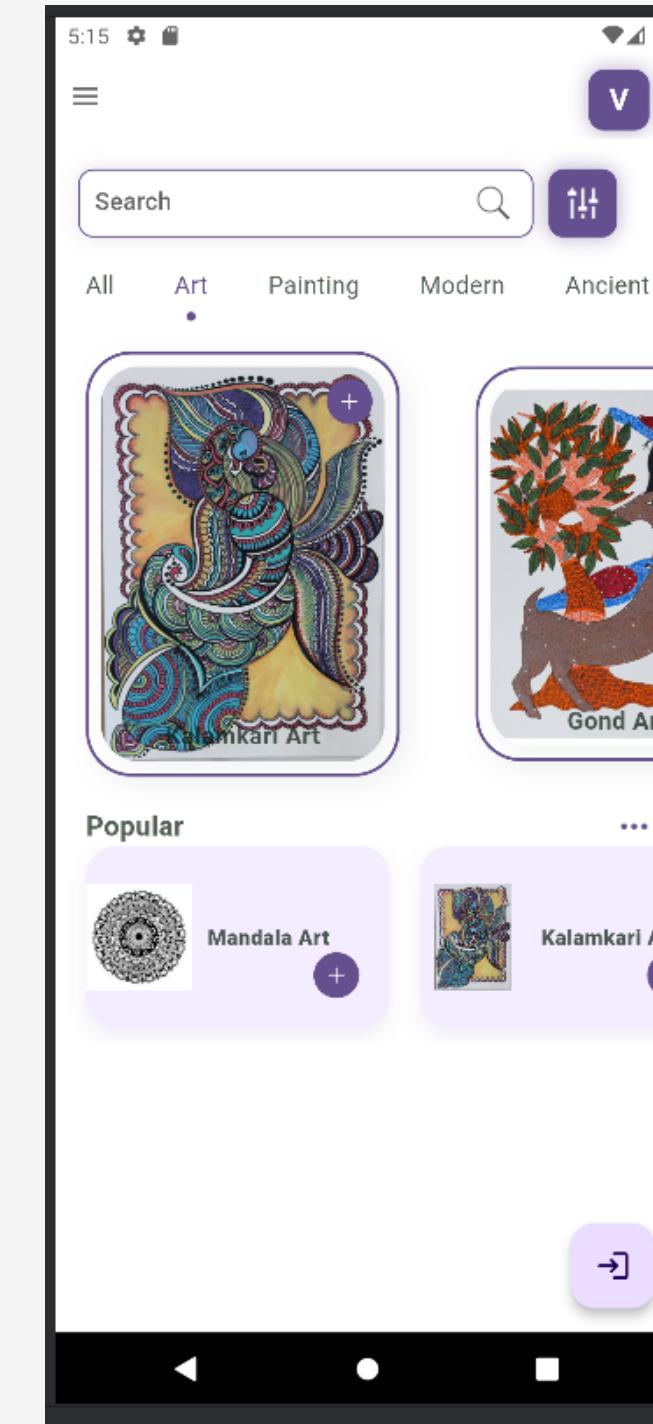
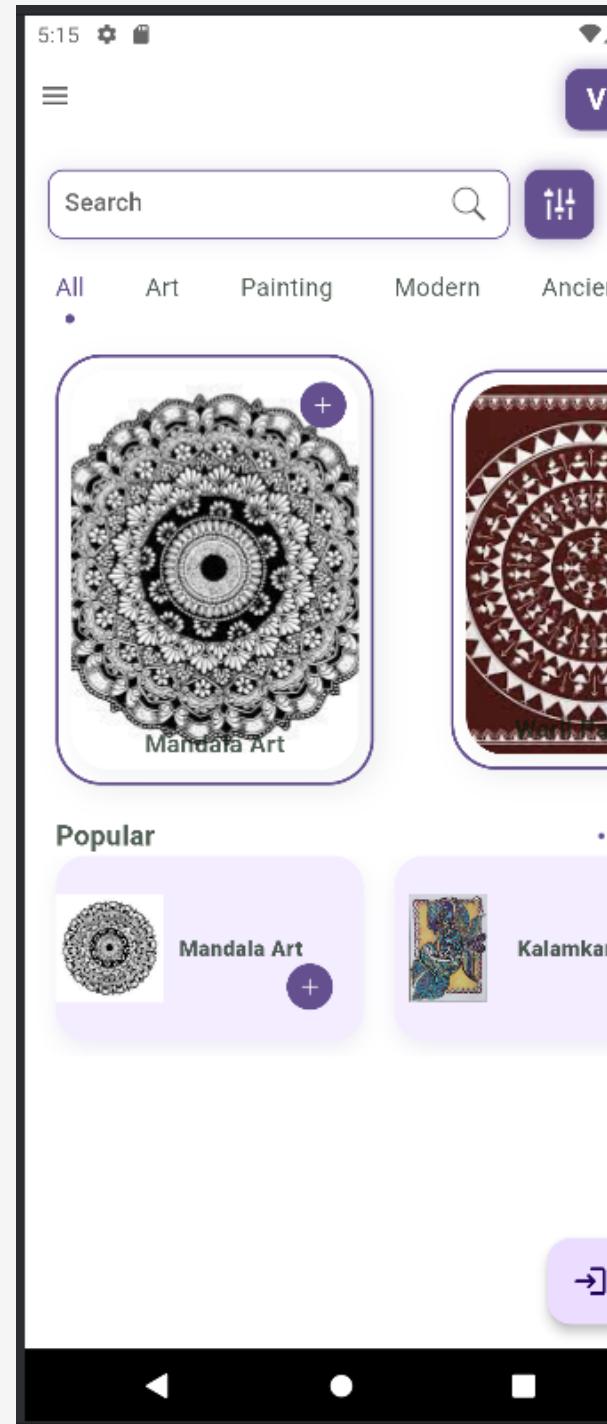
A screenshot of the Firebase Authentication console for the project "ArtistryIndia". The interface shows the "Authentication" tab selected. Below it, the "Users" tab is active, showing a list of users. The table includes columns for Identifier, Providers, Created, Signed In, and User UID. Three users are listed: vikas@gmail.com (Created Mar 31, 2024), vishal@gmail.com (Created Mar 4, 2024), and vaishnavi@gmail.com (Created Mar 4, 2024). The bottom of the screen shows standard browser navigation and search controls.

Identifier	Providers	Created	Signed In	User UID
vikas@gmail.com	✉	Mar 31, 2024		LT39uLjBndXj1Qwe03gIVH1K...
vishal@gmail.com	✉	Mar 4, 2024		byFUU9AApVeC1yr9uMrH0L...
vaishnavi@gmail.com	✉	Mar 4, 2024	Mar 4, 2024	TMaAgkjY0SSsDpwc9VQjpigJ...



IMPLEMENTATION

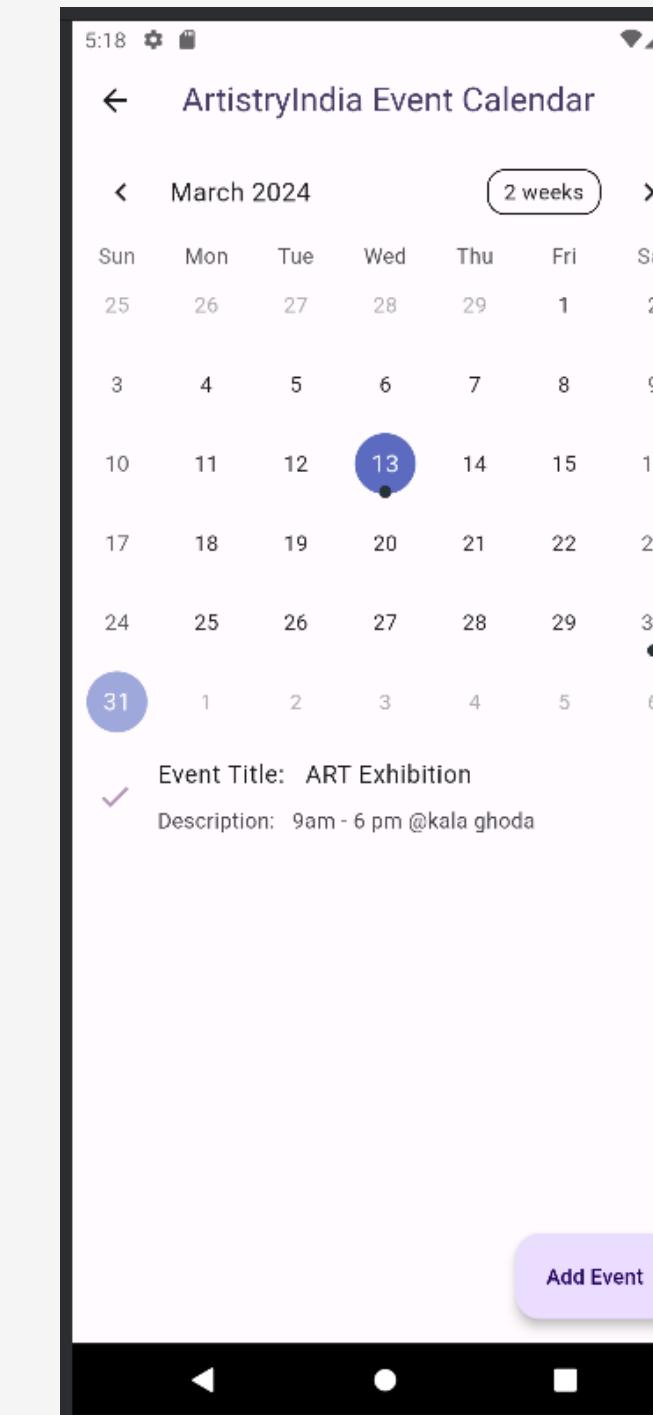
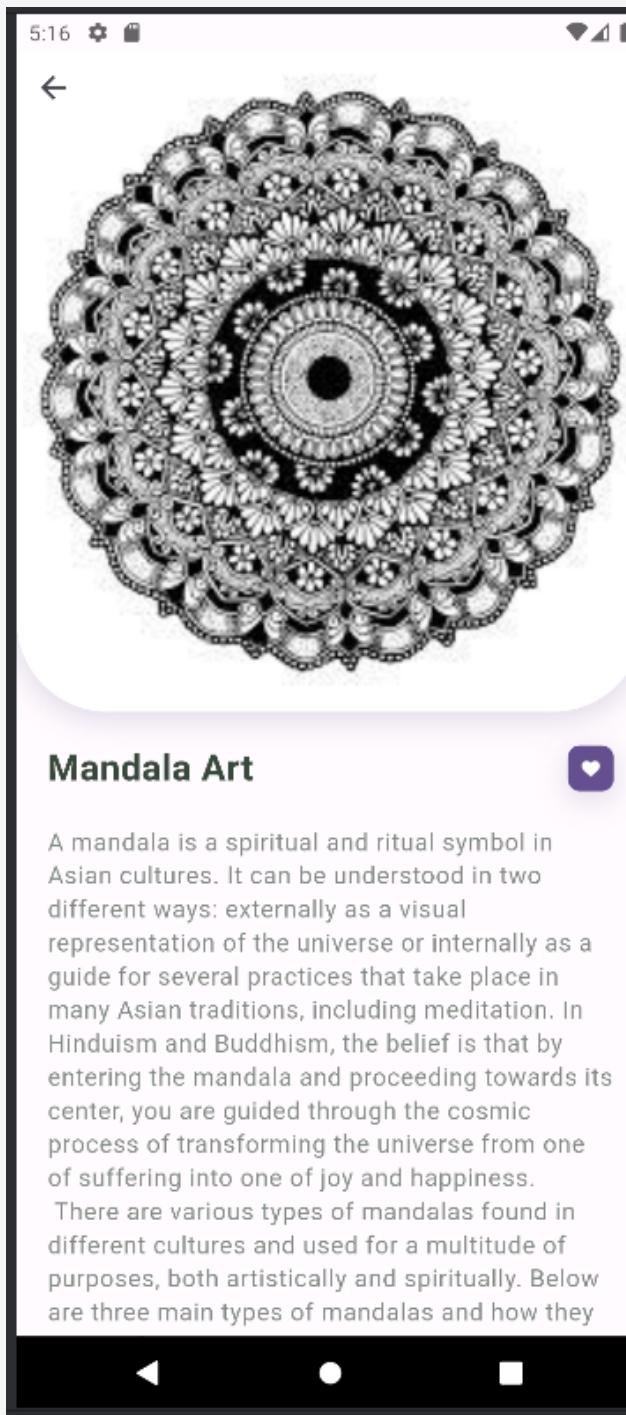
HOME PAGE :



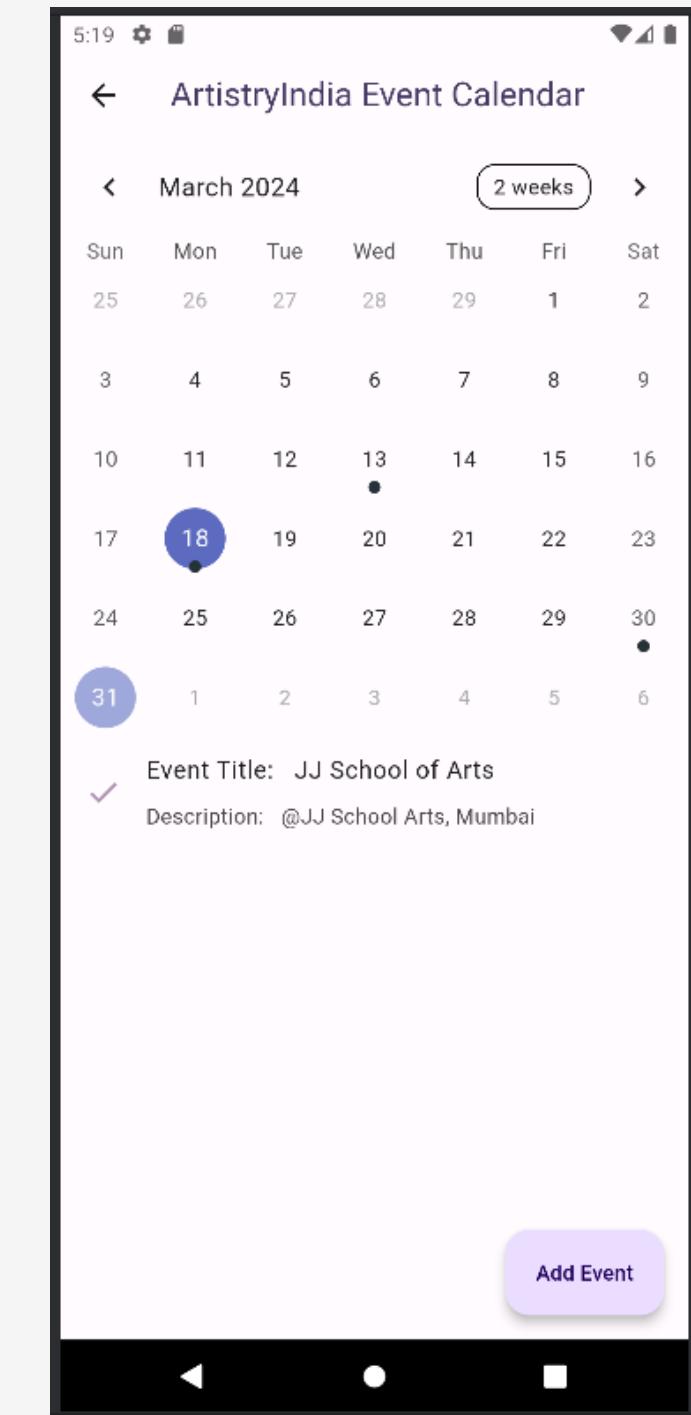
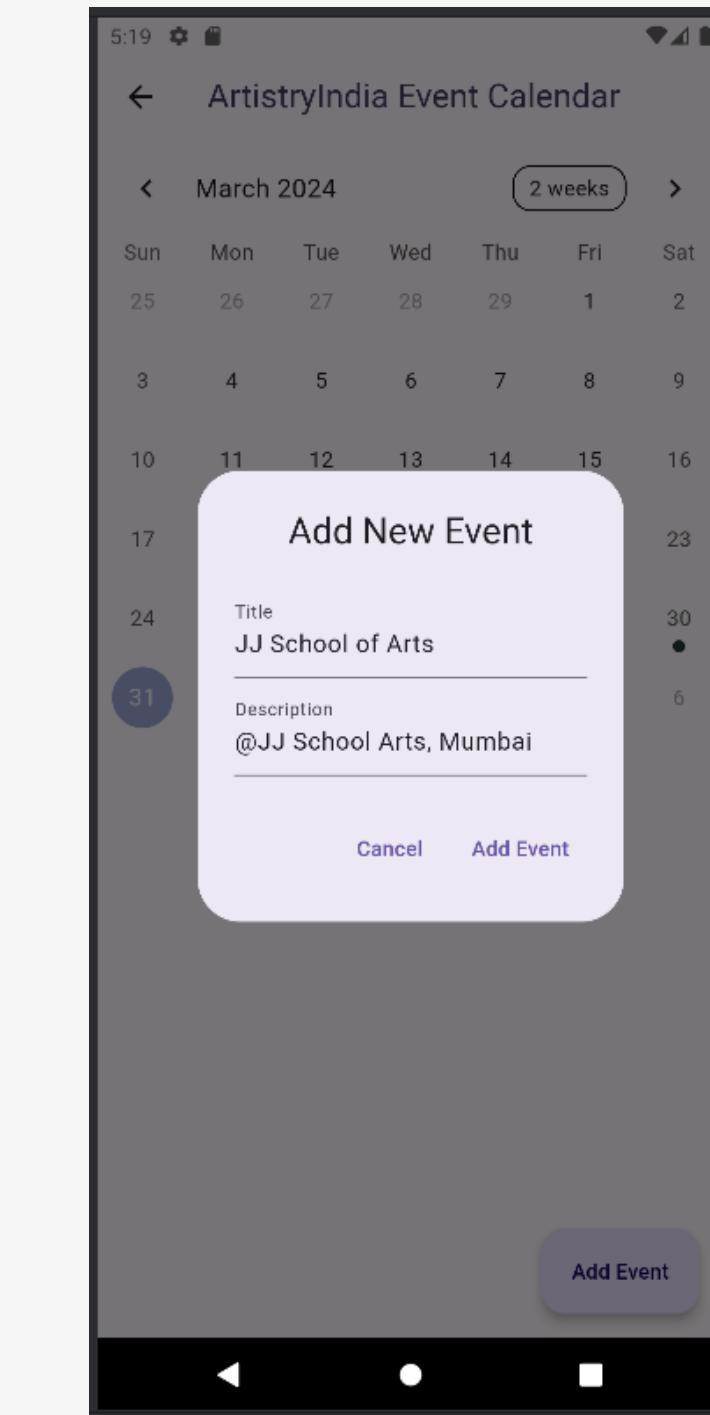


IMPLEMENTATION

DETAILS PAGE:



EVENT CALENDAR:





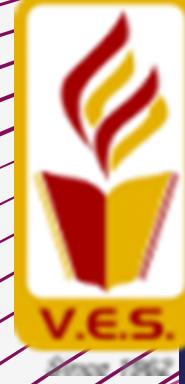
CONCLUSION & DISCUSSION

- **Conclusion:** ArtistryIndia is not just a mobile application; it's a cultural journey, a platform for global connectivity, and a celebration of India's rich artistic heritage. Through seamless integration of Flutter technology, users are transported into the vibrant world of Indian art forms, where they can explore, learn, and engage with a diverse community of artists from around the globe. By blending centuries-old techniques with modern interpretations, ArtistryIndia encapsulates the essence of Indian creativity, offering users an immersive experience that transcends geographical boundaries.
- **Discussion:** The content of the Flutter app, ArtistryIndia, reflects a holistic approach towards promoting Indian art and culture in the digital age. It emphasizes the importance of accessibility and inclusivity by providing a platform that is not only educational but also interactive and engaging. Through features like virtual tours, artist communities, and educational resources, the app fosters a deeper understanding and appreciation for Indian art forms among users of all backgrounds. Furthermore, the utilization of Flutter technology enables seamless cross-platform development, ensuring that the app reaches a wider audience across different devices and operating systems. The incorporation of essential Flutter packages and Firebase integration enhances the app's functionality, offering users a smooth and efficient experience.



FUTURE SCOPE

- 1. Art Style Recognition with ML:** Develop a feature where users can upload artwork images for the app's ML model to identify Indian art styles, providing insights into their origins and significance, enriching users' understanding of Indian art forms.
- 2. Personalized Recommendations with Data Science:** Utilize data science to analyze user preferences and interactions, enabling the app to offer personalized recommendations for new artwork, events, and collaborations tailored to individual interests, enhancing user engagement.
- 3. AR Integration for Interactive Experiences:** Integrate AR technology allowing users to interact with virtual Indian art pieces overlaid in real-world environments via their device's camera, providing immersive experiences and fostering deeper appreciation of Indian art.



REFERENCES

01

Flutter documentation

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

02

MyIndianArt

<https://play.google.com/store/apps/details?id=com.myindianart.wis&hl=en&gl=US>

03

Firebase documentation

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

04

Add Firebase to your Flutter apps

<https://firebase.google.com/docs/flutter/setup?platform=ios>



THANK YOU !



VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY
(AUTONOMOUS INSTITUTE AFFILIATED TO UNIVERSITY OF MUMBAI, APPROVED BY AICTE
& RECOGNIZED BY GOVT. OF MAHARASHTRA, NAAC ACCREDITED WITH 'A' GRADE)

DEPARTMENT OF INFORMATION TECHNOLOGY

PWA presentation

Name: Vaishnavi Chaudhary

D15B / 09

Guided by-
Dr. Ravita Mishra



FEATURES

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.

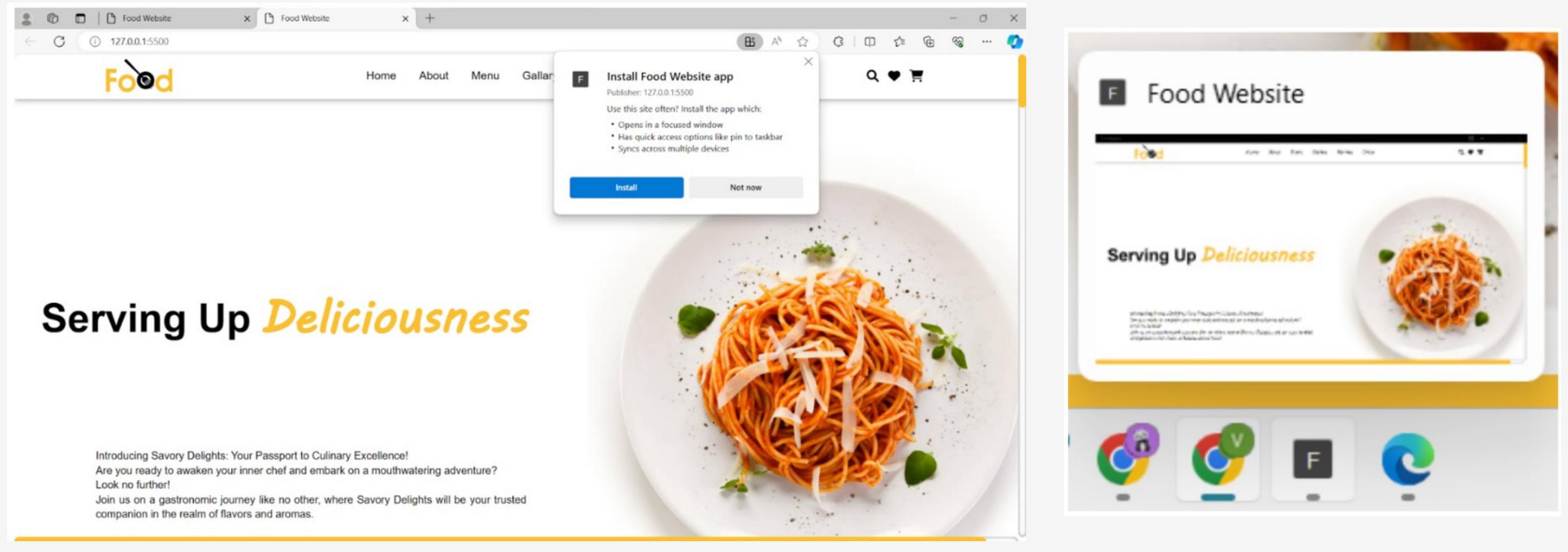
3. Offline Storage:

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



IMPLEMENTATION

- add to homescreen feature





IMPLEMENTATION



service worker events like fetch, sync and push

The screenshot shows the Chrome DevTools Service Worker panel for the URL `http://127.0.0.1:5500/`.
Storage: Local storage, Session storage, IndexedDB, Web SQL, Cookies, Private state t, Interest group, Shared storage, Cache storage (offline - htt, dynamic-pv).
Clients: `http://127.0.0.1:5500/ focus`.
Push: `{"method": "pushMessage", "message": "He"}` Push button.
Sync: `test-tag-from-devtools` Sync button.
Periodic Sync: `test-tag-from-devtools` Periodic Sync button.
Update Cycle: Version #43, Activity: Install, Wait, Activate (Timeline bar).
Background services: Back/forward, Console, Issues, Filter, Default levels 4, Settings.
Console: Fetch successful (from network, cached): `http://127.0.0.1:5500/styles.css` `sw.js:93`, Fetch successful (from cache): `http://127.0.0.1:5500/styles.css` `sw.js:65`, Fetch successful (from network, cached): `http://127.0.0.1:5500/script.js` `sw.js:93`, Fetch successful (from cache): `http://127.0.0.1:5500/script.js` `sw.js:65`.

The screenshot shows the Chrome DevTools Service Worker panel for the URL `http://127.0.0.1:5500/`.
Service worker registration: successful at `http://127.0.0.1:5500/`.
Storage: Local storage, Session storage, IndexedDB, Web SQL, Cookies, Private state t, Interest group, Shared storage, Cache storage (offline - htt).
Status: #41 activated and is running, Stop button.
Push: `{"method": "pushMessage", "message": "He"}` Push button.
Sync: `test-tag-from-devtools` Sync button.
Periodic Sync: `test-tag-from-devtools` Periodic Sync button.
Update Cycle: Version #41, Activity: Install, Wait, Activate (Timeline bar).
Background services: Back/forward, Console, Issues, Filter, Default levels 4, Settings.
Console: Service worker registration successful: `http://127.0.0.1:5500/` `script.js:93`.



IMPLEMENTATION



Offline Storage:

The screenshot shows a food ordering website with a header navigation bar containing Home, About, Menu, Gallery, Review, Order, a search icon, a heart icon, and a shopping cart icon. The main heading is "Order Now". On the left, there is an illustration of a waiter taking an order from a group of five people seated around a table. To the right, there are four input fields for Name, Email, Number, and Address, each with a placeholder text. Below these is a field for "How Much" with a placeholder "Number of Items". There is also a field for "You Order" with a placeholder "Food Name". A large yellow "Order Now" button is positioned at the bottom of the form area.

Name	<input type="text" value="Your Name"/>
Email	<input type="text" value="Your Email"/>
Number	<input type="text" value="Your Number"/>
How Much	<input type="text" value="Number of Items"/>
You Order	<input type="text" value="Food Name"/>
Address	<input type="text" value="Your Address"/>

Order Now



THANK YOU !