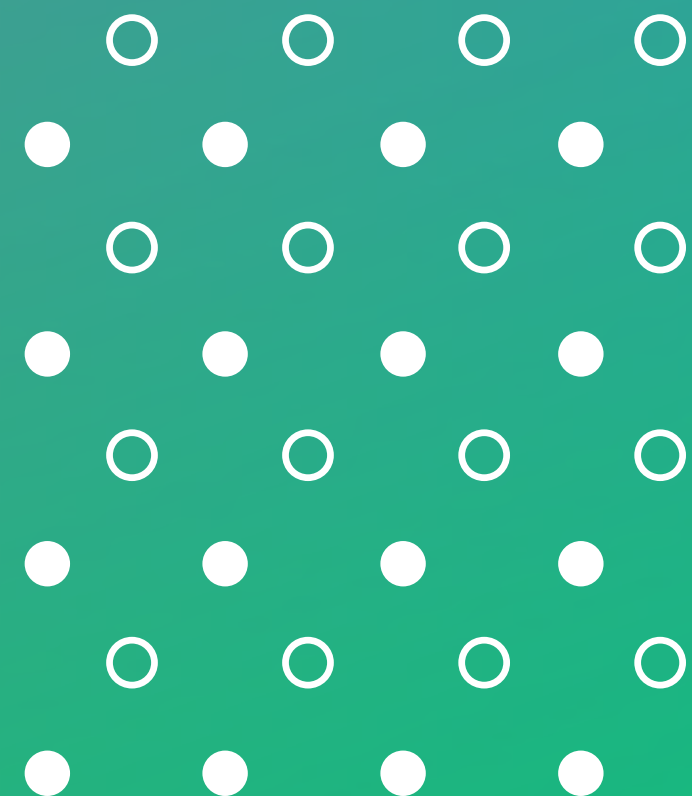


NEWSNEST



NEWSAPP DEVELOPMENT WITH FLUTTER AND FIREBASE

~ Presented by Shivani Nikam

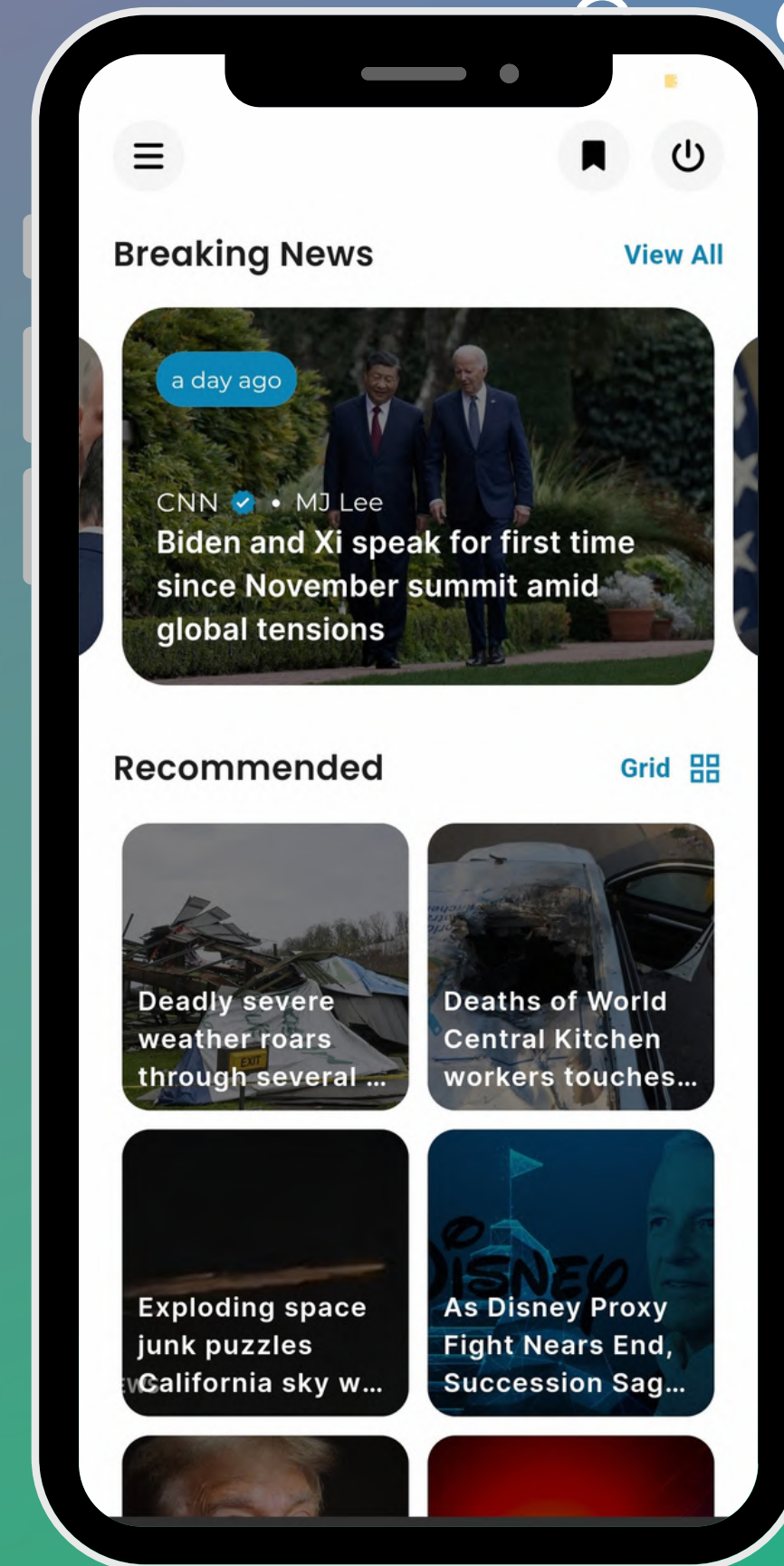


INTRODUCTION



NewsNest is a go-to destination for staying informed, connected, and organized in today's fast-paced world. Built using Flutter and Firebase, NewsNest delivers real-time news updates, seamless Google authentication, and effortless article bookmarking. Whether you're seeking a quick summary or diving deep into the details, NewsNest has you covered. Experience the future of news consumption with NewsNest - your personal news hub.

LETS GET STARTED



UNIQUE FEATURES



01.

REAL-TIME UPDATES:

Delivers real-time news updates to users, ensuring the latest information is readily available.

02.

GOOGLE AUTHENTICATION:

Access your news hub with the power of Google auth, offering a secure and hassle-free login experience.

03.

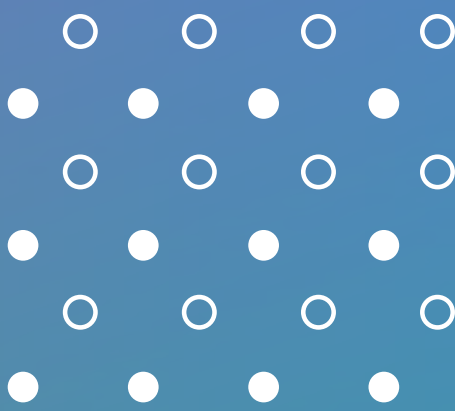
PERSONALIZATION RECOMMENDATION:

Tailor your news experience with personalized recommendations.

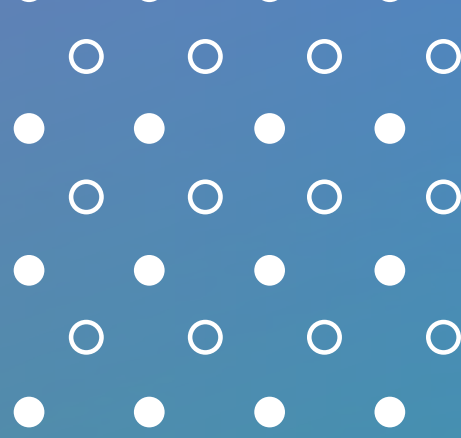
04.

DETAILED ARTICLE VIEW:

Enables users to view full articles by simply clicking on the headline, ensuring a seamless reading experience.



UNIQUE FEATURES



05.

ARTICLE SUMMARY:

Provides users with article summaries for quick understanding of the content, enhancing readability and user engagement.

06.

BREAKING NEWS SECTION:

Offers a dedicated section for breaking news, allowing users to stay informed about the latest developments as they happen.

07.

BOOKMARKING:

Offers the ability to bookmark specific articles within the app for quick access and reference, enhancing organization and ease of navigation for users.

FUTURE SCOPE



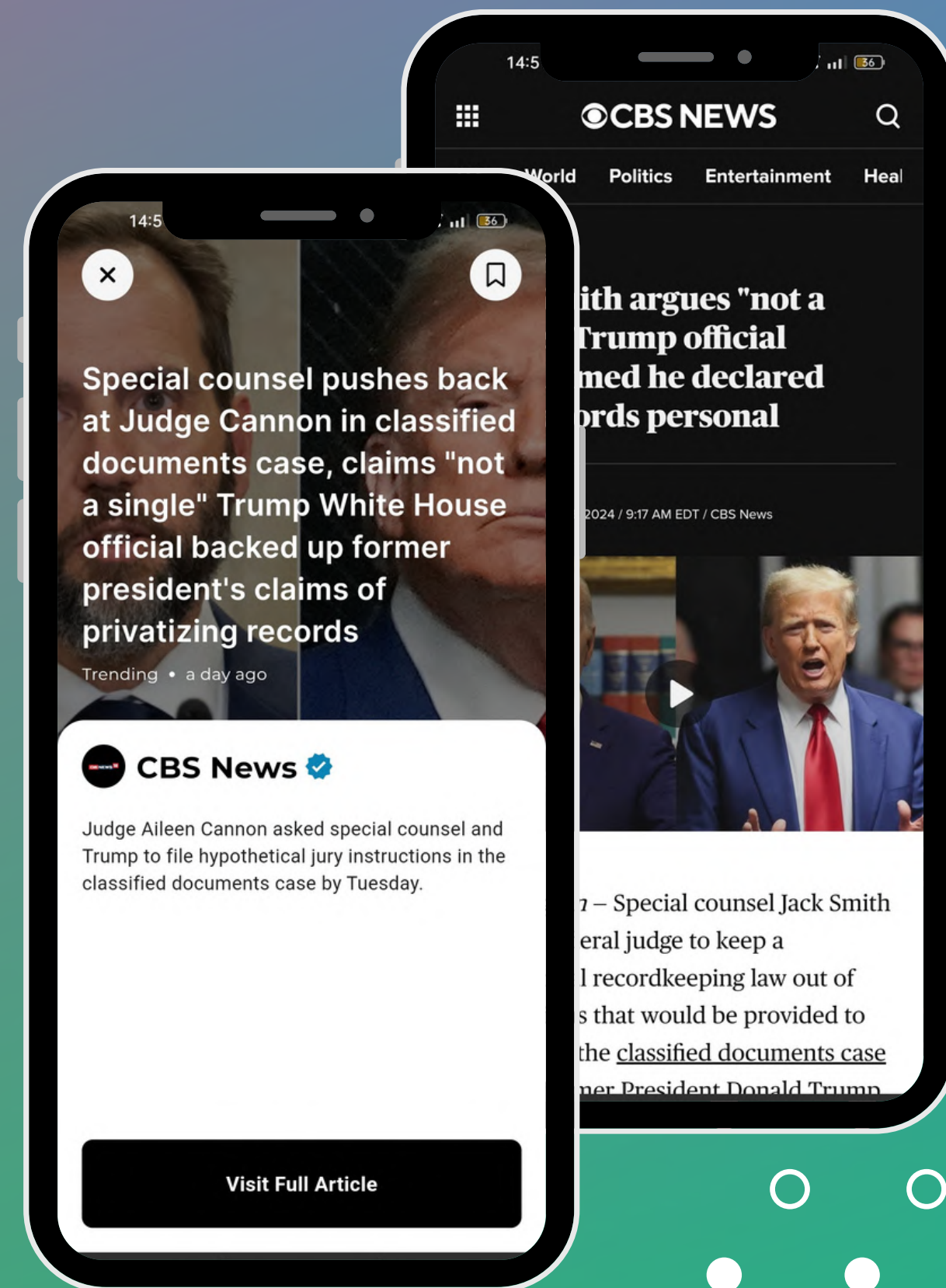
1. ENHANCED COLLABORATION:

Introduce features for users to share articles with friends, discuss news topics in real-time.

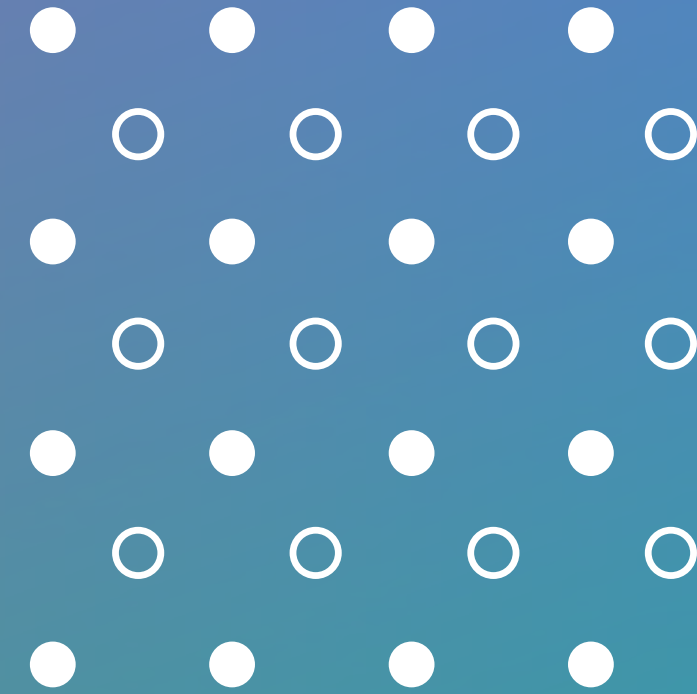


3. MULTILINGUAL SUPPORT:

Providing multilingual support, allowing users to access news content in their preferred language.



TECHNOLOGY STACK



Flutter

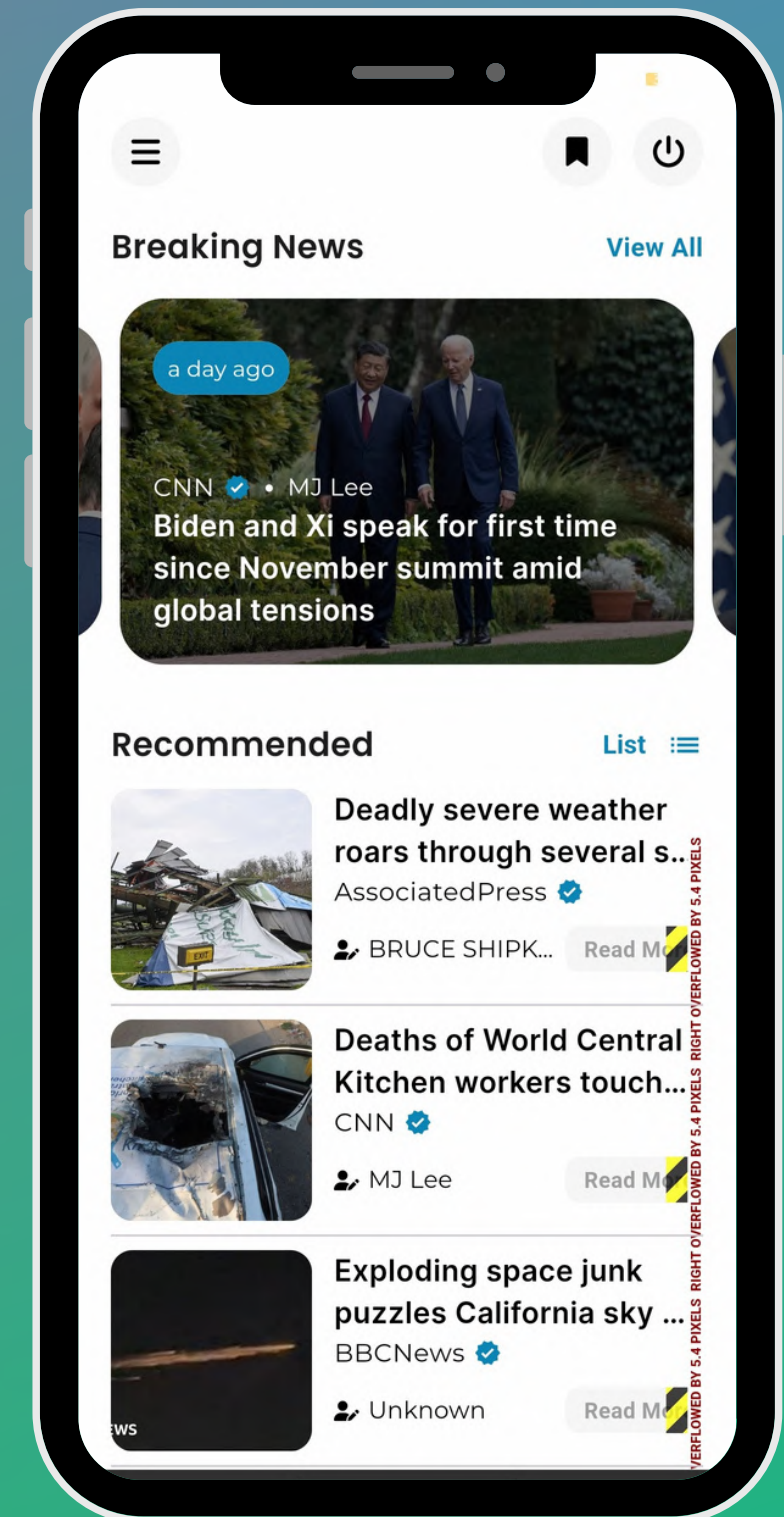
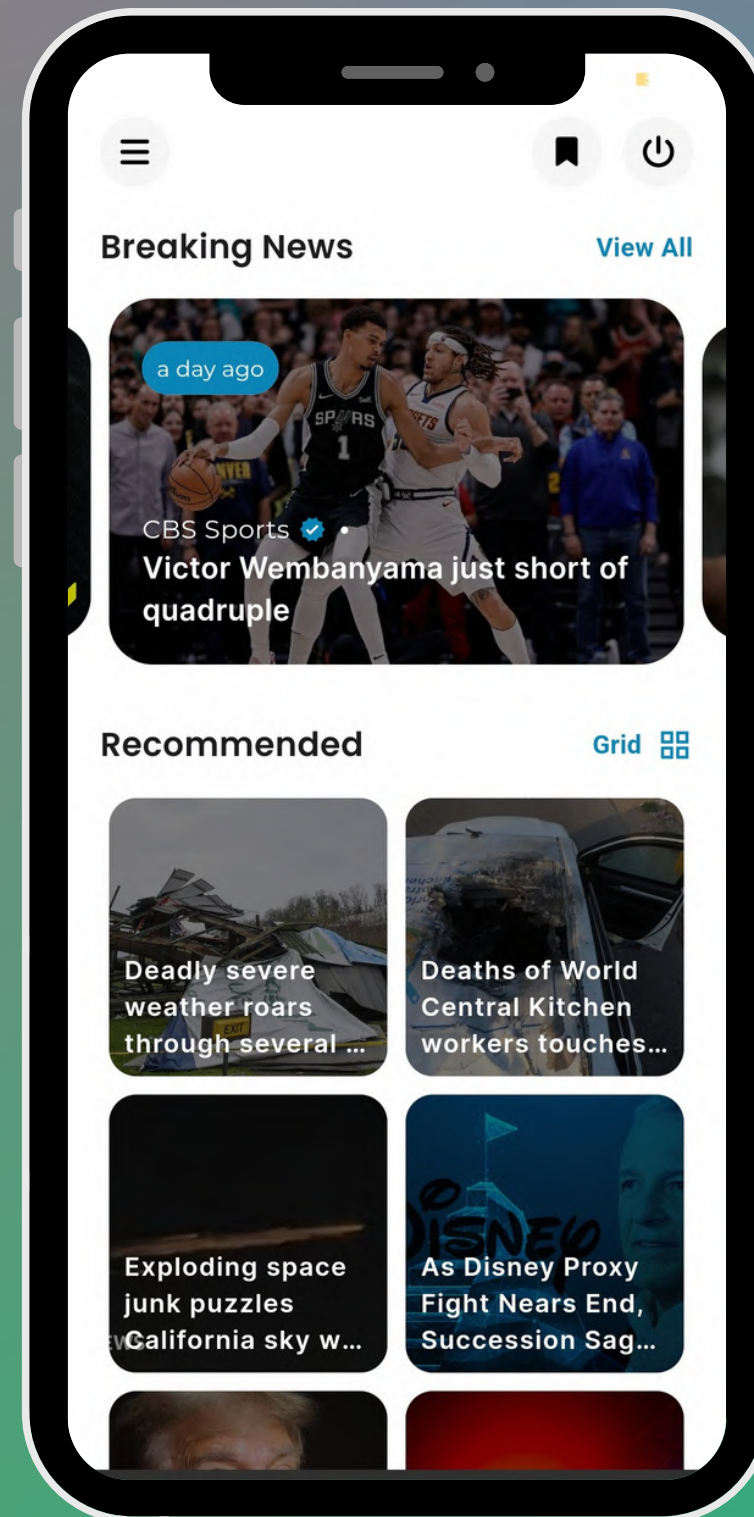
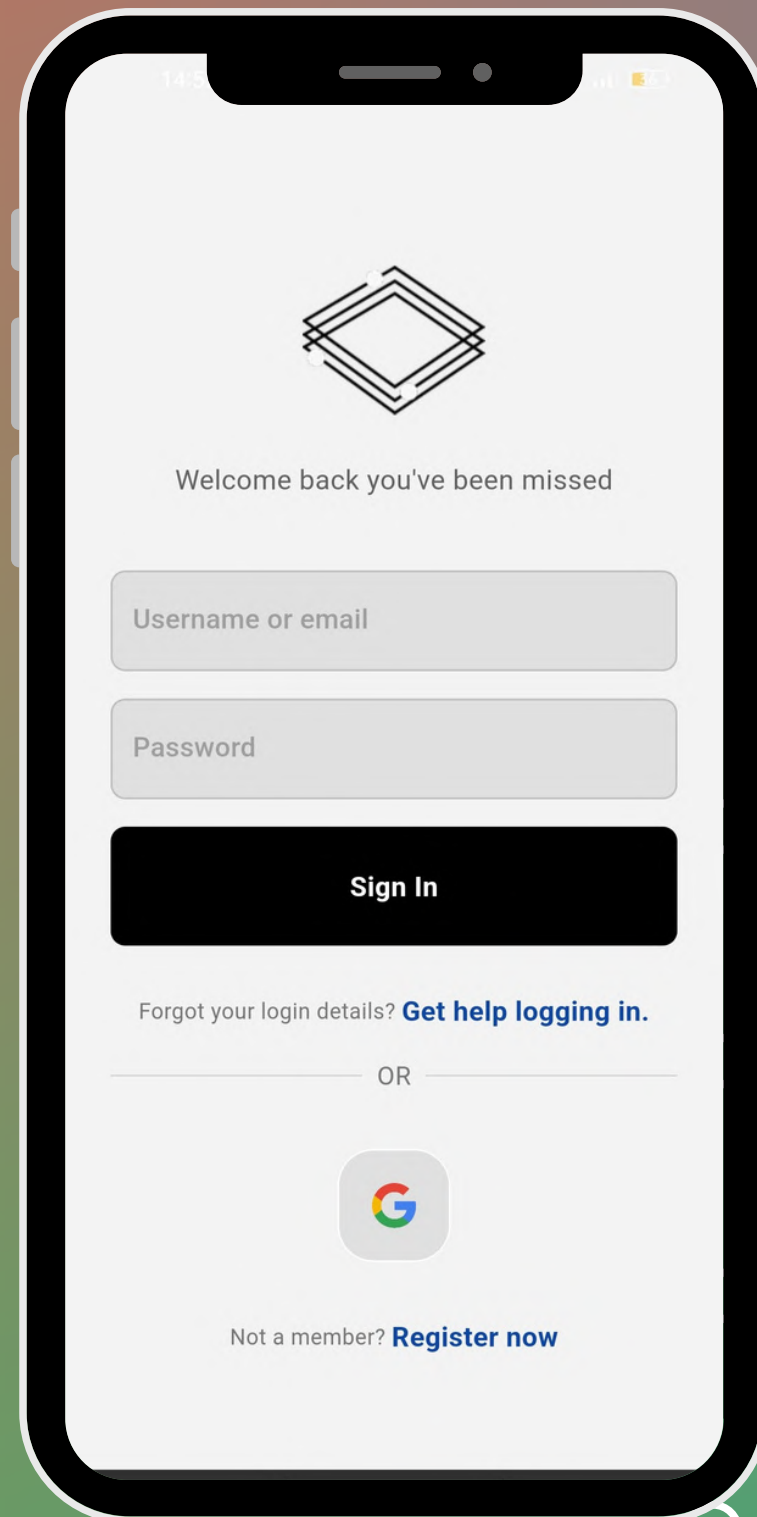


NewsAPI

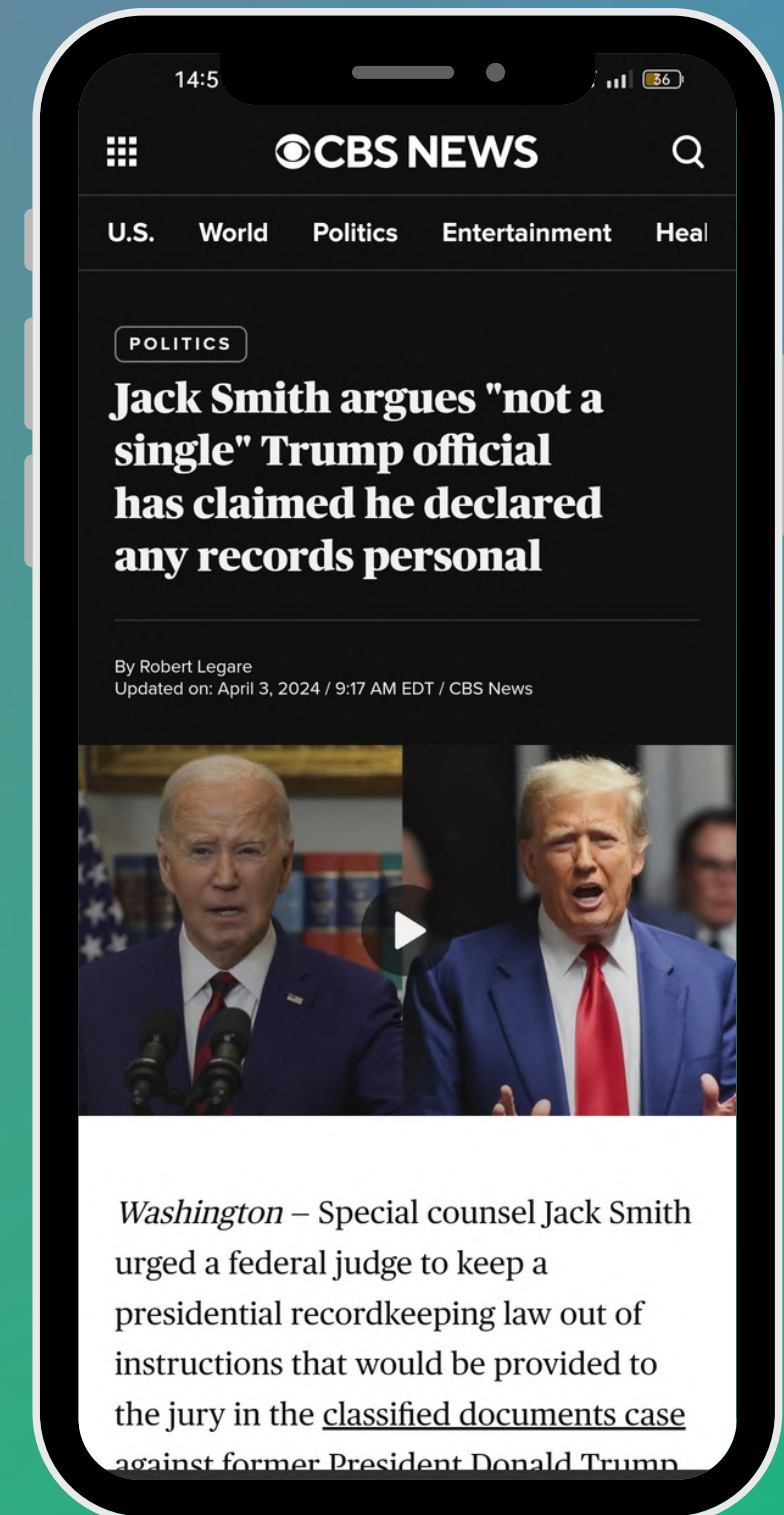
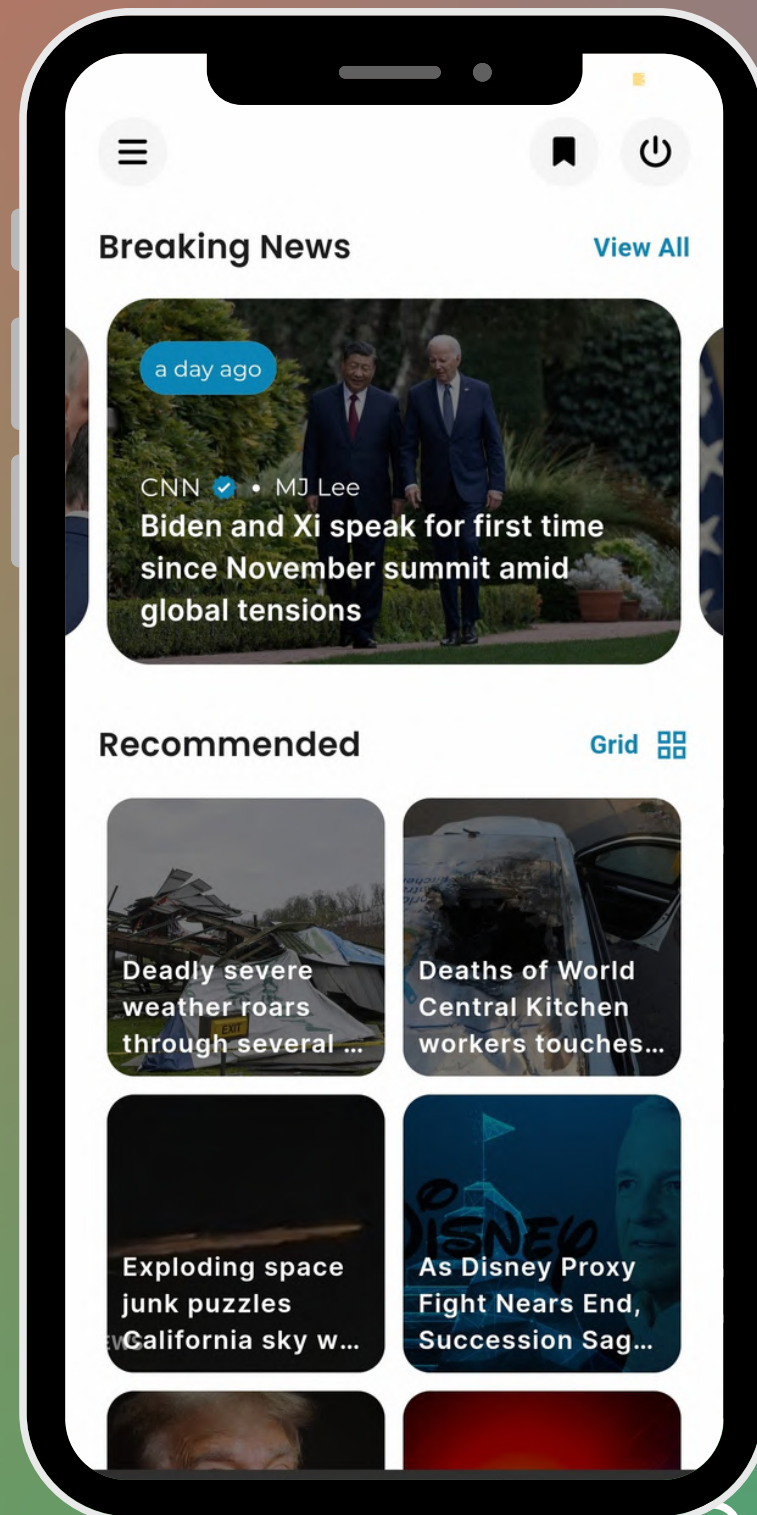


Firebase

IMPLEMENTATION



IMPLEMENTATION



CONCLUSION



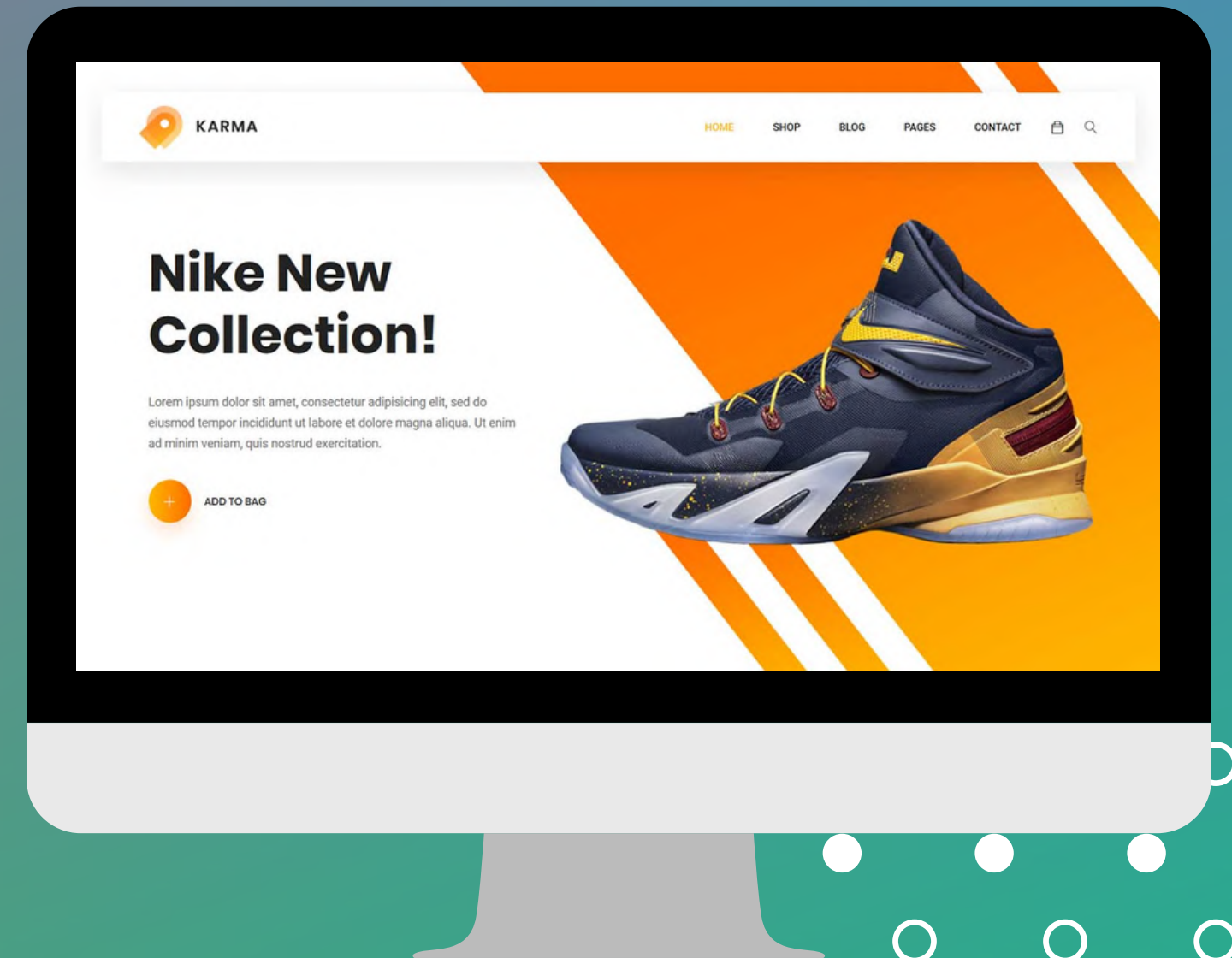
NewsNest redefines news consumption by offering personalized updates, seamless access, and Bookmarking. Through Flutter and Firebase, we've created a platform that empowers users to stay informed effortlessly. Aim is to simplify news consumption.



ECOMMERCE PWA



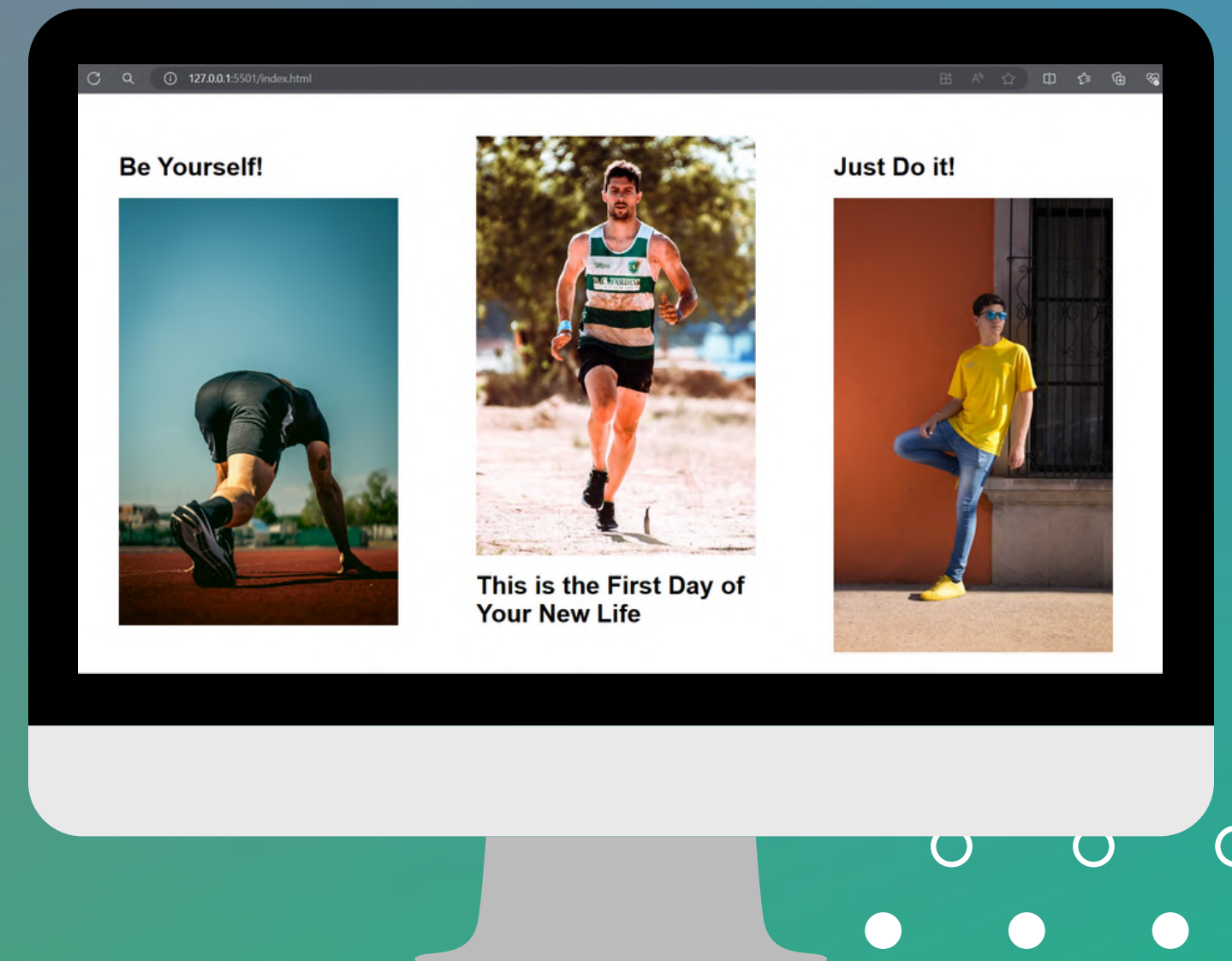
ENHANCING EXPERIENCE WITH PWA'S



INTRODUCTION



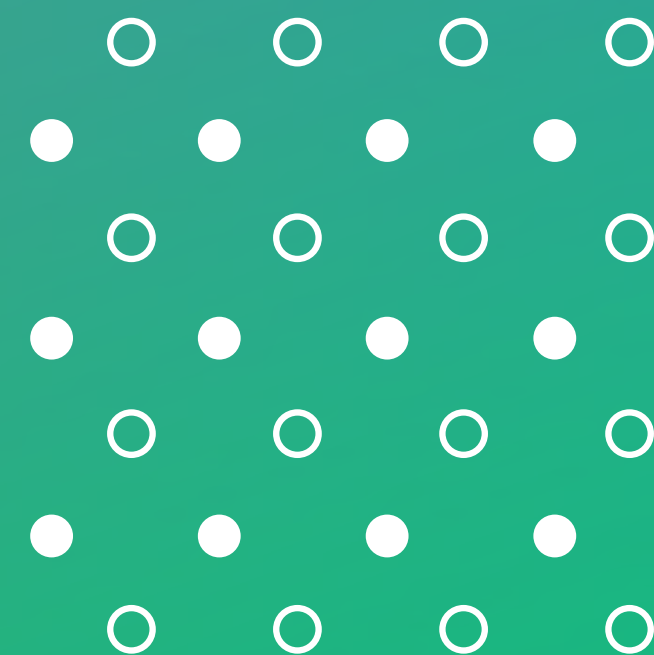
In today's mobile-first world, users expect fast, reliable, and engaging online shopping experiences. Progressive Web Apps (PWAs) provide a solution by bridging the gap between websites and native apps. This presentation will guide you through the key steps to build a powerful e-commerce PWA, enhancing your online store's functionality and user experience.)



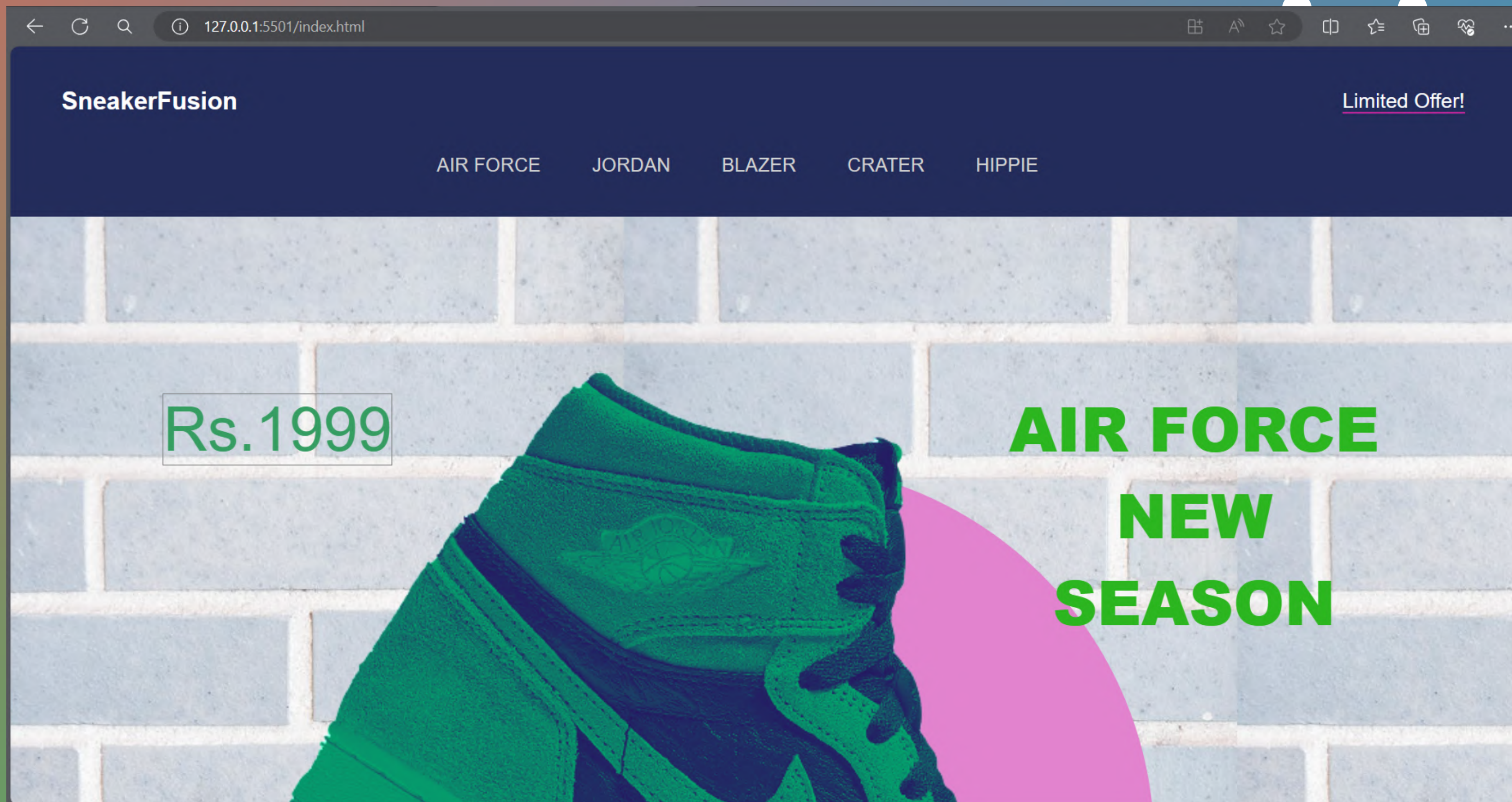
ESSENTIAL FEATURES OF PWA



1. **Add to Homescreen:** Users can add the PWA to their device's homescreen with just a tap, enabling easy access and enhancing user engagement.
2. **Offline Functionality:** PWAs utilize service workers to cache content, allowing users to access certain features and content even without an internet connection,
3. **Push Notifications:** PWAs can send push notifications to users, keeping them informed and engaged with timely updates and notifications.
4. **Fast Loading Speeds:** PWAs are designed to load quickly and respond faster than traditional web applications, providing users with a seamless and responsive experience.



IMPLEMENTATION



IMPLEMENTATION



manifest.json

```
{ } manifest.json > ...
1  {
2    "name": "SneakerFusion",
3    "short_name": "Shivani's App",
4    "start_url": "index.html",
5    "display": "standalone",
6    "background_color": "#232c5b",
7    "theme_color": "#000000",
8    "scope": ".",
9    "description": "Your ultimate destination for trendy sneakers!",
10   "icons": [
11     {
12       "src": "img/icon-192x192.png",
13       "sizes": "192x192",
14       "type": "image/png",
15       "purpose": "any maskable"
16     },
17     {
18       "src": "img/icon-512x512.png",
19       "sizes": "512x512",
20       "type": "image/png",
21       "purpose": "any maskable"
22     }
23   ]
24 }
25
```

file Structure

▼ ECOMM

>  .vscode

>  img

JS app.js

 index.html

{ } manifest.json M

JS script.js

JS serviceworker.js

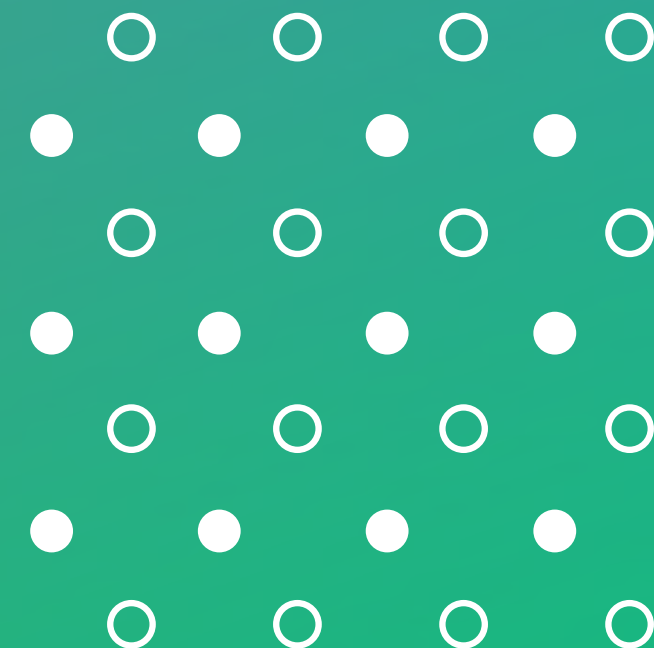
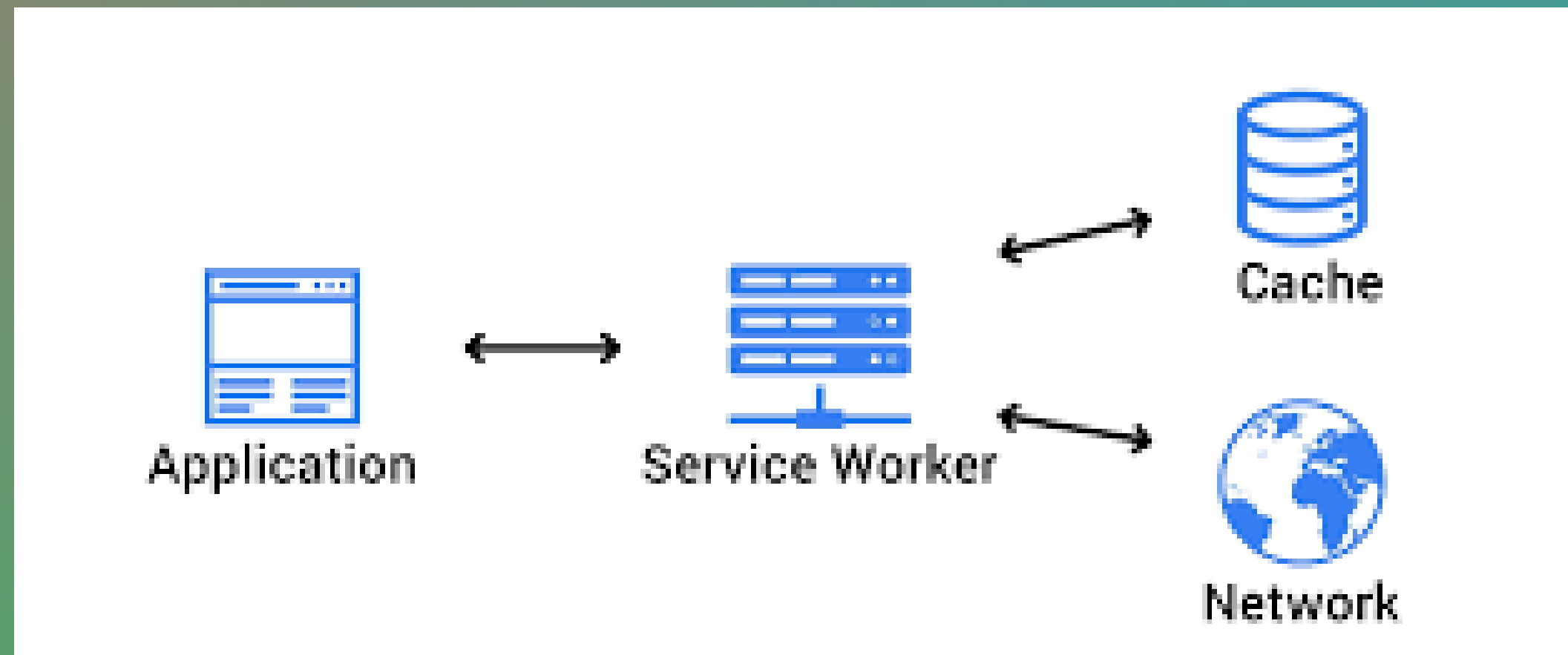
 style.css

JS sw.js

SW EVENTS



Service Worker Events: Service workers are scripts that run in the background of a web application, separate from the main browser thread. They intercept network requests and enable features such as offline functionality, push notifications, and background synchronization. Service workers listen for various events triggered by the browser or the web application, allowing them to execute code and respond accordingly.



FETCH EVENTS



The fetch event allows a service worker to intercept network requests made by your web app. This enables functionalities like caching frequently accessed resources or handling offline scenarios by serving cached responses.

The screenshot displays the Chrome DevTools interface for a service worker. On the left, the 'Storage' panel is expanded, showing various storage mechanisms like Local storage, Session storage, IndexedDB, Web SQL, Cookies, Private state t, Interest group, Shared storage, and Cache storage. Below it, the 'Background services' panel shows a progress bar. The main area is titled 'Clients' and shows the URL 'http://127.0.0.1:5500/ focus'. It includes controls for 'Push' (with a JSON object), 'Sync' (with a text input 'test-tag-from-devtools'), and 'Periodic Sync' (also with 'test-tag-from-devtools'). An 'Update Cycle' table shows the lifecycle of the service worker.

Version	Update Activity	Timeline
#43	Install	
#43	Wait	
#43	Activate	<div></div>

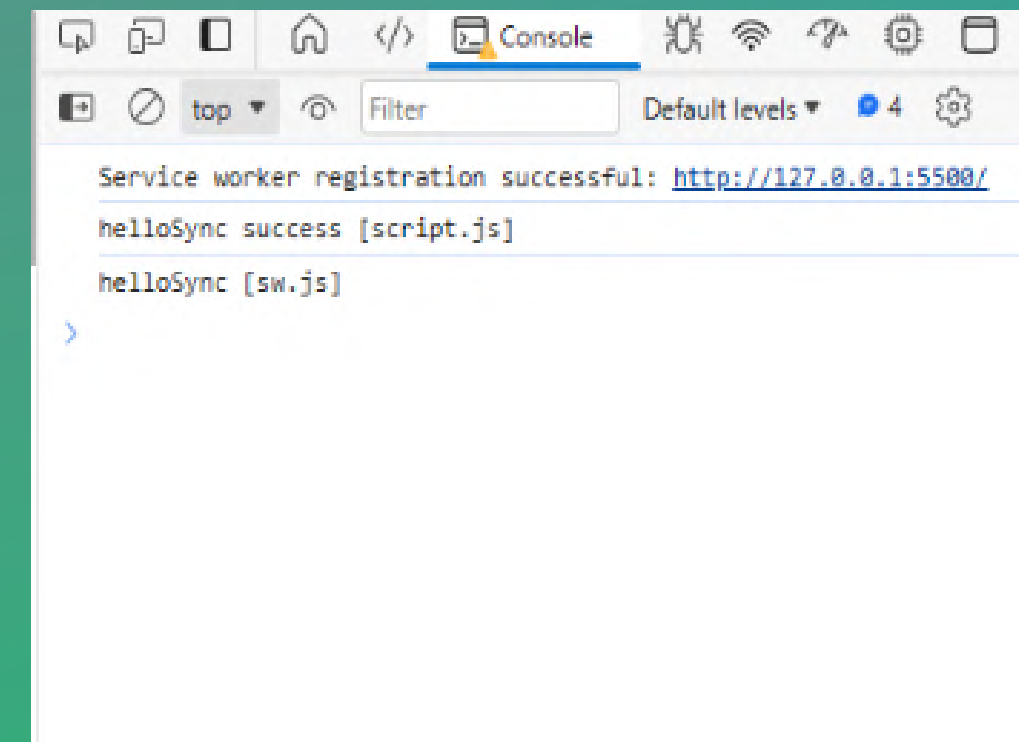
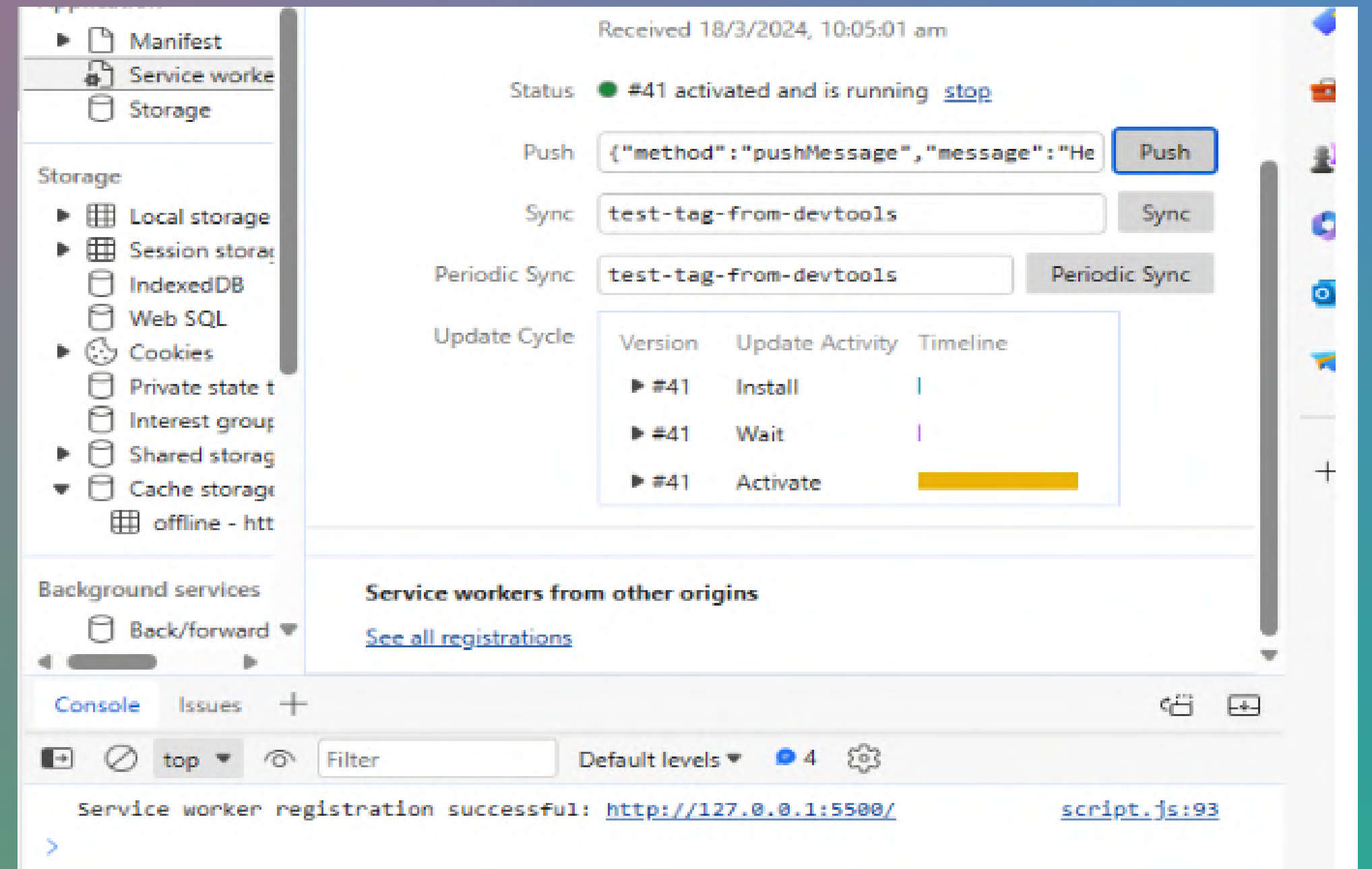
Below the update cycle, there's a section for 'Service workers from other origins'. At the bottom, the 'Console' panel shows four log entries:

- 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

SYNC EVENTS

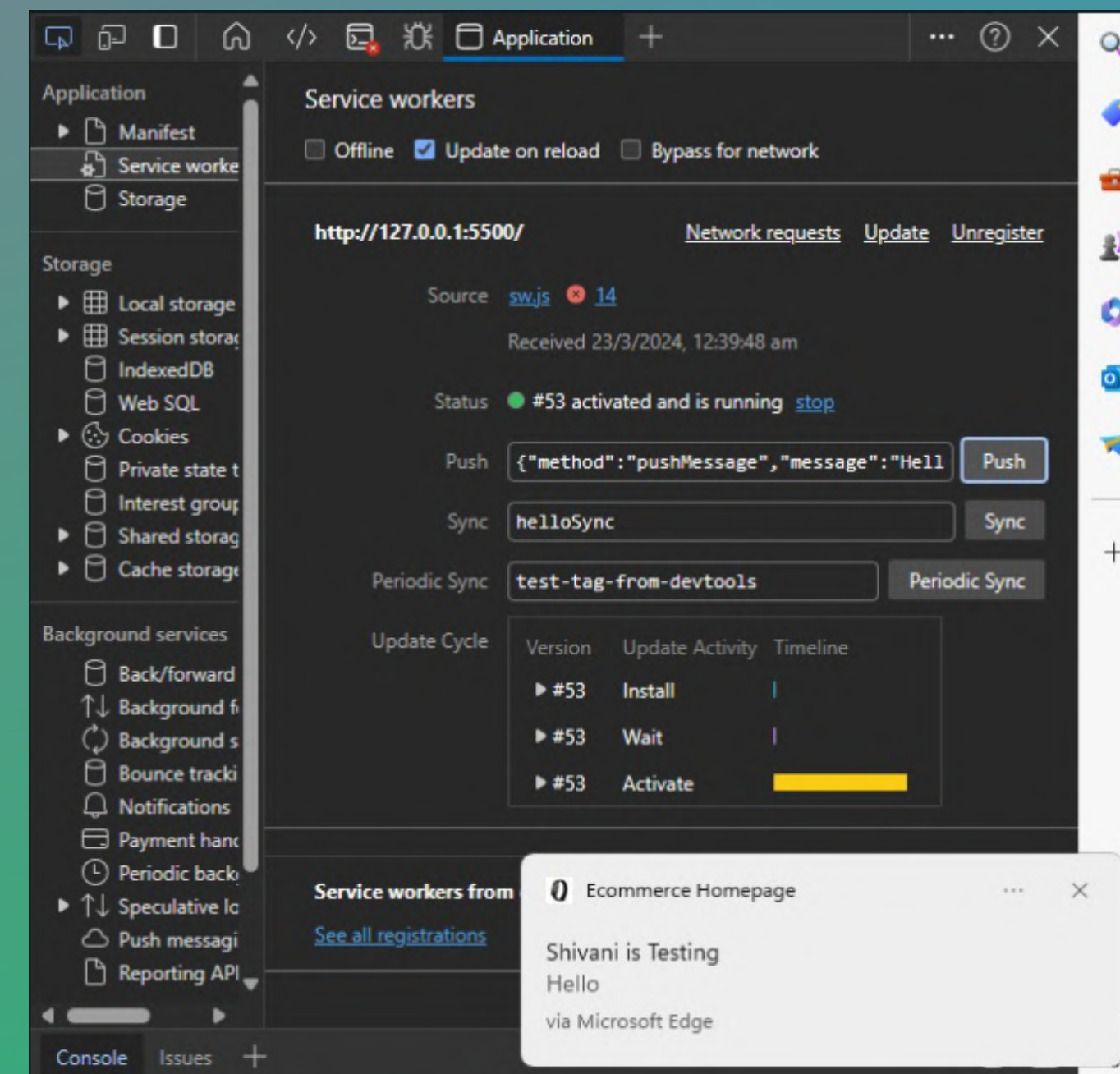
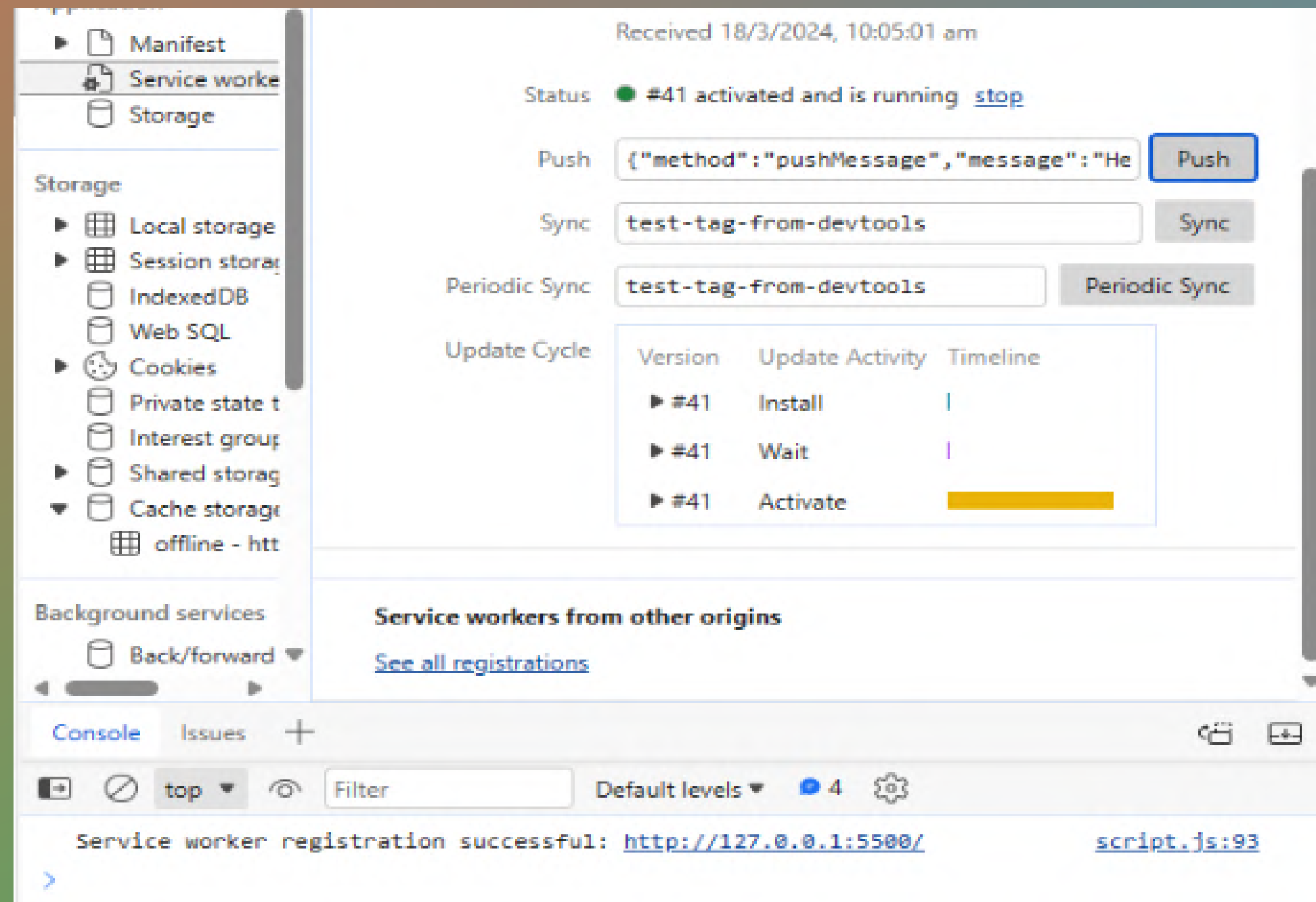


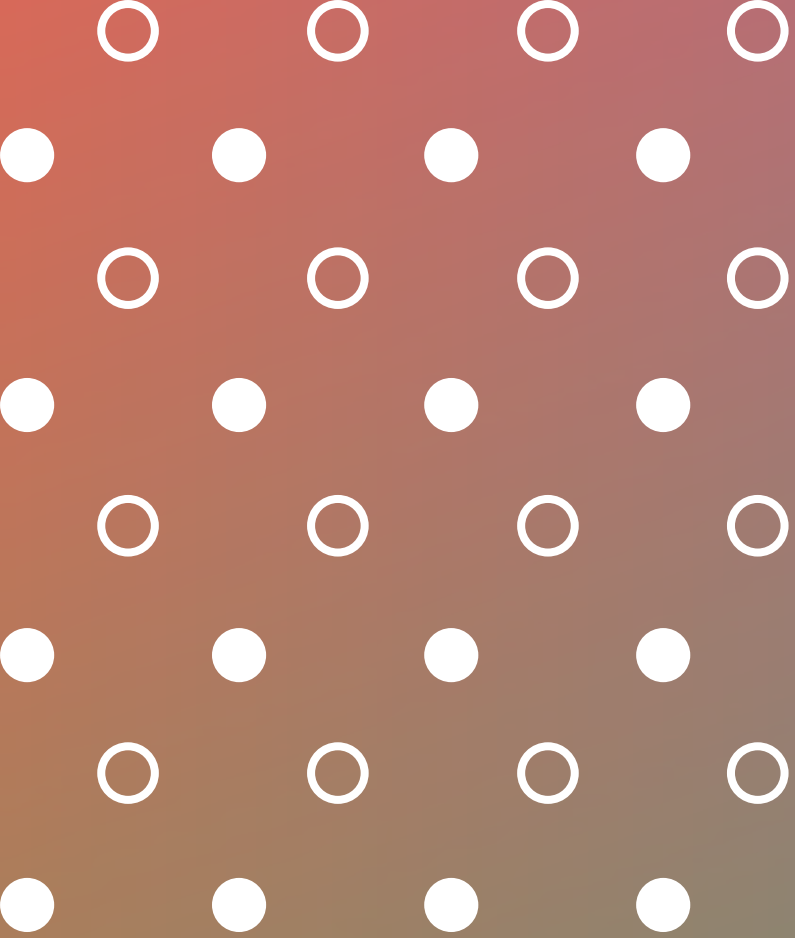
The Sync Event is triggered when the browser detects an internet connection after offline usage. It allows the service worker to synchronize data with the server, update the application's state, and execute queued background tasks, ensuring seamless data synchronization.



PUSH EVENTS

The Push Event occurs when a web application receives a push notification from a server or push service. It allows the service worker to display notifications, update the application's user interface, or execute other actions in response to real-time updates, enhancing user engagement and enabling timely communication even when the web application is not actively open.





THANKYOU :)

...

