Guddu Collection PWA

Developed by:

Gourav Khator

University Roll No.: 13000117104

Table of Contents

Guddu Collection Progressive Web App

Project Name:

Guddu Collection Progressive Web App

Project Description:

A progressive web app for showcasing Men and Women Fashion Wear. The webapp is installable on the device, to give a look of the native application.

Languages/Tools Used:

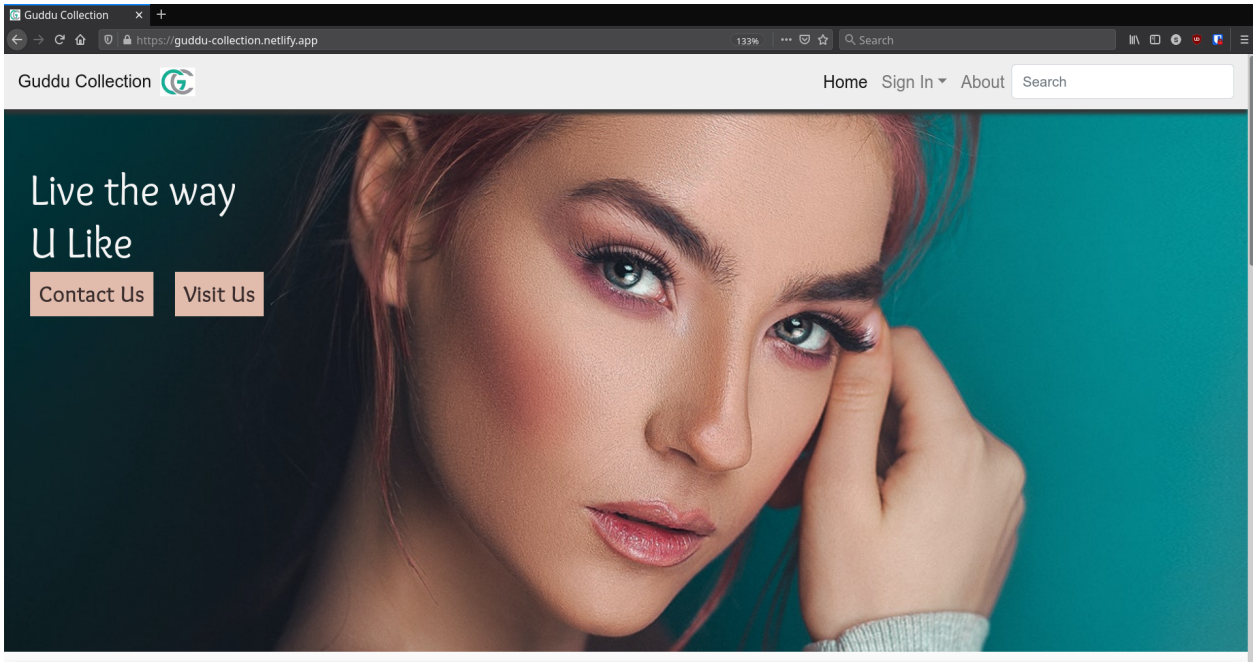
a) JavaScript
b) ReactJS [1] – A Frontend Framework written in JavaScript
c) Firebase [2] – A NoSQL Database developed by Google.
d) Bootstrap [3] – A CSS toolkit to provide pre-built stylesheets.
e) Netlify [4] – A web hosting platform for Deployment
f) Version Control – Git, with the remote repository hosted publicly on Github [5].

Features Available:

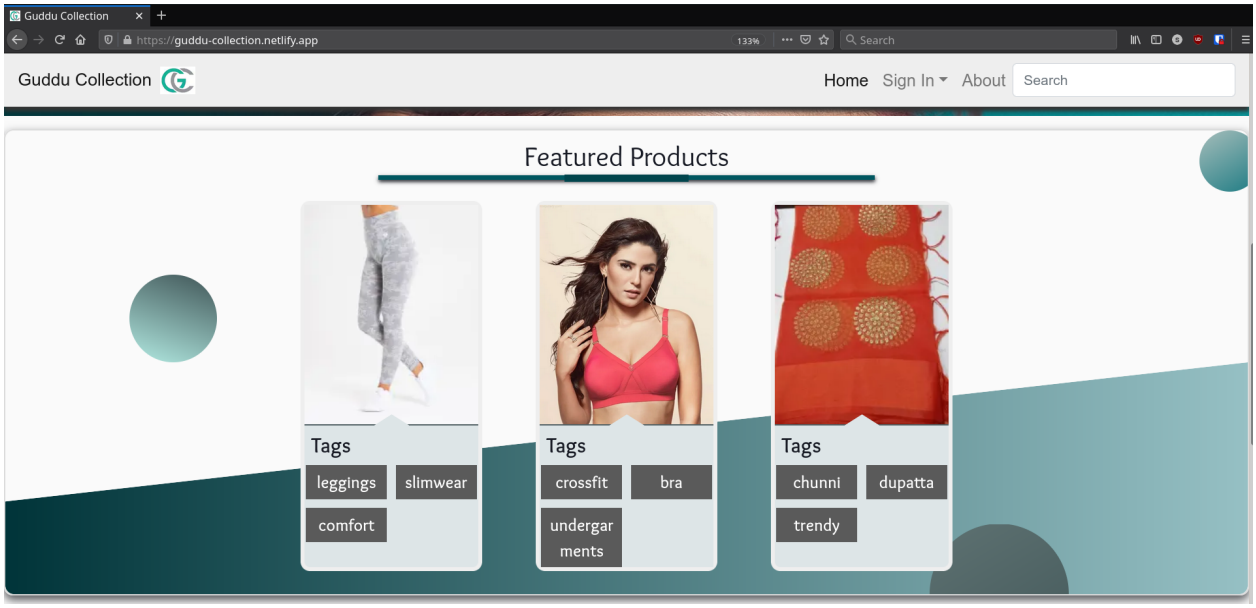
a) Installable Natively
b) Light weight
c) Search through featured and categorically attractive products.
d) Login and Signup with Google or the legacy email/password login.
e) Serverless Architecture using ReactJS for client side and Firebase for authentication and storage.

Project Demo Images:

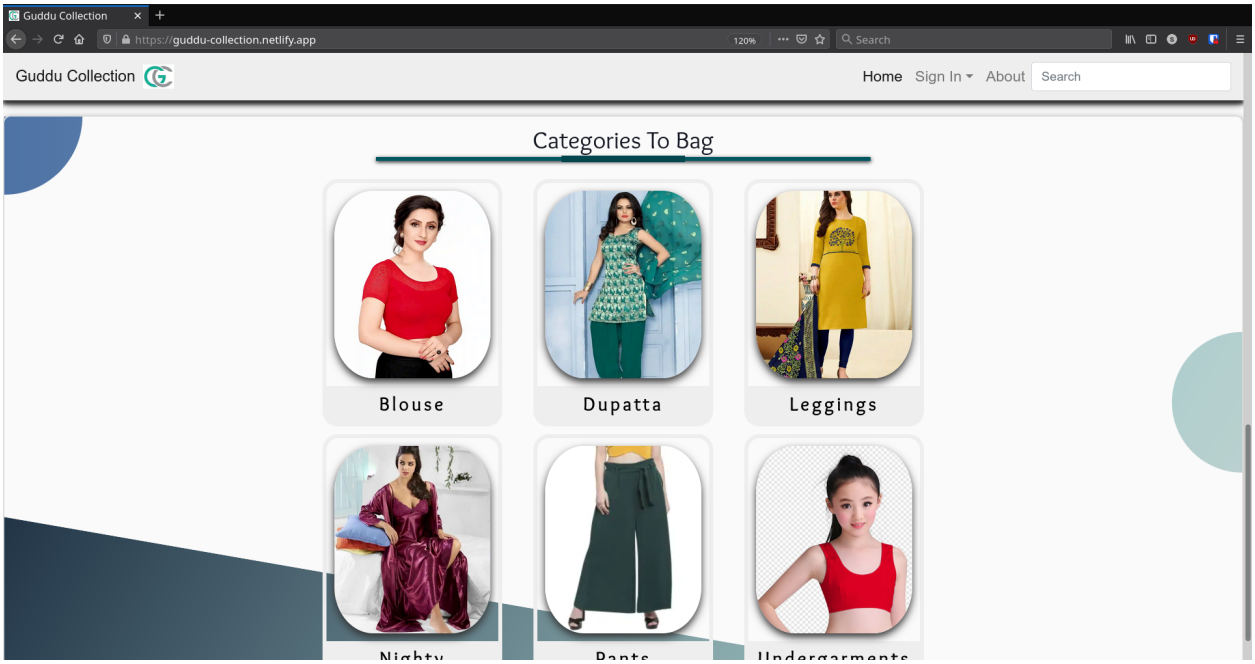
Home Page:



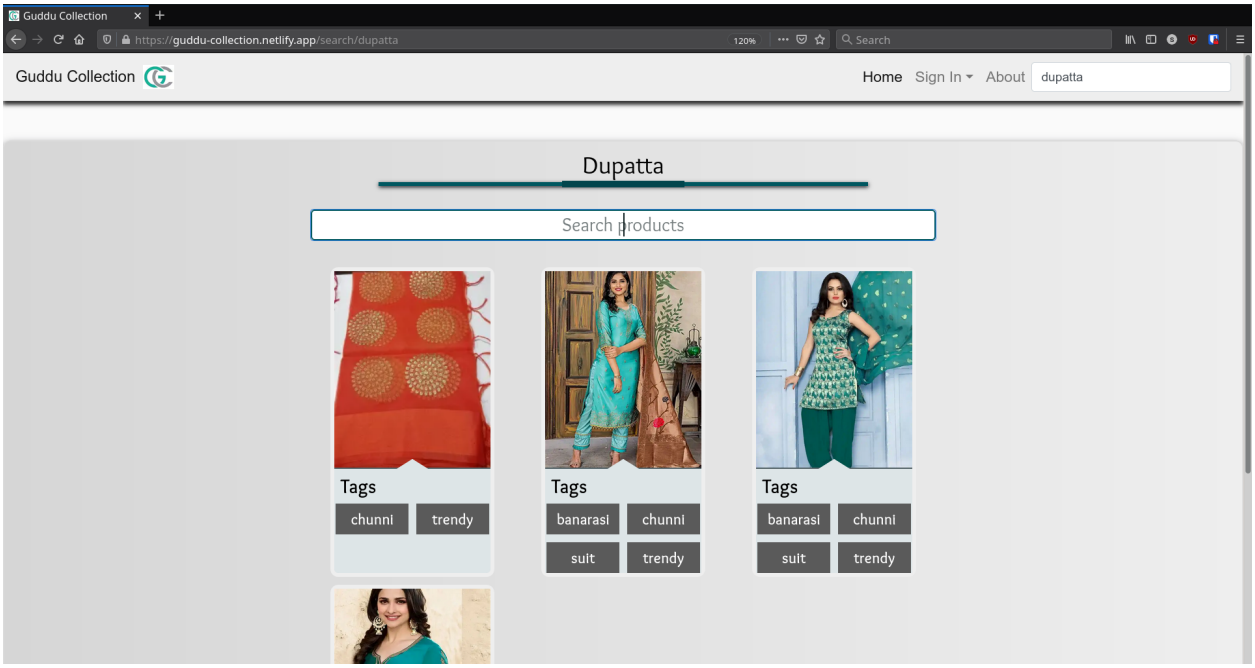
Featured Products Section:



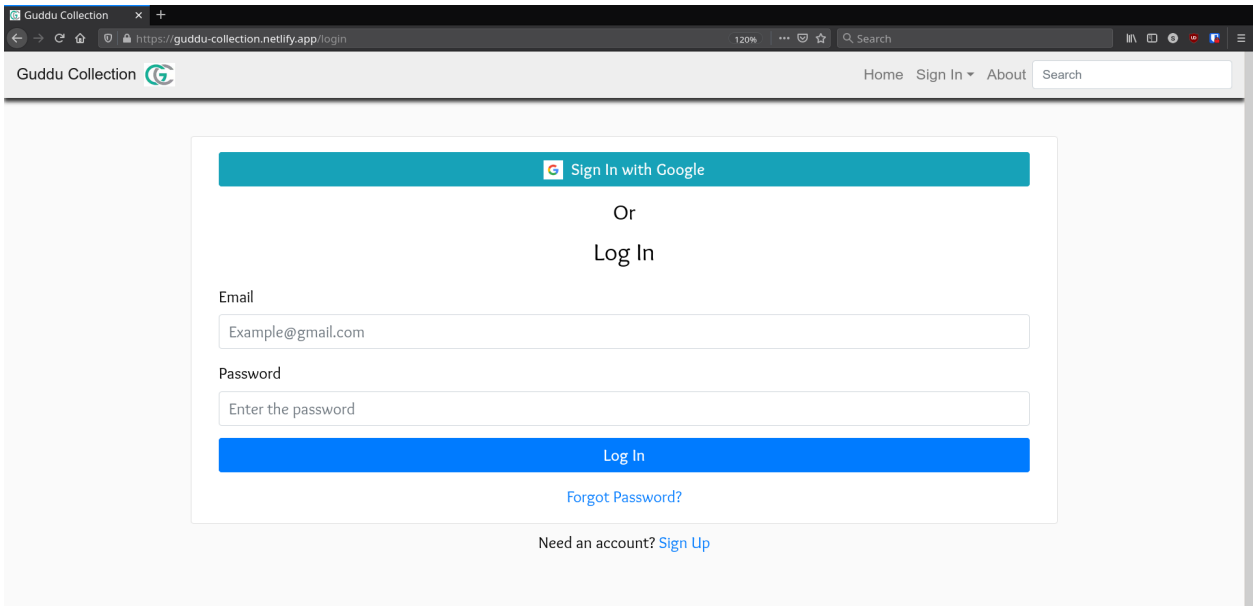
Categories To Bag Section:



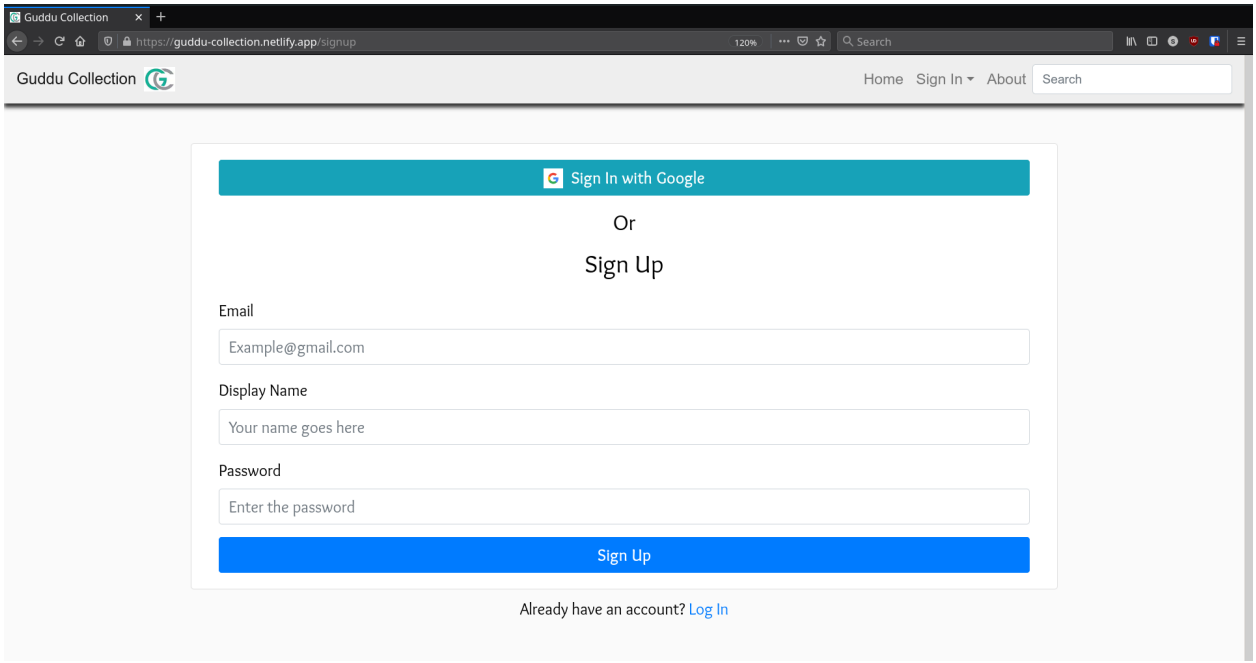
Search Page:



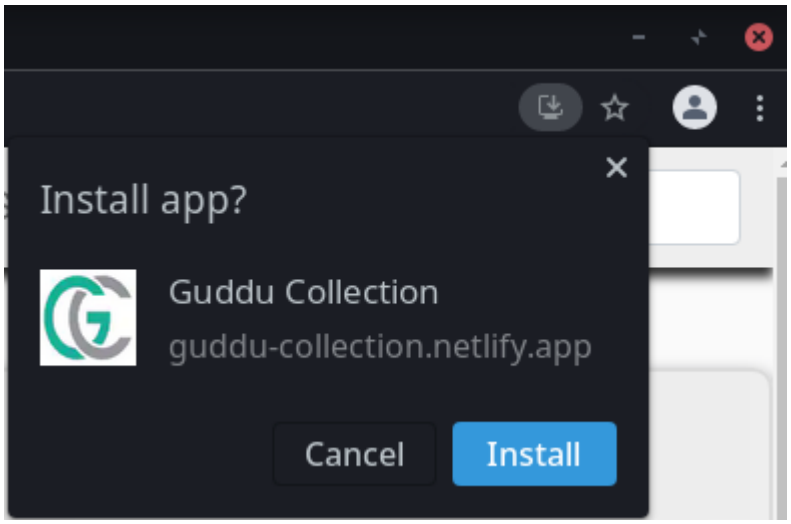
Login Page:



SignUp Page:



Progressive Web App Installable Demo:



Project Usage:

Live Demo is available and deployed at Guddu Collection.

To build and deploy yourself, just go to the Github Repository, clone the repository, and follow the below steps.

Initial Setup:

Add the environment variables, for all firebase API keys, in a file named as .env.local and place that file in the root folder of the project.

Then run:

```
npm install
```

This command installs the package dependencies locally.

Run the Project Locally on a Development Server:

```
npm start
```

Build the project to get the bundles for deployment:

```
npm run build
```

Serve the build/bundles on a local server:

```
npm run serve
```

Deploy the webapp on Netlify using Continuous Integration with Github:

Integrate the github repository with netlify in the netlify CI page. Add all the environment variables in the netlify admin page.

When there will be push to the linked github remote repository, the website will be automatically built in netlify and deployed within 2 minutes.

Why PWA?

Generally, webapps can only be accessed online, but if the user faces internet difficulty, then the user cannot enjoy the content from our webapp. This issue can be solved by making a native app. But, then we have to make separate native apps for **Android** and **IOS** platforms. This is where the PWA comes into handy.

PWA [6] viz. Progressive Web App, is a concept to store static and some dynamic content and files to the browser cache. *When the user sends a request for a particular resource, the request can be intercepted and instead of fetching files from the server, the webapp will fetch files from the browser cache. It will also download the latest version of files, if any, from the server and store them in the browser cache.* This way, on next user visit, the webapp will load the new files from the cache.

This whole process of intercepting request and sending a modified response can be handled by **Service Workers [7]**.

Future Scope

There's a scope for next versions of this webapp. This include:

a) Generating the apk from the pwa version using Bubblewrap [8] npm library.

References

The project was made possible by some libraries, tools, frameworks and concepts which are linked down below.

1) ReactJS - https://reactjs.org/
2) Firebase - https://firebase.google.com/
3) Bootstrap - https://getbootstrap.com/
4) Netlify - https://www.netlify.com/
5) Github - https://github.com/
6) PWA - https://developers.google.com/codelabs/pwa-training/pwa03--going-offline#0
7) Service Workers - https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API
8) BubbleWrap - https://www.npmjs.com/package/@bubblewrap/cli