

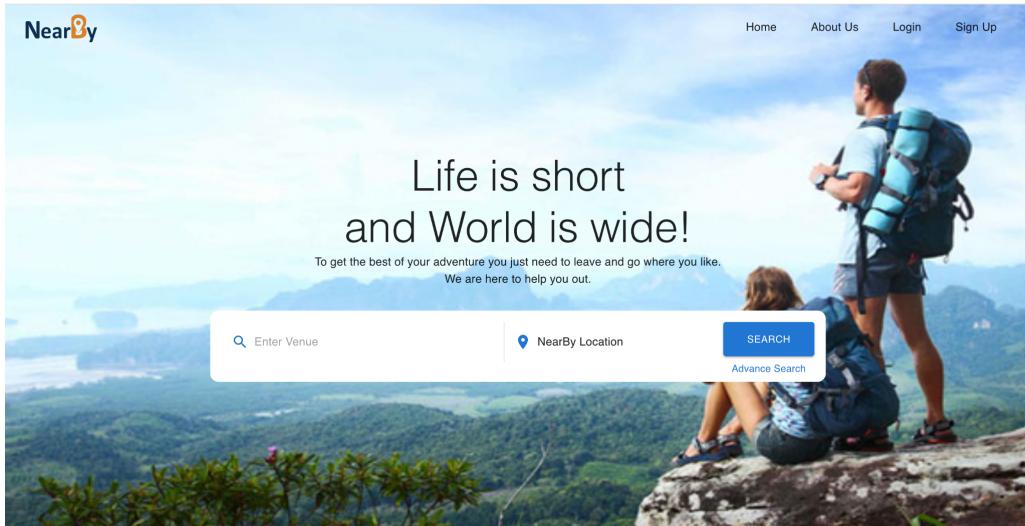


MERN STACK WEB APPLICATION

INCREMENT 4 REPORT

CSEE 5590 - WEB & MOBILE PROGRAMMING

12 • 13 • 2021



TEAM 2: THE PIONEERS

VINEETH REDDY SHERI
KALYANI NIKURE
YOUSEF ALMUTAIRI
PIYUSH NARHIRE

TABLE OF CONTENTS

OUR STORY	3
Features offered	3
THE DATA	4
THE APPLICATION	6
Home Page	6
Search Functionality	7
Venue Details Page	9
About Us	10
Login Page	10
Sign Up Page	12
THE WORKFLOW	13
FINAL DEPLOYMENT	14
IMPROVEMENT FROM THE PREVIOUS INCREMENT	15
IMPORTANT CODE SNIPPETS FROM THE PROJECT	15
COMMONLY ASKED QUESTIONS	22
How we are different from Google Maps?	22
How Foursquare City Guide is different from our application?	22
Is Foursquare API is related to the Application Database?	22
MEMBER CONTRIBUTIONS	22
CHALLENGES	23
APPLICATION WEBSITE	23
THE GITHUB REPOSITORY	23
THE PROJECT VIDEO	23
THE PRESENTATION LINK	23
CONCLUSION	23
REFERENCES	24

OUR STORY

Our focus is to help people with the best way to enjoy travel adventures by finding the NearBy attractions for them. We design and implement a creative solution to the everyday problems of users who love to travel and explore places.

Our intent is to develop a MERN stack web application for finding places nearby and presenting them to its users. Our application aims to satisfy the needs of those who are looking for a simple solution for searching and saving the places they wish to visit. Any newcomer to the place would find this application useful for exploring nearby locations and managing in the profile these places for future reference.

We would like to stay connected with our users on social media ([Facebook](#), [Instagram](#), [Twitter](#)) to share new updates and understand more about their feedback and user experience. We are ecstatic to have this rare opportunity to meet and share experiences with like-minded people interested in exploring this world adventure with The Pioneers Team.

Features offered

- Single Page Application (SPA) with responsive web design
- A variety of places are presented to the user as a result of their search query
- Advanced search options that include subcategories and a mile range
- Provides detailed information about the locations searched
- Next Venues and Similar Places suggestions with complete detailed information on each venue
- The Favorites tab that allows registered users to save liked places
- Video tutorials to help users learn how to use the application
- Quality of the overall user experience

THE DATA

Data management is a very crucial part of any web application. We want to protect users' information and also offer them the best information of their interest on the UI.

Ideally, we'd like to save the login information of registered users as well as their favorite places in a NoSQL-like MongoDB database. This is an open-source database that stores data in JSON-like format.

All the information presented to users on the venues is fetched from the Foursquare API calls through our application.

The usage details of the database and Foursquare API is discussed below table:

DB / API Element	Parameters	Use in Application
Collection - Users	Collection fields: { _id, firstname, lastname, emailAddress, password, city, state, zipcode, verificationCode, createdAt, updatedAt }	We use users collection to store users' information provided by them with each user registration. This collection is also referred at the time of user login authentication. Once the user provides credentials on the User Interface, they are validated against the information in this collection. If these details match, the user is redirected to the further session.
Collection - Favorites	Collection Fields: { _id, emailAddress, favourites: {Array:Venue IDs}, createdAt, updatedAt }	This Collection holds the information of favorited places by any registered user on the platform. The Array of Venue IDs is stored uniquely mapped with the username (email) of any

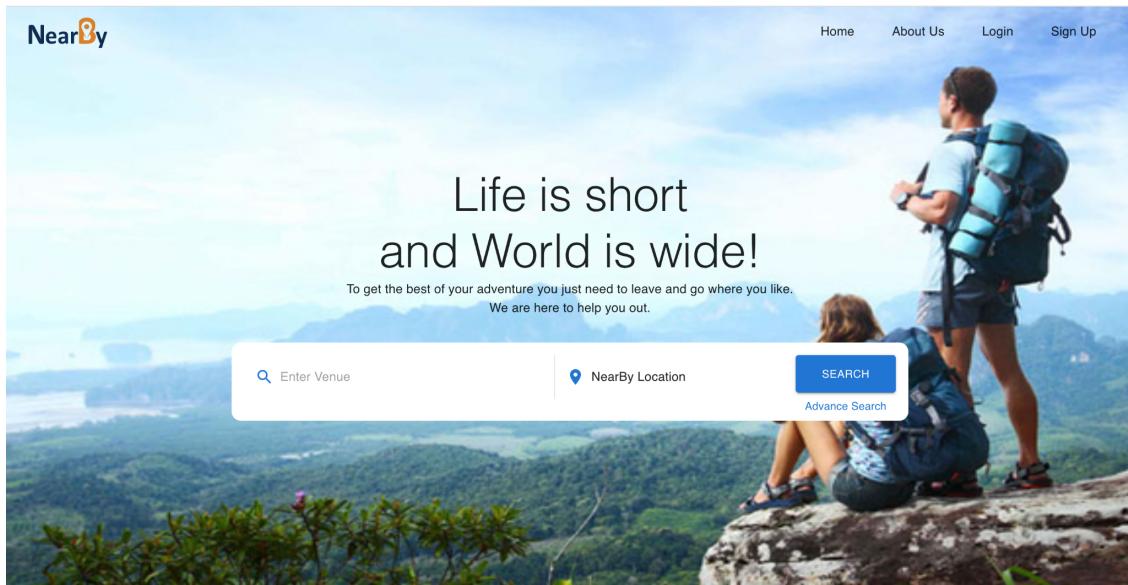
		particular user along with timestamp information.
Collection - Userverifications	<p>Collection Fields</p> <pre>{ _id, emailAddress, verificationCode, createdAt }</pre>	<p>This collection comes into play when any user clicks on Forgot Password link and provides an email address to send instructions.</p> <p>The collection then stores the verification code which is also sent to the user for authentication and resetting the password. Later, the application verifies this code from the database to authorize reset password activity.</p> <p>Since the code is only valid for 4 minutes, the record gets removed after its expiry.</p> <p>The record expiry is mentioned in the collection schema to ensure the automatic deletion of the record.</p>
API- Foursquare	<p>API Endpoints used</p> <p>https://api.foursquare.com/v2/venues/VENUE_ID</p> <p>https://api.foursquare.com/v2/venues/categories</p> <p>https://api.foursquare.com/v2/search/autocomplete</p>	<p>We utilized various endpoints offered through Foursquare API to provide information about places on the user interface.</p> <p>Endpoint /venues/VENUE_ID provides information of a specific venue.</p> <p>Endpoint /venues/categories get the list of categories and are populated in the search dropdown.</p> <p>Endpoint /venues/autocomplete will return all the possible categories and names of the venue based on your "search" string. Every time you enter a letter in the text field, the</p>

	<p>autocomplete API will be called and displayed on the screen for the user to select. If the search string itself is empty will display the general categories such as top picks, food, etc.</p> <p>https://api.foursquare.com/v2/venues/explore</p> <p>Endpoint /venues/explore is called when the user selects any of the categories from the search (for example food category). The API will return all the venues that are related to food. This API will be called only when the user selects a category or search string. But not when the user selects the Venue directly in the search option.</p> <p>https://api.foursquare.com/v2/venues/VENUE_ID/nextvenues</p> <p>Endpoint /venues/VENUE_ID/nextvenues provides a list of Next Venues related to selected Venue ID.</p> <p>https://api.foursquare.com/v2/venues/VENUE_ID/similar</p> <p>Endpoint /venues/VENUE_ID/similar helps to get the list of similar locations in relation to the selected venue.</p>
--	---

THE APPLICATION

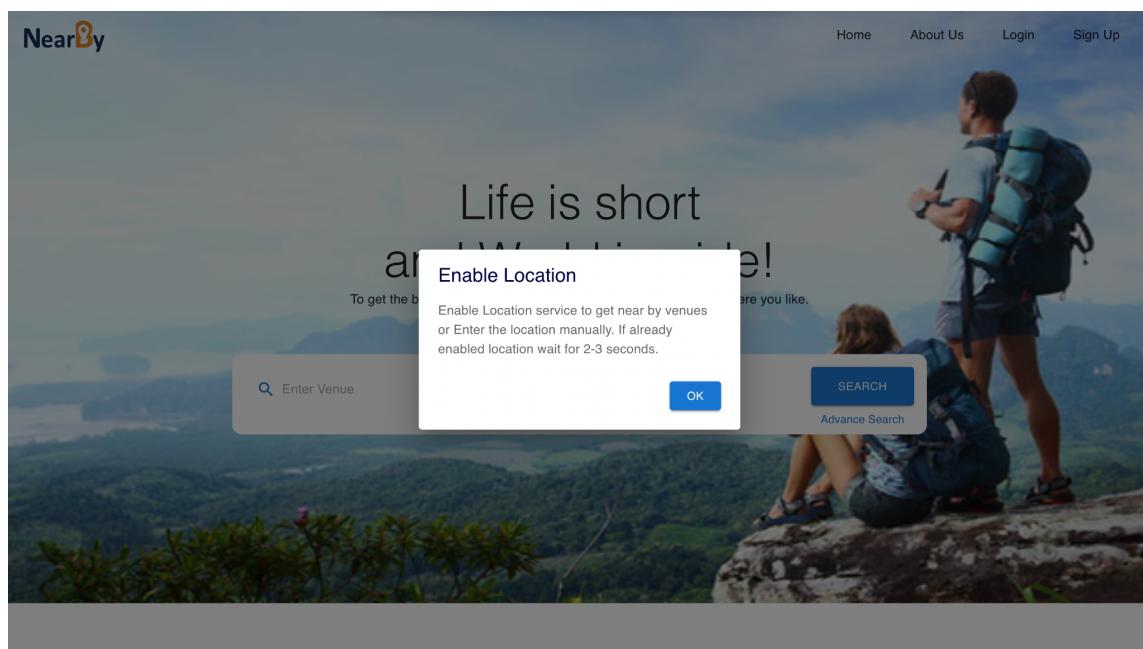
1. Home Page

When the user clicks on the web link of our application, he or she will be taken to the Home page where the application will ask the user to enable the location permission. The screen below shows our default home page's navigation bar, which includes Home, About Us, Login, and Sign Up. Users can enter specific search criteria to find the desired venue, and the results will appear on the screen. The user can keep up with us by using the icons in the footer to follow us on social media.



2. Search Functionality

If the user tries to enter the venue without giving a location, a 'Enable Location' dialog will appear; if the user does not wish to provide permission to access the current location, he/she must enter the location and continue the search. In addition, when a user authorizes permission to access their location, the NearBy location tab takes the user's current location in order to provide results in the user's general neighborhood.



If the location is provided, the user can search by entering a venue name or selecting a category from the drop-down options. Advanced search allows users to narrow down their search by selecting a category, sub-category, and range of miles. For example, if the category is food, the sub-category is a Mexican restaurant, and the range is 15 miles, the app will display the results accordingly.

3. Venue Details Page

On the Venue Details Page the user will be able to read more about the venue's details such as a summary of the venue, a link to the venue's website, directions to the location (Google Map), recent user reviews, next venues, and similar locations. We're using the Foursquare API to get information about venue lists that match the search criteria of our users. Foursquare provides information such as the venue's name, address, popularity, images, tips, category, and a summary. Some of the endpoints used are venue/venue_id, /categories, /autocomplete, /explore, /nextvenues, and /similar.

The screenshot shows the venue details page for 'Fiorella's Jack Stack Barbecue'. At the top, there are navigation links: Home, About Us, Login, and Sign Up. Below that is a 'Save' button with a heart icon. The main content area features three images: a large sign that reads 'Fiorella's JACK STACK BARBECUE', a view of the restaurant interior with tables and chairs, and another view of the interior. Below the images, the venue's name 'Fiorella's Jack Stack Barbecue' is displayed with a rating of 9. A link to the website and a location pin icon are also present. At the bottom, there is a section titled 'About the Location' with a paragraph of text and a 'Reviews' section featuring two user reviews with profile pictures and names.

This screenshot shows the same venue details page for 'Fiorella's Jack Stack Barbecue' but with a different layout. It includes a 'NearBy' logo at the top left and navigation links: Home, About Us, Login, and Sign Up. The venue's name and rating are at the top. Below is a 'Website Link' and 'Location' button. The 'About the Location' section is present. The 'Reviews' section is expanded, showing two reviews with user profiles ('The P' and 'Tony J') and their comments. The overall design is cleaner and more focused on the reviews section.

4. About Us

We are a team “The Pioneers” studying at the University of Missouri - Kansas City and we are working on our web application “Nearby”. Our focus is to help people with the best way to enjoy travel adventures by finding the NearBy attractions for them. Our product aims to satisfy those who are looking for a simple solution for searching and saving the places they wish to visit.

This page helps users to know more about us. This tab helps users with a [User Guide Video](#) which tutors them with application usage with a demo of the overflow.

The screenshot shows the 'About Us' section of the NearBy website. At the top, there is a navigation bar with links for Home, About Us, Login, and Sign Up. Below the navigation, there is a message: 'interested in exploring this world adventure with The Pioneers Team.' A note encourages users to tag '#NearBy'. The main content area is titled 'OUR TEAM' and features a heading 'The Pioneers' with the subtext 'Dedicated Team. For Your Dedicated Dreams'. Four team members are listed in boxes: Kalyani Nikure (Backend), Yousef Almutairi (Backend), Piyush Narhire (Frontend), and Vineeth Reddy (Sheri). At the bottom, there is a dark footer bar with social media icons for Facebook, Instagram, and Twitter, along with links for 'Terms of use | Privacy Policy' and '© 2021 NearBy'.

5. Login Page

If the user already has an account with us, they can log in to our application by entering their username (email) and password. If the user enters an incorrect username or password, an error message displays, stating that either the username or password is incorrect and that the user should try again. User information is crucial to us in order for us to keep users' data private and protected.



Login

Welcome to NearBy

Enter the credentials to access your account

Remember me

[Forgot Password?](#)

[SIGN IN](#)

Don't have an account yet? [Register here.](#)

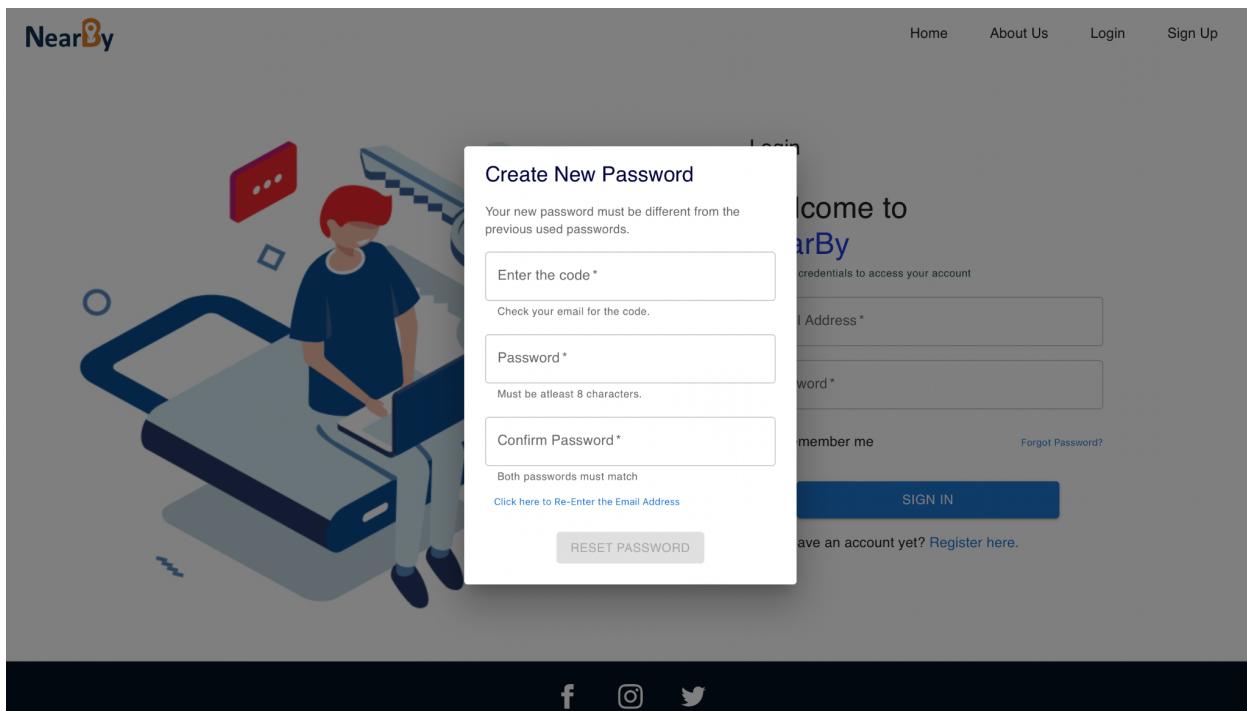
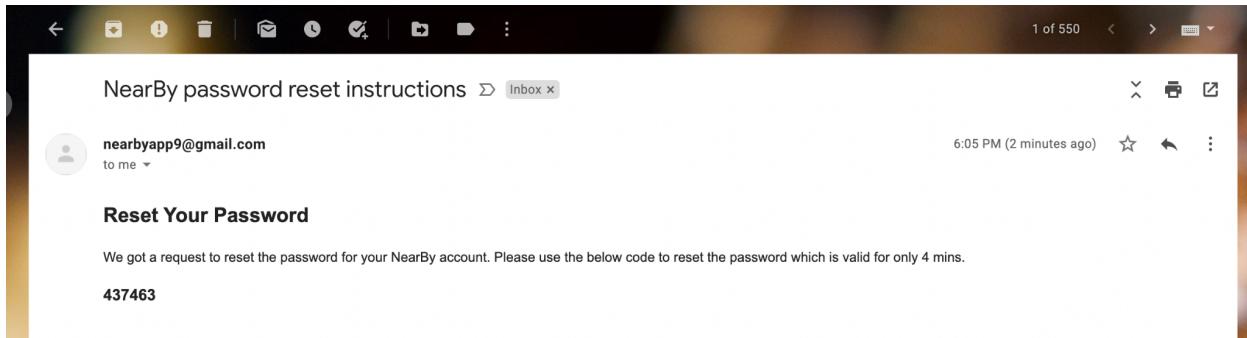


If a user forgets their password, we provide a forgot password feature. The user will receive an email with password reset instructions and a code for verification which allows them to change their password and try logging in again. The code will be valid for 4 minutes only. After logging in, customers can save their favorite venues so that they can visit them in the future by opening our app and accessing them.

Forgot Password

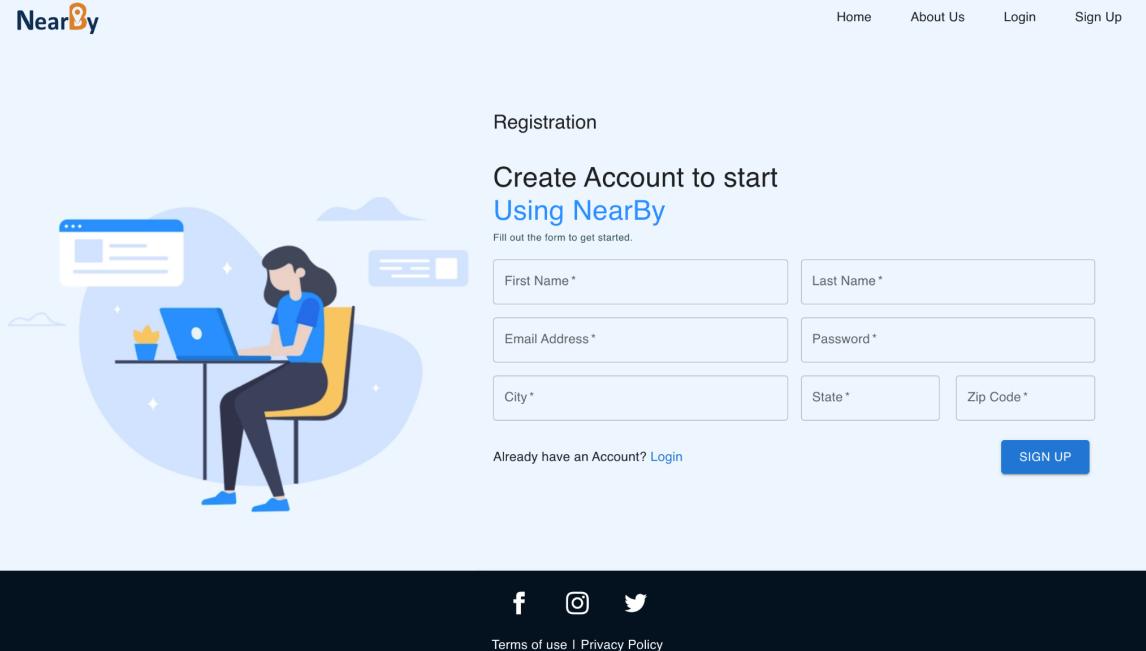
Enter the email associated with your account and we'll send an email with instructions to reset your password.

SEND INSTRUCTIONS



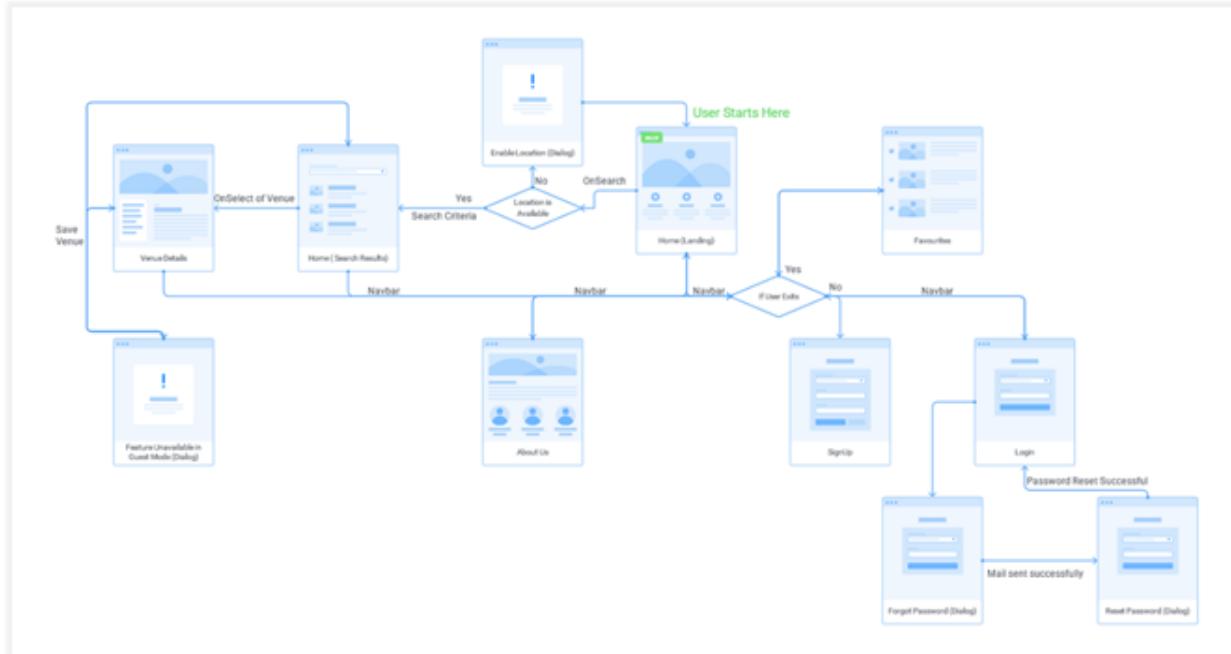
6. Sign Up Page

We've simplified the sign-up procedure so that users just have to provide the bare minimum of information in order to register an account with us and make use of all of the services. Also, we ensure that the proper data is being recorded in the database by activating Client-Side Error Messages Handling. For example, the length of the password string does not satisfy the minimum criteria, the format of your email address is incorrect, or mandatory fields are missing. To favorite a location and save it for a later visit, users must first create an account on our web application.



THE WORKFLOW

The below diagram covers the application workflow and user redirection routes with various user activities and scenarios. For a better quality vision of the below workflow diagram [click here](#).



FINAL DEPLOYMENT

We have used the Firebase platform to deploy our web application. We have successfully deployed the front end. The "Hosting" feature in Firebase CLI was used to deploy all files from our local machine to the hosting server. In contrast, we used the "Functions" feature and upgraded our Firebase plan to "Blaze" in order to deploy the back-end of our web application. Although we have followed all the steps outlined on the Firebase website, this step has not been implemented and the files have not been uploaded to the function cloud. Therefore, we tried different cloud hosting services (including AWS, Azure, and Heroku) and finally developed the backend changes using Heroku.

Using the Procfile, we can run the Express Server application as shown below in the screenshot. [Click Here](#) to access the deployed version of the application.

The screenshot shows the Firebase Project Overview dashboard for a project named 'NearBy'. The left sidebar contains navigation links for Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning, Release & Monitor, and Extensions. It also indicates a 'Spark' plan with a free \$0/month usage and an 'Upgrade' button. The main dashboard area has tabs for 'Dashboard' (selected) and 'Usage'. A 'Cloud Logging' section with a 'Get started' button is present. Below it, the 'nearby-cb4a0 domains' section lists two domains: 'nearby-cb4a0.web.app' (Status: Default) and 'nearby-cb4a0.firebaseio.com' (Status: Default). A blue 'Add custom domain' button is visible. At the bottom, the 'nearby-cb4a0 release history' section shows a single entry: a 'Current' deployment by user 'yalmutairi2030@gmail.com' (ID: 0b5d73) on Dec 12, 2021, at 10:40 PM, with 36 files.

IMPROVEMENT FROM THE PREVIOUS INCREMENT

We strive to improve our application in every possible way. As part of phase 4, we did minor changes to our User Interface to provide an enhanced user experience to our users. Till now we were hosting our project locally and to make it publicly available we finally deployed the final version of our application on the server using Firebase and Heroku.

IMPORTANT CODE SNIPPETS FROM THE PROJECT

Application Routing navigation between different screens.

```
<Header />
<Switch>
  <Route exact path="/">
    <Redirect to="/home" />
  </Route>
  <Route exact path="/login">
    <Login />
  </Route>
  <Route exact path="/signup">
    <Signup />
  </Route>
  <Route exact path="/home">
    <Home />
  </Route>
  <Route exact path="/aboutUs">
    <AboutUs />
  </Route>
  <Route exact path="/venue/:id">
    <VenueDetails />
  </Route>
  <Route exact path="/favourites">
    <Favourites />
  </Route>
</Switch>
<Footer />
```

To get venue/categories suggestion based on the typed search string in the Search Text Filed.

```
getSuggestions = () => {
  const FOURSQUARE_SUGGESTION_BASE_URL =
    "https://api.foursquare.com/v2/search/autocomplete?";

  let params = {
    client_id: "H5EETN0LSQSI5BI543FMCNKXV1EYWJFI2FVW3JRGPRSVS4UK",
    client_secret: "MIXINUYYB1S2TB542ZYEA5U1HXS1VC0HNTR41DKR3CLB34CG",
    group: "unified",
    query: this.state.venueSearchTerm,
    v: 20211002,
    limit: 100,
    radius: 20000,
  };
  if (this.state.location === "NearBy Location") {
    params.ll = this.state.latitudeLongitude;
  } else {
    params.near = this.state.location;
  }

  try {
    axios
      .get(FOURSQUARE_SUGGESTION_BASE_URL + new URLSearchParams(params))
      .then((res) => {
        this.setState({
          showSuggestion: true,
          suggestions: res.data.response.groups[0].items,
        });
      });
  } catch (err) {}
};
```

To get detailed information of a specific venue

```
getVenueDetails = async () => {
  const FOURSQUARE_VENUE_BASE_URL = "https://api.foursquare.com/v2/venues/";

  let params = {
    client_id: FourSquareCredentials[0].client_id,
    client_secret: FourSquareCredentials[0].client_secret,
    VENUE_ID: "",
    v: 20211002,
  };
  try {
    let venueList = this.state.venueList;
    this.state.exploreVenueList.forEach((venue, idx) => {
      if (idx >= (this.state.noOfVenues - 10) && idx < this.state.noOfVenues) {
        params.client_id = FourSquareCredentials[idx % 4].client_id;
        params.client_secret = FourSquareCredentials[idx % 4].client_secret;
        params.VENUE_ID = venue.venue.id;

        let currentVenueURL =
          FOURSQUARE_VENUE_BASE_URL + venue.venue.id + "?";
        axios
          .get(currentVenueURL + new URLSearchParams(params))
          .then((res) => {
            venueList.push(res.data.response.venue);
            this.setState({ venueList });
          });
      }
    });
  } catch (err) {}
};
```

Adding the venue to the favorite list

```
addToFavouriteList = () => {
  if (!this.props.cookies.get("emailId")) {
    this.setState({ openLoginDialog: true });
  } else {
    try {
      axios
        .post("http://localhost:3001/addFavourite", {
          emailAddress: this.props.cookies.get("emailId"),
          venueId: this.props.venue.id,
        })
        .then((res) => {
          if (res.data.status === "success") {
            this.props.addVenueToFavourite(this.props.venue.id, res);
          }
        });
    } catch (err) {}
  }
};
```

Remove the venue from the venue list

```
removeFromFavouriteList = () => {
  try {
    axios
      .post("http://localhost:3001/removeFavourite", {
        emailAddress: this.props.cookies.get("emailId"),
        venueId: this.props.venue.id,
      })
      .then((res) => {
        if (res.data.status === "success") {
          this.props.removeVenueFromFavourite(this.props.venue.id, res);
        }
      });
  } catch (err) {}
};
```

Resetting the password with a new one.

```
resetPwd = (code, pwd) => {
  try {
    axios
      .post("http://localhost:3001/resetPwd", {
        emailAddress: this.state.resetEmailAddress,
        code: code,
        password: pwd,
      })
      .then((res) => {
        if (res.data.status === "success") {
          this.closeResetPwd();
          this.setState({ resetEmailAddress: "" });
        }
        this.setState({ msgSB: res.data.response, openSB: true, typeSB: res.data.status });
      });
  } catch (err) {}
};
```

Sending the Forgot reset instruction to the email

```
sendPwdResetInstruction = (emailAddress) => {
  try {
    axios
      .post("http://localhost:3001/forgotPwdEmail", {
        emailAddress: emailAddress,
      })
      .then((res) => {
        if (res.data.status === "success") {
          this.closeForgotPwd();
          this.openResetPwdDialog();
          this.setState({ resetEmailAddress: emailAddress });
        }
        this.setState({ msgSB: res.data.response, openSB: true, typeSB: res.data.status });
      });
  } catch (err) {}
};
```

Validating the User credentials

```
validateCredentials = () => {
  if (
    this.state.emailAddress !== "" &&
    !this.state.errorStatus.emailAddress &&
    this.state.password !== ""
  ) {
    try {
      axios
        .post("http://localhost:3001/login", {
          emailAddress: this.state.emailAddress,
          password: this.state.password,
        })
        .then((res) => {
          if (res.data.status === "success") {
            let fullName =
              res.data.response.firstname + " " + res.data.response.lastname;
            this.props.cookies.set("name", fullName, { path: "/" });
            this.props.cookies.set([
              "emailId",
              res.data.response.emailAddress,
              { path: "/" }
            ]);
            this.state.rememberMe &&
              this.props.cookies.set(
                "rememberEmail",
                res.data.response.emailAddress,
                { path: "/" }
              );
            this.props.history.push("/home");
          } else {
            this.setState({ errMsg: res.data.error });
          }
        });
    } catch (err) {}
  }
};
```

Creating the User account

```
createAccount = () => {
  try {
    axios
      .post("http://localhost:3001/signup", {
        firstname: this.state.firstName,
        lastname: this.state.lastName,
        emailAddress: this.state.emailAddress,
        password: this.state.password,
        city: this.state.city,
        state: this.state.state,
        zipcode: this.state.zipCode,
      })
      .then((res) => {
        if(res.data.status === "success") {
          this.props.history.push("/login");
        } else {
          this.setState({ errMsg: res.data.error });
        }
      });
  } catch (err) {}
};
```

Getting the current geolocation of the User

```
getAndsetCurrentLocation = () => {
  navigator.geolocation.getCurrentPosition((loc) => {
    this.setState({
      geoLocation: loc,
      latitudeLongitude: loc.coords.latitude + "," + loc.coords.longitude,
      location: "NearBy Location"
    });
  });
};
```

COMMONLY ASKED QUESTIONS

1. How we are different from Google Maps?

We offer our users a rich set of features that are completely different from those offered by Google Maps. In every way, front-end and back-end development are different. The directions functionality of Google Maps is a part of our application, and we do not focus on navigation strictly from source to destination, but we are aiming to create a kind of bookkeeping application that users can save and refer to these locations for future use or for planning travel adventures. Additionally, we have created social media connections to allow users to connect with us and create a community of like-minded people.

2. How Foursquare City Guide is different from our application?

While Foursquare's City Guide application offers many of the same functionalities as our application, it does not provide as rich a user experience as our application. We also have social media platforms so that people can connect with us and share their experiences after using our application as well as their travel adventures.

3. Is Foursquare API is related to the Application Database?

To display information like similar locations, next venues, and a list of venues based on search criteria, we utilize the Foursquare API. We are NOT storing any information from the Foursquare API to the MongoDB database directly. Instead, we use our database to save information of our Users who register with us and also store favorite places of our users.

MEMBER CONTRIBUTIONS

We have distributed our project activities among our team members based on Back-end and Front-end application development. The front-end development was proactively handled by Vineeth Reddy Sheri and Piyush Narhire to give a better user experience.

Kalyani Nikure and Yousef Almutairi, on the other hand, worked on the back-end integration with the database and other tasks such as deployment activities. Documentation for the project was done collaboratively by everyone.

CHALLENGES

As with any other project, we faced multiple challenges in our project. One of the main challenges that we are facing every time is resolving the bugs and issues during the development process. These came as the static images are breaking while reloading in VenueDetails, Link Component Conflict between Material UI and React Router DOM due to the same naming convention, and UI scrolling issue, for instance, when the user scrolled down to the bottom of the homepage for example and want to go to about us page, it will start at the same point (bottom page)

In addition, FourSquare is providing 3 different packages for using their API. The free package that we are using for our project is limited to calling only 2 photos and 2 tips per venue which restricts us from providing more information.

APPLICATION WEBSITE

<https://nearby-cb4a0.firebaseio.com/home>

THE GITHUB REPOSITORY

<https://github.com/VINEETHREDDYSHERI/PlacesNearBy>

THE PROJECT VIDEO

<https://youtu.be/wWUkbqb8hjU>

THE PRESENTATION LINK

[https://docs.google.com/presentation/d/1qjo18053PehThTT9hpjsr7m8sdLIUdAD/edit?
usp=sharing&ouid=103738542969294587525&rtpof=true&sd=true](https://docs.google.com/presentation/d/1qjo18053PehThTT9hpjsr7m8sdLIUdAD/edit?usp=sharing&ouid=103738542969294587525&rtpof=true&sd=true)

CONCLUSION

This project aims to be part of people's travel adventures. We provide adequate and relevant information on the venues in order to provide travelers with the information they need. In order to get our users to the best places, we tried to match their preferences. Additionally, Next Venues suggests other potential destinations for users to visit. As well, they can select places that have similar specifications to those they are interested in from the similar location suggestions.

Moreover, since we place a high priority on the user experience, we strive to offer platforms that connect us to our users. For connecting with our users and sharing the latest updates and features, we have created social media accounts on Facebook, Twitter, and Instagram. This provides us with a unique opportunity to stay connected and also from a community of people who are interested in exploring the world together.

Although we tried to make the best of our efforts by creating this application for our users, we will be focusing our future efforts on addressing the challenges we have mentioned in the Challenges section and adding features that allow users to interact with the website at their convenience and get accurate information about places they would like to visit.

REFERENCES

1. <https://docs.mongodb.com/manual/core/data-model-design/>
2. <https://developer.foursquare.com/docs/places-api/>
3. <https://spectralops.io/blog/yelp-api-guide/>
4. <https://mui.com>
5. <https://reactjs.org>
6. <https://nodemailer.com/usage/>
7. <https://reactrouter.com/docs/en/v6>
8. <https://expressjs.com/en/starter/basic-routing.html>
9. <https://developer.foursquare.com/docs/places-api/endpoints/>
10. <https://medium.com/swlh/how-to-deploy-a-react-app-with-firebase-hosting-98063c5bf425>
11. <https://blog.logrocket.com/8-ways-to-deploy-a-react-app-for-free/>
12. <https://www.youtube.com/watch?v=IDHfvpsYShs>
13. <https://medium.com/@sandun.isuru/how-to-deploy-nodejs-express-app-to-firebase-as-function-31515c304e70>