Resort Website (Group-5)

By: Shreykumar Patel, Shivani Chaudhari, Radhesh Patel

Introduction

We have used HTML, CSS, Jquery, Firebase, Bootstrap, and the Google Cloud Platform for this project. We have also implemented various libraries that will be explained further in this presentation.

Overview of the technologies used:

HTML: HTML stands for Hyper Text Markup Language. This is the standard markup language for creating Web pages. The HTML code describes the structure of a Web page.

<tagname> Content goes here... </tagname>

CSS: CSS stands for Cascading Style Sheets. The specification describes how HTML elements should be displayed on screens, on paper, or in other media.

JavaScript: JavaScript was developed in order to provide a way for web pages to "become alive" or dynamic. Scripts are programs written in this language. They can be written directly in the HTML of a web page and run automatically when the page is loaded.

Overview

Jquery: jQuery is a fast, small, and feature-rich JavaScript library. It simplifies navigation and manipulation of HTML documents, event handling, animation, and Ajax using an API that works across a wide range of browsers.

A Brief Look

DOM Traversal and Manipulation

Get the `<button>` element with the class 'continue' and change its HTML to 'Next Step...'

```
$( "button.continue" ).html( "Next Step..." )
```

Event Handling

Show the `#banner-message` element that is hidden with `display:none` in its CSS when any button in `#button-container` is clicked.

```
var hiddenBox = $( "#banner-message" );
$( "#button-container button" ).on( "click", function( event ) {
  hiddenBox.show();
});
```

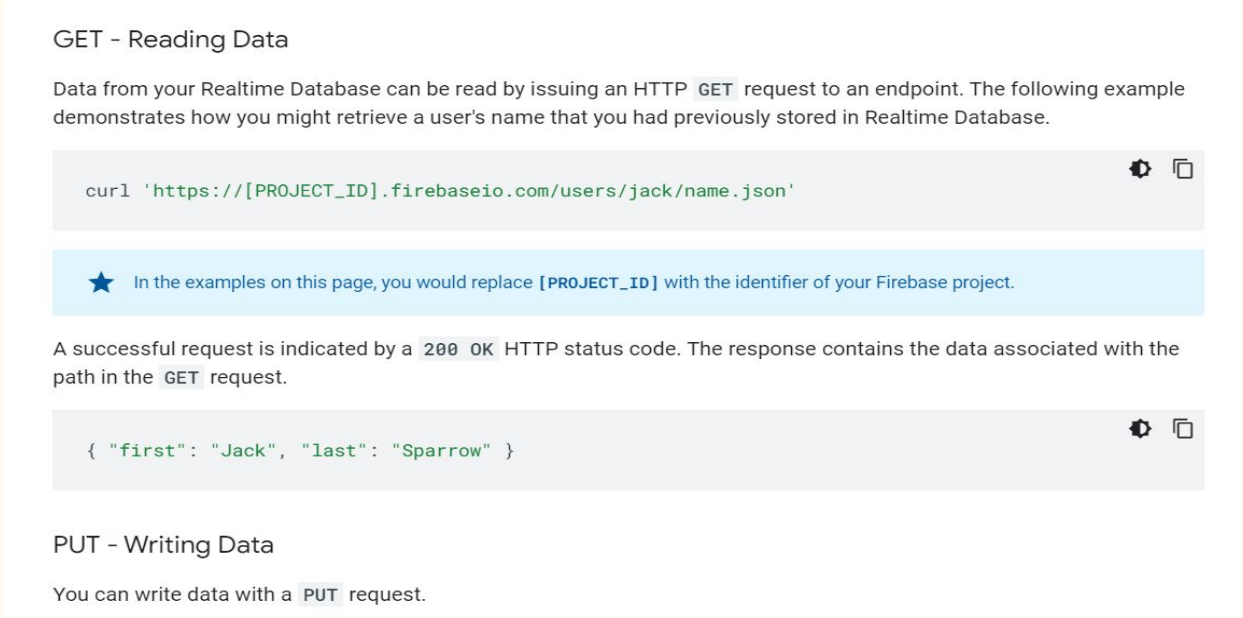
Ajax

Call a local script on the server `/api/getWeather` with the query parameter `zipcode=97201` and replace the element `#weather-temp`'s html with the returned text.

```
$.ajax({
  url: "/api/getWeather",
  data: {
    zipcode: 97201
  },
  success: function( result ) {
    $( "#weather-temp" ).html( "<strong>" + result + "</strong> degrees" );
  }
});
```

Overview

Firebase: Firebase Realtime Database is a cloud-based database. The data is stored as JSON and is synchronized in real-time across all connected clients.



GET - Reading Data

Data from your Realtime Database can be read by issuing an HTTP `GET` request to an endpoint. The following example demonstrates how you might retrieve a user's name that you had previously stored in Realtime Database.

```
curl 'https://[PROJECT_ID].firebaseio.com/users/jack/name.json'
```

In the examples on this page, you would replace `[PROJECT_ID]` with the identifier of your Firebase project.

A successful request is indicated by a `200 OK` HTTP status code. The response contains the data associated with the path in the `GET` request.

```
{ "first": "Jack", "last": "Sparrow" }
```

PUT - Writing Data

You can write data with a `PUT` request.

Overview

Bootstrap:

- Bootstrap is a front-end framework that makes web development faster and easier
- This framework includes HTML and CSS design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and other features, as well as optional JavaScript plugins.
- Additionally, Bootstrap makes it easy to create responsive designs

Google Maps API: The Maps JavaScript API lets clients customize maps with their own content and imagery for display on web pages and mobile devices. There are four basic map types offered by the Maps JavaScript API (roadmap, satellite, hybrid, and terrain), which can be modified using layers, styles, controls, and events, as well as various services and libraries.

Netlify: For hosting the website we have used netlify. Netlify provides continuous deployment services and is free for hosting.

Technology resources and libraries used:

Font-awesome: https://cdnjs.com/libraries/font-awesome

Swiper: https://swiperjs.com/changelog

Bootstrap 4.4.1: https://getbootstrap.com/docs/4.4/getting-started/download/

Firebase Database : https://www.gstatic.com/firebasejs/9.15.0/firebase-database.js

EmailJS : https://cdn.jsdelivr.net/npm/@emailjs/browser@3/dist/email.min.js

Google map API : https://console.cloud.google.com/google/maps-apis/studio/styles?project=gapi-372101

Design

Link to documentation and starter tutorials:

HTML:

Starter tutorial: https://www.w3schools.com/html/

Documentation : https://developer.mozilla.org/en-US/docs/Web/HTML

CSS:

Starter tutorial: https://www.w3schools.com/css/

Documentation: https://developer.mozilla.org/en-US/docs/Web/CSS

JavaScript:

Starter tutorial: https://www.w3schools.com/js/

Documentation: https://devdocs.io/javascript/

Firebase:

Starter tutorial: https://www.tutorialspoint.com/firebase/index.htm

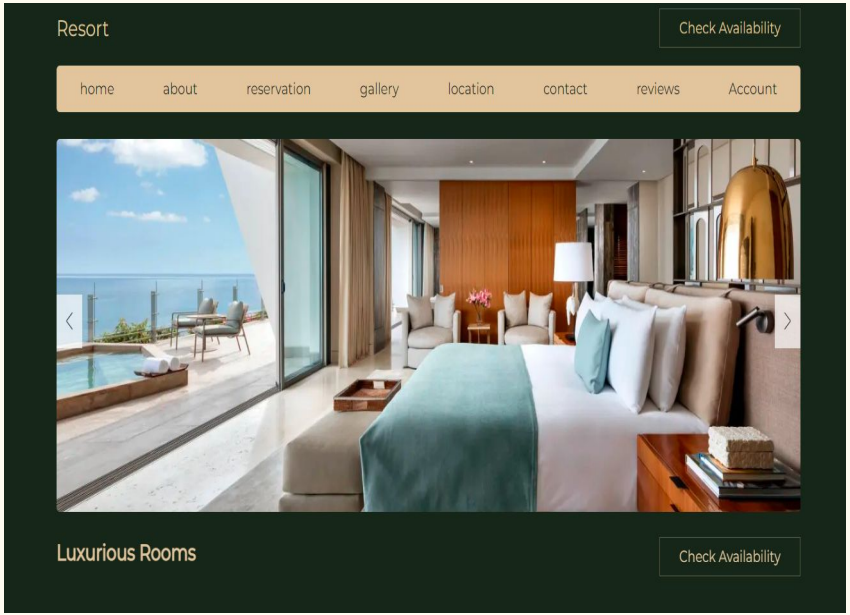
Documentation: https://firebase.google.com/docs/functions/get-started

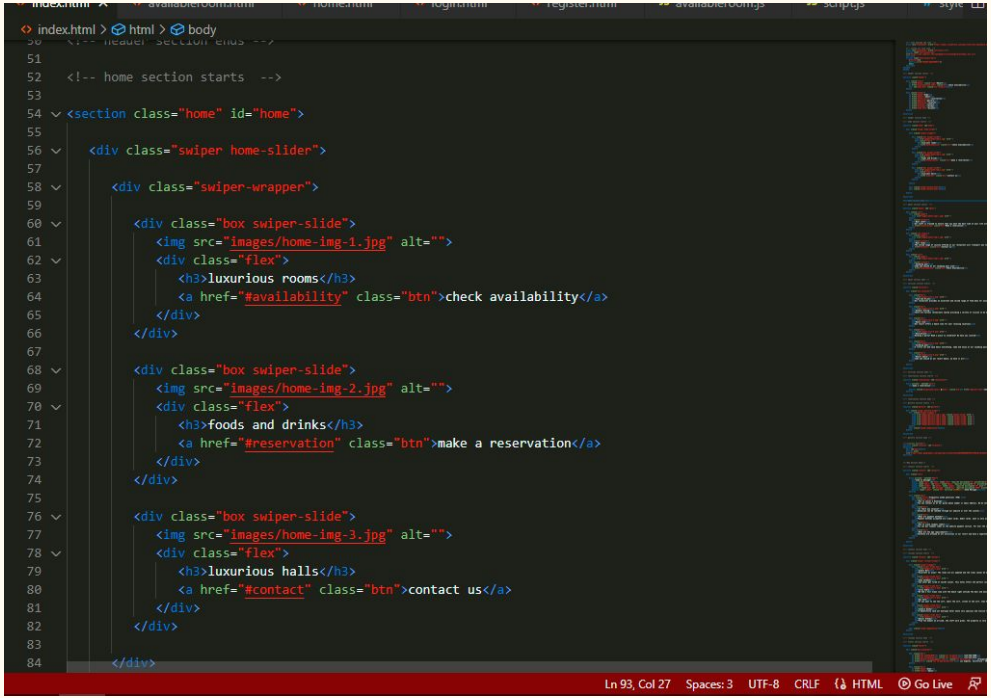
Google Maps API:

Starter tutorial: https://console.cloud.google.com/home/dashboard?project=gapi-372101&walkthrough_id=assistant__webhosting__index

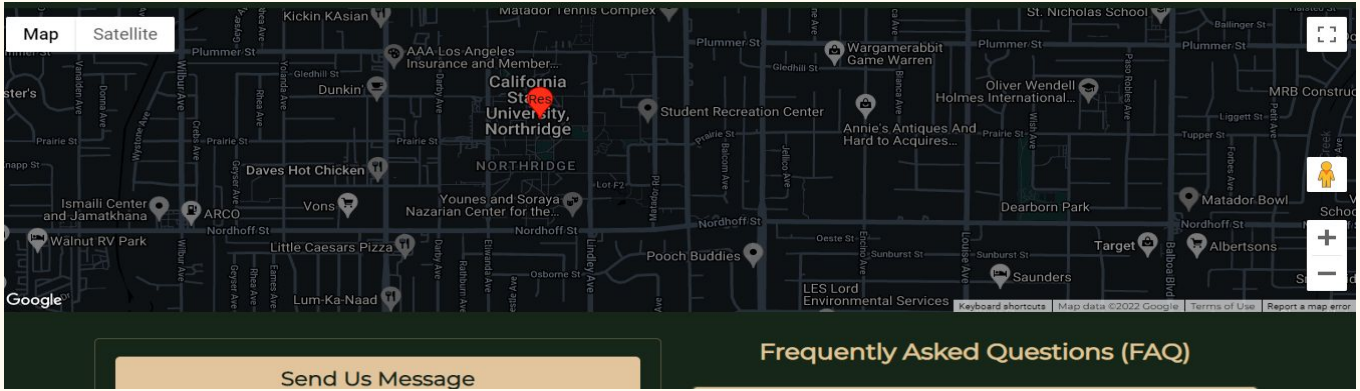
Documentation: https://developers.google.com/maps/documentation/javascript/overview?hl=en_GB#maps_map_simple-javascript

Demonstration





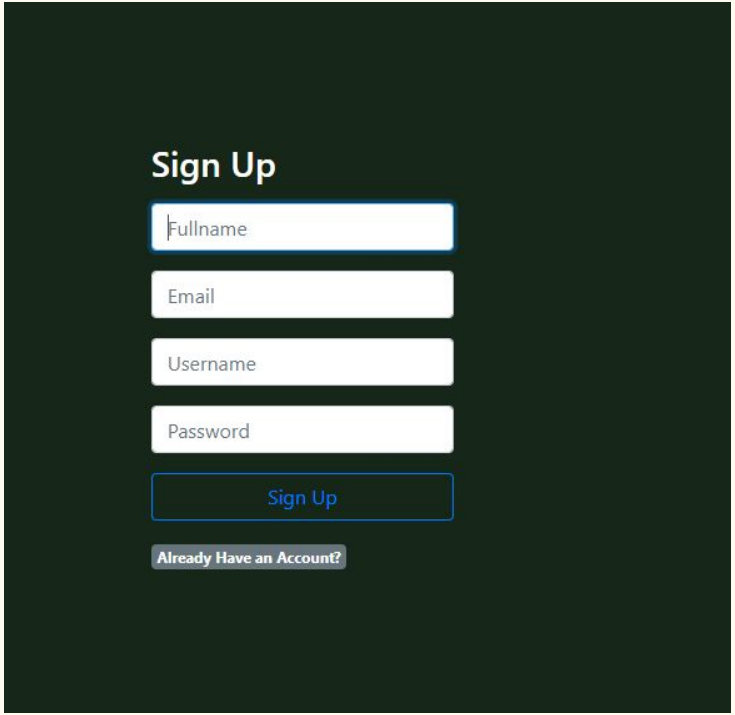
Google Map(Using API)




```
<!--Map Section-->
<section class="location" id="location">
</section>
    <div id="map"></div>
    <script async
    src="https://maps.googleapis.com/maps/api/js?key=AIzaSyB41DRUbKWJHPxaFjMAwdrzWzbVKartNGg&callback=initMap">
</script>



<!--Map Section Ends-->
```

```
//Google map
let map;
const csun = { lat: 34.23833238 , lng: -118.523664572 };
const csunlib = { lat: 34.240029 , lng: -118.529499 }
function initMap() {
  map = new google.maps.Map(document.getElementById("map"), {
    center: csun,
    zoom: 15,
    mapId: "490687d5666a03fe",
  });

  const marker = new google.maps.Marker({
    position: csunlib,
    label: "Res",
    map: map,
  });
}
window.initMap = initMap;
```

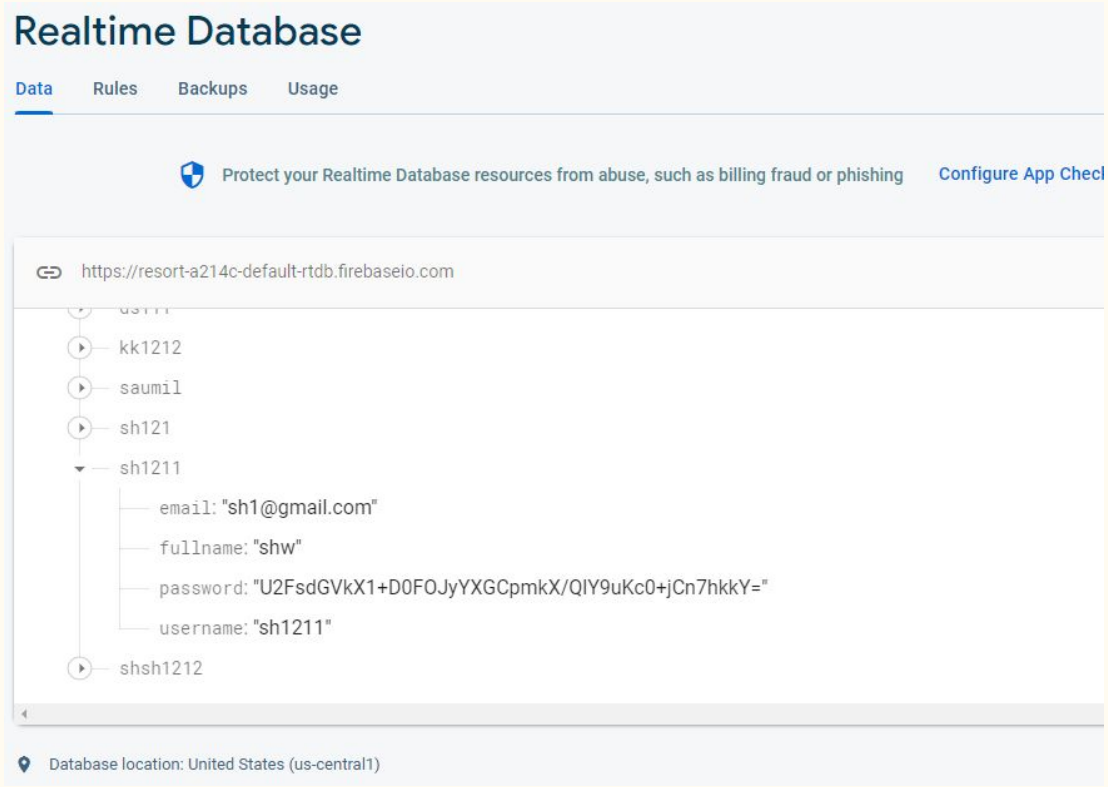
Sign-Up (Firebase and Authentication)



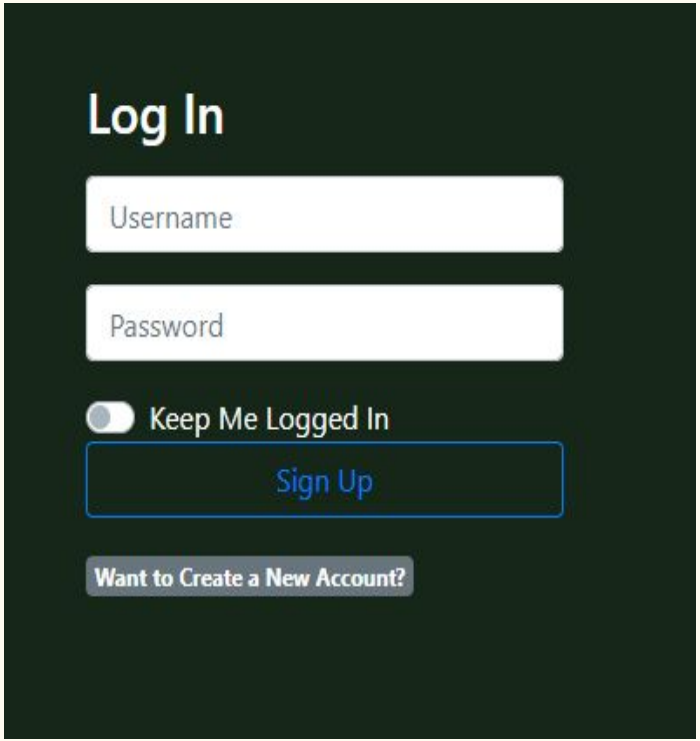

```
<!--Firebase Config-->
<script type="module">
    // Import the functions you need from the SDKs you need
    import { initializeApp } from "https://www.gstatic.com/firebasejs/9.15.0/firebase-app.js";
    import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.15.0/firebase-analytics.js";
    import { getDatabase, ref, set, child, get } from "https://www.gstatic.com/firebasejs/9.15.0/firebas

    const firebaseConfig = {
      apiKey: "AIzaSyCCyLpdOYBt6wOwmWbMPmVdpgnwEhsD7Gs",
      authDomain: "resort-a214c.firebaseapp.com",
      databaseURL: "https://resort-a214c-default-rtdb.firebaseio.com",
      projectId: "resort-a214c",
      storageBucket: "resort-a214c.appspot.com",
      messagingSenderId: "577361144057",
      appId: "1:577361144057:web:9daa1fe6b639a5e10f5a28",
      measurementId: "G-2RZ69QFRRW"
    };

    // Initialize Firebase
    const app = initializeApp(firebaseConfig);
    const analytics = getAnalytics(app);
    const db = getDatabase();
```

```
    //Validation(Trough reg exp.)

    //To check the empty fields
    function isEmptyOrSpaces(str){
        return str === null || str.match(/^ *$/) !== null;
    }

    function Validation(){
        let nameregex = /^[a-zA-Z]+$/; //any alphabets small or big any
        let emailregex = /^[a-zA-Z0-9]+@(gmail|outlook|yahoo|icloud)\.com$/;//any number or alphabets is
        let usernameregex = /^[a-zA-Z0-9]{5,}$/;//contains 5 char min


        if (isEmptyOrSpaces(name.value) || isEmptyOrSpaces(email.value) || isEmptyOrSpaces(username.valu
            alert("you cannot left anyfield empty!");
            return false;
        }

        if (!nameregex.test(name.value)){
            alert("The name should only contains alphabets!");
            return false;
        }

        if (!emailregex.test(email.value)) {
            alert("Enter Valid Email");
            return false;
        }
```

Firebase RealTime Database(With Password Encryption)



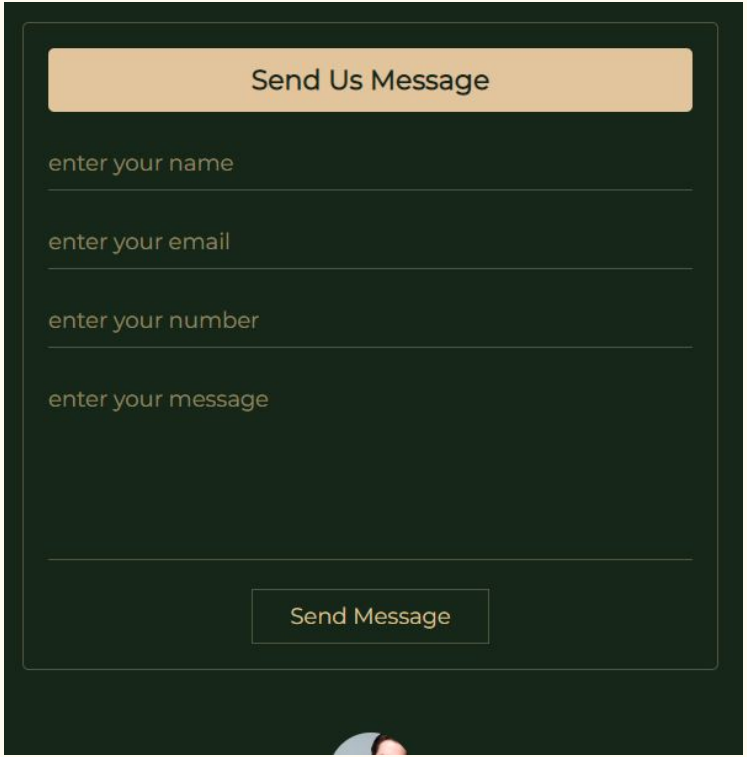
Log-In (Firebase and Authentication)




```
<!--Firebase Config-->
<script type="module">
    // Import the functions you need from the SDKs you need
    import { initializeApp } from "https://www.gstatic.com/firebasejs/9.15.0/firebase-app.js";
    import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.15.0/firebase-analytics.js";
    import { getDatabase, ref, set, child, get } from "https://www.gstatic.com/firebasejs/9.15.0/fireba
    // TODO: Add SDKs for Firebase products that you want to use
    // https://firebase.google.com/docs/web/setup#available-libraries

    // Your web app's Firebase configuration
    // For Firebase JS SDK v7.20.0 and later, measurementId is optional
    const firebaseConfig = {
      apiKey: "AIzaSyCCyLpdOYBt6wOwmWbMPmVdpgnwEhsD7Gs",
      authDomain: "resort-a214c.firebaseapp.com",
      databaseURL: "https://resort-a214c-default-rtdb.firebaseio.com",
      projectId: "resort-a214c",
      storageBucket: "resort-a214c.appspot.com",
      messagingSenderId: "577361144057",
      appId: "1:577361144057:web:9daa1fe6b639a5e10f5a28",
      measurementId: "G-2RZ69QFRRW"
    };

    // Initialize Firebase
    const app = initializeApp(firebaseConfig);
    const analytics = getAnalytics(app);
```

```
//Authentication Process

function AuthenticateUser(){
    const dbref = ref(db);
    get(child(dbref, "UsersList/"+ username.value)).then((snapshot)=>{
        if(snapshot.exists()){
            let dbpass = decPass(snapshot.val().password);
            if (dbpass == pass.value) {
                login(snapshot.val());
            }
            else{
                alert("Userdoes not exist!");
            }
        }
        else{
            alert("Invalid Username and Password!");
        }

    });
}


//Decrypt Process


function decPass(dbpass){
    var pass12 = CryptoJS.AES.decrypt(dbpass, pass.value);
    return pass12.toString(CryptoJS.enc.Utf8);
}


//Login
```

Contact (EmaiJs)




```
<!-- contact section starts  -->

<section class="contact" id="contact">

   <div class="row">

      <form action="" method="POST">
         <h3>send us message</h3>
         <input type="text" id="name" name="name" required maxlength="50" placeholder="enter your name" class="
         <input type="email" id="email" name="email" required maxlength="50" placeholder="enter your email" clas
         <input type="number" id="phone" name="number" required maxlength="10" min="0" max="9999999999" placeho
         <textarea name="msg" id="message" class="box" required maxlength="1000" placeholder="enter your message
         <button type="submit" class="btn" onclick="sendMail()">Send Message</button>
      </form>

      <div class="fag">
```

```
91
92    //Email Send From Contact Field
93    function sendMail(){
94       var params = {
95          name: document.getElementById("name").value,
96          email: document.getElementById("email").value,
97          message: document.getElementById("message").value,
98       };
99       const serviceID = "service_xgiyey8";
100      const templateID = "template_srzg2zq";
101
102      emailjs.send(serviceID, templateID, params).then(function (res){
103         alert("Email Sent!");
104      })
105   }
106
107
```

Conclusion

- Free hosting usually means that there is a limit on functionalities that can be added to a website.
- We learnt firebase database connectivity and usage of real time data.
- We learnt formatting and designing a website application.
- We learn API connectivity and using and implementing its functionalities.
- We tried and implemented automated email functionality for contact details.

Alternative libraries/ framework

SMTPJS - For Sending email

PHP - For backend

Radar - Map API

MySql - For Database

Pros and Cons of libraries used

Library	Pros	Cons
FireBase	Uses Realtime Database	Can’t Use for Admin Page
Google API	Edit Map in any Way	Generate New API key after some time
EmailJS	Send Automated Email	Vulnerable (Use Own Account)
Bootstrap	Fewer Cross browser bugs	JavaScript is tied to jQuery

References

[1] https://www.w3schools.com/html/

[2] https://firebase.google.com/docs/functions/get-started

[3] https://developers.google.com/maps/documentation/javascript/overview?hl=en_GB#maps_map_simple-javascript

[4] https://www.emailjs.com/docs/

[5] https://getbootstrap.com/docs/4.4/getting-started/introduction/

Links For Website and Github

Website - https://resorts2r.netlify.app/index.html

Github - https://github.com/shrey-patel14/resorts2r.git