<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Professor Availability</title>

<!-- Firebase & Razorpay SDK -->

<script src="https://www.gstatic.com/firebasejs/10.10.0/firebase-app.js"></script>

<script src="https://www.gstatic.com/firebasejs/10.10.0/firebase-auth.js"></script>

<script src="https://www.gstatic.com/firebasejs/10.10.0/firebase-firestore.js"></script>

<script src="https://checkout.razorpay.com/v1/checkout.js"></script>

<style>

body { font-family: Arial, sans-serif; text-align: center; padding: 20px; }

.container { max-width: 600px; margin: auto; }

input, button { margin: 10px; padding: 10px; }

.status { font-size: 18px; font-weight: bold; color: green; }

#availability, #adminPanel { display: none; margin-top: 20px; }

</style>

</head>

<body>

<div class="container">

<h1>Professor Availability Checker</h1>

<h2>User Login / Register</h2>

<input type="email" id="email" placeholder="Enter Email">

<input type="password" id="password" placeholder="Enter Password">

<button onclick="register()">Register</button>

<button onclick="login()">Login</button>

<button onclick="logout()">Logout</button>

<br><br>

<button onclick="makePayment()">Pay ₹10 to Access</button>

<p class="status" id="status"></p>

<div id="availability">

<h2>Check Professor Availability</h2>

<input type="text" id="professorName" placeholder="Enter Professor's Name">

<button onclick="checkAvailability()">Check</button>

<p class="status" id="availabilityStatus"></p>

</div>

<div id="adminPanel">

<h2>Admin Panel - Update Availability</h2>

<input type="text" id="adminProfessorName" placeholder="Professor Name">

<select id="adminStatus">

<option value="Available">Available</option>

<option value="Not Available">Not Available</option>

</select>

<button onclick="updateAvailability()">Update Status</button>

</div>

</div>

<script>

// Firebase Configuration (Replace with your details)

const firebaseConfig = {

apiKey: "YOUR\_API\_KEY",

authDomain: "YOUR\_PROJECT\_ID.firebaseapp.com",

projectId: "YOUR\_PROJECT\_ID",

storageBucket: "YOUR\_PROJECT\_ID.appspot.com",

messagingSenderId: "YOUR\_MESSAGING\_SENDER\_ID",

appId: "YOUR\_APP\_ID"

};

// Initialize Firebase

firebase.initializeApp(firebaseConfig);

const auth = firebase.auth();

const db = firebase.firestore();

// User Registration

function register() {

let email = document.getElementById("email").value;

let password = document.getElementById("password").value;

auth.createUserWithEmailAndPassword(email, password)

.then(() => alert("Registered Successfully!"))

.catch(error => alert("Error: " + error.message));

}

// User Login

function login() {

let email = document.getElementById("email").value;

let password = document.getElementById("password").value;

auth.signInWithEmailAndPassword(email, password)

.then(() => checkAccess())

.catch(error => alert("Error: " + error.message));

}

// Logout Function

function logout() {

auth.signOut().then(() => {

document.getElementById("availability").style.display = "none";

document.getElementById("adminPanel").style.display = "none";

document.getElementById("status").innerText = "Logged out";

}).catch(error => alert("Logout failed: " + error.message));

}

// Payment Function (Razorpay)

function makePayment() {

var options = {

"key": "YOUR\_RAZORPAY\_KEY",

"amount": 1000, // 10 INR in paise

"currency": "INR",

"name": "Professor Availability",

"description": "Access for 12 hours",

"handler": function (response) {

let user = auth.currentUser;

if (user) {

db.collection("users").doc(user.uid).set({

access: true,

expiry: Date.now() + (12 \* 60 \* 60 \* 1000) // 12 hours

}).then(() => {

alert("Payment Successful! Refreshing...");

location.reload();

}).catch(error => alert("Database error: " + error.message));

}

}

};

try {

var rzp = new Razorpay(options);

rzp.open();

} catch (error) {

alert("Payment failed: " + error.message);

}

}

// Check if user has access

function checkAccess() {

let user = auth.currentUser;

if (user) {

db.collection("users").doc(user.uid).get().then((doc) => {

if (doc.exists && doc.data().access && Date.now() < doc.data().expiry) {

document.getElementById("availability").style.display = "block";

document.getElementById("status").innerText = "Access granted";

} else {

document.getElementById("status").innerText = "Payment required to access";

}

}).catch(error => alert("Error checking access: " + error.message));

// Show Admin Panel if user is admin

if (user.email === "youradminemail@example.com") {

document.getElementById("adminPanel").style.display = "block";

}

}

}

// Check Professor Availability

function checkAvailability() {

let professor = document.getElementById("professorName").value;

let status = document.getElementById("availabilityStatus");

db.collection("professors").doc(professor).get().then((doc) => {

if (doc.exists) {

status.innerText = professor + " is " + doc.data().status;

} else {

status.innerText = "No data available";

}

}).catch(error => alert("Error fetching data: " + error.message));

}

// Admin Updates Professor Availability

function updateAvailability() {

let professor = document.getElementById("adminProfessorName").value;

let status = document.getElementById("adminStatus").value;

db.collection("professors").doc(professor).set({

status: status

}).then(() => {

alert(professor + " status updated to " + status);

}).catch(error => alert("Update failed: " + error.message));

}

</script>

</body>

</html>