// Firebase Configuration

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"

};

const app = firebase.initializeApp(firebaseConfig);

const auth = firebase.auth();

const firestore = firebase.firestore();

const storage = firebase.storage();

// Stripe Configuration

const stripe = Stripe('YOUR\_PUBLIC\_KEY');

const payButton = document.getElementById('payButton');

// Handle user login/signup

auth.onAuthStateChanged(function(user) {

if (user) {

console.log('User logged in:', user.email);

} else {

console.log('No user logged in');

}

});

// User Profile Update

const profileForm = document.getElementById('update-profile-form');

profileForm.addEventListener('submit', function(e) {

e.preventDefault();

const newEmail = document.getElementById('email').value;

const newPassword = document.getElementById('new-password').value;

const user = auth.currentUser;

if (newEmail !== user.email) {

user.updateEmail(newEmail).then(() => {

alert('Email updated successfully!');

}).catch((error) => {

console.error('Error updating email:', error);

});

}

if (newPassword) {

user.updatePassword(newPassword).then(() => {

alert('Password updated successfully!');

}).catch((error) => {

console.error('Error updating password:', error);

});

}

});

// Support Form

const supportForm = document.getElementById('support-form');

supportForm.addEventListener('submit', function(e) {

e.preventDefault();

const query = document.getElementById('support-query').value;

// Save support query to Firestore

firestore.collection('support').add({

query: query,

user: auth.currentUser.email,

timestamp: firebase.firestore.FieldValue.serverTimestamp()

}).then(() => {

alert('Your support request has been submitted!');

}).catch((error) => {

console.error('Error submitting support:', error);

});

});

// Content Upload Functionality

const uploadForm = document.getElementById('upload-form');

uploadForm.addEventListener('submit', function(e) {

e.preventDefault();

const file = document.getElementById('content').files[0];

const name = document.getElementById('name').value;

const description = document.getElementById('description').value;

const price = document.getElementById('price').value;

const storageRef = storage.ref('uploads/' + file.name);

const uploadTask = storageRef.put(file);

uploadTask.on('state\_changed', function(snapshot) {

// Progress tracking (optional)

}, function(error) {

console.error('Upload failed:', error);

}, function() {

console.log('File uploaded successfully');

uploadTask.snapshot.ref.getDownloadURL().then(function(downloadURL) {

firestore.collection('content').add({

name: name,

description: description,

url: downloadURL,

price: price,

likes: 0,

followers: [],

comments: [],

created\_at: firebase.firestore.FieldValue.serverTimestamp()

}).then(() => {

alert('Content uploaded successfully!');

loadContent();

});

});

});

});

// Load Content (same as before)

function loadContent() {

firestore.collection('content').orderBy('created\_at', 'desc').get().then(snapshot => {

const contentList = document.getElementById('content-list');

contentList.innerHTML = '';

snapshot.forEach(doc => {

const content = doc.data();

const contentDiv = document.createElement('div');

contentDiv.innerHTML = `

<h3>${content.name}</h3>

<img src="${content.url}" alt="Content" />

<p>${content.description}</p>

<button class="like-button" data-id="${doc.id}">Like (${content.likes})</button>

<button class="follow-button" data-id="${doc.id}">Follow</button>

<button class="share-button" data-url="${content.url}">Share</button>

`;

contentList.appendChild(contentDiv);

});

// Add event listeners for like, follow, and share

document.querySelectorAll('.like-button').forEach(button => {

button.addEventListener('click', function() {

const id = button.getAttribute('data-id');

likeContent(id);

});

});

document.querySelectorAll('.follow-button').forEach(button => {

button.addEventListener('click', function() {

const id = button.getAttribute('data-id');

followUploader(id);

});

});

document.querySelectorAll('.share-button').forEach(button => {

button.addEventListener('click', function() {

const url = button.getAttribute('data-url');

window.open(`https://twitter.com/share?url=${url}`, '\_blank');

});

});

});

}

// Like and Follow functionality (same as before)

function likeContent(contentId) {

const contentRef = firestore.collection('content').doc(contentId);

contentRef.update({

likes: firebase.firestore.FieldValue.increment(1)

}).then(() => {

loadContent();

});

}

function followUploader(contentId) {

const contentRef = firestore.collection('content').doc(contentId);

contentRef.update({

followers: firebase.firestore.FieldValue.arrayUnion(auth.currentUser.uid)

}).then(() => {

alert('You followed the uploader!');

});

}

// Call to load content initially

loadContent();