PROGRAM CODE

# server.js:

const express = require('express'); const mongoose = require('mongoose'); const bcrypt = require('bcryptjs');

const bodyParser = require('body-parser'); const cookieParser = require('cookie-parser'); const multer = require('multer');

const path = require('path'); const fs = require('fs');

const QRCode = require('qrcode'); const app = express();

const PORT = 3000;

// Middleware app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true })); app.use(cookieParser());

app.use(express.static(path.join( dirname, 'public'))); // Serve static files from public directory

app.use('/uploads', express.static(path.join( dirname, 'uploads'))); // Serve static files from uploads directory

// Ensure the uploads directory exists

const uploadDir = path.join( dirname, 'uploads');

if (!fs.existsSync(uploadDir)) fs.mkdirSync(uploadDir);

// Configure multer for file uploads const storage = multer.diskStorage({

destination: (req, file, cb) => { cb(null, uploadDir);

},

filename: (req, file, cb) => {

cb(null, `${Date.now()}-${file.originalname}`);

},

});

const upload = multer({ storage });

// Connect to MongoDB mongoose.connect('mongodb://127.0.0.1:27017/loginSystem', {

useNewUrlParser: true, useUnifiedTopology: true,

})

.then(() => console.log('MongoDB Connected'))

.catch(err => {

console.error('MongoDB connection error:', err); process.exit(1); // Exit process on failure

});

// User Schema

const userSchema = new mongoose.Schema({

email: { type: String, required: true, unique: true }, password: { type: String, required: true },

});

const User = mongoose.model('User', userSchema);

// File Schema

const fileSchema = new mongoose.Schema({

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

filename: String, filepath: String, qrCodePath: String,

uploadDate: { type: Date, default: Date.now },

});

const File = mongoose.model('File', fileSchema);

// Routes

// Upload Route

app.post('/upload', upload.single('file'), async (req, res) => { const { userId } = req.body;

if (!req.file) return res.status(400).send('No file uploaded.');

try {

const filePath = `/uploads/${req.file.filename}`; const qrCodePath = path.join(uploadDir,

`${req.file.filename}.png`); // Path for QR code image

// Generate QR code for the uploaded file URL await QRCode.toFile(qrCodePath,

`http://localhost:${PORT}${filePath}`); // Adjust for production URL

const newFile = new File({ userId,

filename: req.file.originalname, filepath: filePath,

qrCodePath: `/uploads/${req.file.filename}.png`, // Store relative path of QR code

});

await newFile.save();

res.send(`File uploaded successfully! <br> <img src="${newFile.qrCodePath}" alt="QR Code" />`); // Send back QR code image

} catch (error) {

console.error('Error uploading file:', error); res.status(500).send('Error uploading file');

}

});

// Signup Route

app.post('/signup', async (req, res) => { const { email, password } = req.body;

try {

const existingUser = await User.findOne({ email }); if (existingUser) {

return res.status(400).send('Email already in use.');

}

const hashedPassword = await bcrypt.hash(password, 10);

const newUser = new User({ email, password: hashedPassword }); await newUser.save();

res.redirect('/'); // Redirect to login after successful signup

} catch (error) {

console.error('Error creating user:', error); res.status(500).send('Error creating user');

}

});

// Login Route

app.post('/login', async (req, res) => { const { email, password } = req.body; try {

const user = await User.findOne({ email });

if (!user) return res.status(400).send('Invalid email or password');

const isMatch = await bcrypt.compare(password, user.password); if (!isMatch) return res.status(400).send('Invalid email or

password');

res.redirect(`/upload.html?userId=${user.\_id}`); // Redirect to upload page with user ID

} catch (error) {

console.error('Error during login:', error); res.status(500).send('Error during login');

}

});

// File Upload Route

app.post('/upload', upload.single('file'), async (req, res) => { const { userId } = req.body;

if (!req.file) return res.status(400).send('No file uploaded.');

try {

const newFile = new File({ userId,

filename: req.file.originalname, filepath: `/uploads/${req.file.filename}`,

});

await newFile.save();

res.send('File uploaded successfully!');

} catch (error) {

console.error('Error uploading file:', error); res.status(500).send('Error uploading file');

}

});

// Function to delete a file function deleteFile(fileId) {

fetch(`/files/${fileId}`, { method: 'DELETE',

})

.then(response => {

if (!response.ok) throw new Error('Network response was not ok'); return response.text();

})

.then(message => {

alert(message); // Show success message location.reload(); // Reload the page to refresh the file list

})

.catch(error => console.error('Error deleting file:', error));

}

// Delete File Route

app.delete('/files/:id', async (req, res) => { const { id } = req.params;

try {

// Find the file in the database

const file = await File.findById(id);

if (!file) return res.status(404).send('File not found');

// Delete the file from the filesystem

const filePath = path.join( dirname, 'uploads', path.basename(file.filepath));

fs.unlink(filePath, async (err) => { if (err) {

console.error('Error deleting file:', err);

return res.status(500).send('Error deleting file');

}

// Remove the file record from the database await File.deleteOne({ \_id: id }); res.send('File deleted successfully');

});

} catch (error) {

console.error('Error deleting file:', error); res.status(500).send('Error deleting file');

}

});

// Retrieve User Files app.get('/files', async (req, res) => {

const { userId } = req.query; try {

const files = await File.find({ userId }); res.json(files);

} catch (error) {

console.error('Error fetching files:', error); res.status(500).send('Error fetching files');

}

});

// Serve HTML Files app.get('/', (req, res) => {

res.sendFile(path.join( dirname, 'public', 'index.html'));

});

// Serve upload.html directly when requested app.get('/upload.html', (req, res) => {

res.sendFile(path.join( dirname, 'public', 'upload.html'));

});

// Start Server

app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));

# first.html

<!DOCTYPE html>

<html>

<head>

<!-- Basic -->

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<!-- Mobile Metas -->

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

<!-- Site Metas -->

<meta name="keywords" content="" />

<meta name="description" content="" />

<meta name="author" content="" />

<link rel="shortcut icon" href="s1.png" type="">

<title>CloudBin.</title>

<!-- bootstrap core css -->

<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />

<!-- fonts style -->

<link href=["https://fonts.googleapis.com/](https://fonts.googleapis.com/css2?family=Roboto%3Awght%40400%3B500%3B)c[ss2?family=Roboto:wght@400;500;](https://fonts.googleapis.com/css2?family=Roboto%3Awght%40400%3B500%3B) 700;900&display=swap" rel="stylesheet">

<!--owl slider stylesheet -->

<link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/o wl.carousel.min.css" />

<!-- font awesome style -->

<link href="css/font-awesome.min.css" rel="stylesheet" />

<!-- Custom styles for this template -->

<link href="css/style.css" rel="stylesheet" />

<!-- responsive style -->

<link href="css/responsive.css" rel="stylesheet" />

</head>

<body>

<div class="hero\_area">

<div class="hero\_bg\_box">

<div class="bg\_img\_box">

<img src="images/hero-bg.png" alt="">

</div>

</div>

<!-- header section strats -->

<header class="header\_section">

<div class="container-fluid">

<nav class="navbar navbar-expand-lg custom\_nav-container ">

<a class="navbar-brand" href="#">

<span> CloudBin.

</span>

</a>

<button class="navbar-toggler" type="button" data- toggle="collapse" data-target="#navbarSupportedContent" aria- controls="navbarSupportedContent" aria-expanded="false" aria- label="Toggle navigation">

<span class=""> </span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav ">

<li class="nav-item active">

<a class="nav-link" href="index.html">Home <span class="sr- only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="index.html"> <i class="fa fa-user" aria-hidden="true"></i> Login</a>

</li>

</ul>

</div>

</nav>

</div>

</header>

<!-- end header section -->

<!-- slider section -->

<section class="slider\_section ">

<div id="customCarousel1" class="carousel slide" data- ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">

<div class="container ">

<div class="row">

<div class="col-md-6 ">

<div class="detail-box">

<h1>

CLoudBin

</h1>

<p>

CloudBin is "made in India" storage platform, that allows users to upload, manage,

and share their files effortlessly while benefiting from low- latency access. With features like unique QR code sharing.

</p>

<div class="btn-box">

</div>

</div>

</div>

<div class="col-md-6">

<div class="img-box">

<img src="slider-img.png" alt="">

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- end slider section -->

</div>

<!-- service section -->

<section class="service\_section layout\_padding">

<div class="service\_container">

<div class="container ">

<div class="heading\_container heading\_center">

<h2>

Our <span>Services</span>

</h2>

</div>

<div class="row">

<div class="col-md-4 ">

<div class="box ">

<div class="img-box">

<img src="qr.png" alt="">

</div>

<div class="detail-box">

<h5>

QR Code Sharing

</h5>

<p>

Generate unique QR codes for each uploaded file, allowing you to share access quickly and efficiently with anyone, anywhere.

Recipients can scan the code to download or view the file

instantly.

</p>

</div>

</div>

</div>

<div class="col-md-4 ">

<div class="box ">

<div class="img-box">

<img src="images/s2.png" alt="">

</div>

<div class="detail-box">

<h5>

Security Features

</h5>

<p>

Enjoy peace of mind with our advanced security measures that protect your files during upload and while stored in the cloud.

We prioritize your privacy and data integrity.

</p>

</div>

</div>

</div>

<div class="col-md-4 ">

<div class="box ">

<div class="img-box">

<img src="chatbot.png" alt="">

</div>

<div class="detail-box">

<h5>

User Support

</h5>

<p>

Our dedicated support team is available to assist you with any questions or issues you may encounter while using CloudBin.

We are committed to providing excellent customer service.

</p>

</div>

</div>

</div>

</div>

<div class="btn-box">

</div>

</div>

</div>

</section>

<!-- end service section -->

<!-- about section -->

<section class="about\_section layout\_padding">

<div class="container ">

<div class="heading\_container heading\_center">

<h2>

About <span>Us</span>

</h2>

<h3>

We Are CloudBin.

</h3>

<p>

CloudBin is an innovative platform designed to simplify file management in the digital age. Our mission is to empower users by providing a seamless experience for uploading, sharing and managing their files online. With CloudBin, you can easily upload various types of files, from documents to images and share them effortlessly through unique QR codes. This not only enhances accessibility but also ensures that your files are securely stored in the cloud.

</p>

<p>

At CloudBin, we prioritize user control and privacy. Users have the flexibility to delete their files whenever they choose, ensuring that they maintain full ownership and management of their data. Our user- friendly interface and robust security measures make CloudBin the ideal solution for individuals and businesses alike who seek a reliable file- sharing platform.

</p>

</div>

</section>

<!-- end about section -->

<!-- team section -->

<section class="team\_section layout\_padding">

<div class="container-fluid">

<div class="heading\_container heading\_center">

<h2 class="">

Our <span> Team</span>

</h2>

</div>

<div class="team\_container">

<div class="row">

<div class="col-lg-3 col-sm-6">

<div class="box ">

<div class="img-box">

<img src="team-1.png" class="img1" alt="">

</div>

<div class="detail-box">

<h5>

Giridhar

</h5>

<p>

Project Head / Fullstack Developer

</p>

</div>

<div class="social\_box">

<a href="https://[www.linkedin.com/in/giridhar-uppili-](http://www.linkedin.com/in/giridhar-uppili-) 966358192/">

<i class="fa fa-linkedin" aria-hidden="true"></i>

</a>

<a href="https://[www.instagram.com/\_giridharphotography/"](http://www.instagram.com/_giridharphotography/)>

<i class="fa fa-instagram" aria-hidden="true"></i>

</a>

</div>

</div>

</div>

<div class="col-lg-3 col-sm-6">

<div class="box ">

<div class="img-box">

<img src="team-2.jpg" class="img1" alt="">

</div>

<div class="detail-box">

<h5>

Ezhil Adhithya

</h5>

<p>

Backend engineer

</p>

</div>

<div class="social\_box">

<a href="https://[www.linkedin.com/in/ezhiladhithya/"](http://www.linkedin.com/in/ezhiladhithya/)>

<i class="fa fa-linkedin" aria-hidden="true"></i>

</a>

<a href="https://[www.instagram.com/\_hakkanoodles\_/"](http://www.instagram.com/_hakkanoodles_/)>

<i class="fa fa-instagram" aria-hidden="true"></i>

</a>

</div>

</div>

</div>

<div class="col-lg-3 col-sm-6">

<div class="box ">

<div class="img-box">

<img src="team-3.jpg" class="img1" alt="">

</div>

<div class="detail-box">

<h5>

Harish

</h5>

<p>

Database Administrator

</p>

</div>

<div class="social\_box">

<a href="https://[www.linkedin.com/in/harish-t-7b4428316/"](http://www.linkedin.com/in/harish-t-7b4428316/)>

<i class="fa fa-linkedin" aria-hidden="true"></i>

</a>

<a href="https://[www.instagram.com/harixh.](http://www.instagram.com/harixh) .2510/">

<i class="fa fa-instagram" aria-hidden="true"></i>

</a>

</div>

</div>

</div>

<div class="col-lg-3 col-sm-6">

<div class="box ">

<div class="img-box">

<img src="team-4.jpg" class="img1" alt="">

</div>

<div class="detail-box">

<h5>

Aravind

</h5>

<p>

Frontend Developer

</p>

</div>

<div class="social\_box">

<a href="https://[www.linkedin.com/in/aravind-s-2659102a2/"](http://www.linkedin.com/in/aravind-s-2659102a2/)>

<i class="fa fa-linkedin" aria-hidden="true"></i>

</a>

<a href="https://[www.instagram.com/aravinds106">](http://www.instagram.com/aravinds106)

<i class="fa fa-instagram" aria-hidden="true"></i>

</a>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- end team section -->

<!-- footer section -->

<section class="footer\_section">

<div class="container">

<p>

&copy; <span id="displayYear"></span> All Rights Reserved By Cloudbin Pvt Ltd.

</p>

</div>

</section>

<!-- footer section -->

</body>

</html>

# index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<link rel="stylesheet" href="styles.css" />

<title>Cloud Bin</title>

</head>

<body>

<div class="loading-screen">

<div class="loader"></div>

<div class="loading-text">CloudBin.</div>

</div>

<div class="wrapper">

<div class="title-text">

<div class="title login">Login Form</div>

<div class="title signup">Signup Form</div>

</div>

<div class="form-container">

<div class="slide-controls">

<input type="radio" name="slide" id="login" checked />

<input type="radio" name="slide" id="signup" />

<label for="login" class="slide login">Login</label>

<label for="signup" class="slide signup">Signup</label>

<div class="slider-tab"></div>

</div>

<div class="form-inner">

<!-- Login Form -->

<form action="/login" method="POST" class="login">

<div class="field">

<input type="text" name="email" placeholder="Email Address" required />

</div>

<div class="field">

<input type="password" name="password" placeholder="Password" required />

</div>

<div class="pass-link"><a href="#">Forgot password?</a></div>

<div class="field btn">

<div class="btn-layer"></div>

<input type="submit" value="Login" />

</div>

<div class="signup-link">Not a member? <a href="">Signup now</a></div>

</form>

<!-- Signup Form -->

<form action="/signup" method="POST" class="signup">

<div class="field">

<input type="text" name="email" placeholder="Email Address" required />

</div>

<div class="field">

<input type="password" name="password" placeholder="Password" required />

</div>

<div class="field">

<input type="password"

name="confirmPassword"

placeholder="Confirm password" required

/>

</div>

<div class="field btn">

<div class="btn-layer"></div>

<input type="submit" value="Signup" />

</div>

</form>

</div>

</div>

</div>

<script src="script.js"></script>

<script>

// Hide loading screen after 3 seconds setTimeout(() => {

document.querySelector('.loading-screen').style.display = 'none';

}, 1000); // Adjust time as needed

</script>

</body>

</html>

# style.css

{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

html,body{ display: grid; height: 100%;

width: 100%;

place-items: center;

background: -webkit-linear-gradient(left, #003366,#004080,#0059b3

, #0073e6);

}

::selection{ background: #1a75ff; color: #fff;

}

.loading-screen { position: fixed; top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.8); /\* Semi-transparent background

\*/

display: flex;

flex-direction: column; justify-content: center; align-items: center;

z-index: 999; /\* Ensure it covers everything \*/

}

.loading-text {

font-size: 48px; /\* Adjust size as needed \*/ font-weight: bold;

color: #fff; /\* Text color \*/

margin-top: 20px; /\* Space between loader and text \*/

}

/\* Loader styles \*/

.loader {

border: 16px solid #f3f3f3; /\* Light grey \*/ border-top: 16px solid #003366; /\* Blue \*/

border-radius: 50%;

width: 120px; /\* Size of the loader \*/ height: 120px; /\* Size of the loader \*/

animation: spin 2s linear infinite; /\* Animation for spinning effect \*/

}

/\* Keyframes for spin animation \*/ @keyframes spin {

0% { transform: rotate(0deg); } 100% { transform: rotate(360deg); }

}

.wrapper{ overflow: hidden;

max-width: 390px; background: #fff; padding: 30px; border-radius: 15px;

box-shadow: 0px 15px 20px rgba(0,0,0,0.1);

}

.wrapper .title-text{ display: flex; width: 200%;

}

.wrapper .title{ width: 50%; font-size: 35px;

font-weight: 600; text-align: center;

transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);

}

.wrapper .slide-controls{ position: relative; display: flex;

height: 50px; width: 100%; overflow: hidden;

margin: 30px 0 10px 0;

justify-content: space-between; border: 1px solid lightgrey; border-radius: 15px;

}

.slide-controls .slide{ height: 100%;

width: 100%; color: #fff;

font-size: 18px;

font-weight: 500; text-align: center; line-height: 48px; cursor: pointer;

z-index: 1;

transition: all 0.6s ease;

}

.slide-controls label.signup{ color: #000;

}

.slide-controls .slider-tab{ position: absolute; height: 100%;

width: 50%;

left: 0;

z-index: 0;

border-radius: 15px;

background: -webkit-linear-gradient(left,#003366,#004080,#0059b3

, #0073e6);

transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);

}

input[type="radio"]{ display: none;

}

#signup:checked ~ .slider-tab{ left: 50%;

}

#signup:checked ~ label.signup{ color: #fff;

cursor: default; user-select: none;

}

#signup:checked ~ label.login{ color: #000;

}

#login:checked ~ label.signup{ color: #000;

}

#login:checked ~ label.login{ cursor: default;

user-select: none;

}

.wrapper .form-container{ width: 100%;

overflow: hidden;

}

.form-container .form-inner{ display: flex;

width: 200%;

}

.form-container .form-inner form{ width: 50%;

transition: all 0.6s cubic-bezier(0.68,-0.55,0.265,1.55);

}

.form-inner form .field{ height: 50px;

width: 100%; margin-top: 20px;

}

.form-inner form .field input{ height: 100%;

width: 100%; outline: none; padding-left: 15px; border-radius: 15px;

border: 1px solid lightgrey; border-bottom-width: 2px; font-size: 17px;

transition: all 0.3s ease;

}

.form-inner form .field input:focus{ border-color: #1a75ff;

/\* box-shadow: inset 0 0 3px #fb6aae; \*/

}

.form-inner form .field input::placeholder{ color: #999;

transition: all 0.3s ease;

}

form .field input:focus::placeholder{ color: #1a75ff;

}

.form-inner form .pass-link{ margin-top: 5px;

}

.form-inner form .signup-link{ text-align: center;

margin-top: 30px;

}

.form-inner form .pass-link a,

.form-inner form .signup-link a{ color: #1a75ff;

text-decoration: none;

}

.form-inner form .pass-link a:hover,

.form-inner form .signup-link a:hover{ text-decoration: underline;

}

form .btn{ height: 50px; width: 100%;

border-radius: 15px; position: relative; overflow: hidden;

}

form .btn .btn-layer{ height: 100%;

width: 300%; position: absolute; left: -100%;

background: -webkit-linear-gradient(right,#003366,#004080,#0059b3

, #0073e6);

border-radius: 15px; transition: all 0.4s ease;;

}

form .btn:hover .btn-layer{

left: 0;

}

form .btn input[type="submit"]{ height: 100%;

width: 100%;

z-index: 1; position: relative; background: none; border: none; color: #fff; padding-left: 0;

border-radius: 15px; font-size: 20px;

font-weight: 500; cursor: pointer;

}

# upload.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Cloud Bin</title>

<link rel="stylesheet" href="styless.css" />

</head>

<body>

<div class="container">

<header>

<h1>CloudBin.</h1>

<button id="logout-button">Logout</button>

</header>

<form id="upload-form" action="/upload" method="POST" enctype="multipart/form-data">

<h2>Upload File</h2>

<input type="file" name="file" required />

<input type="hidden" name="userId" id="userId" />

<button type="submit">Upload</button>

</form>

<div id="file-list">

<h3>Your Files</h3>

<div id="file-display"></div>

</div>

</div>

<!-- Chatbot Widget -->

<div id="chatbot">

<div class="chat-header">Chat with us!</div>

<div class="chat-body">

<div id="chat-messages"></div>

<input type="text" id="chat-input" placeholder="Type your message..." />

</div>

</div>

<script>

// Logout functionality

document.getElementById('logout-button').addEventListener('click', () => {

window.location.href = 'first.html';

});

// Extract userId from the query string const userId = new

URLSearchParams(window.location.search).get('userId'); document.getElementById('userId').value = userId;

// Fetch and display files for the logged-in user fetch(`/files?userId=${userId}`)

.then(response => {

if (!response.ok) throw new Error('Failed to fetch files'); return response.json();

})

.then(files => {

const fileDisplay = document.getElementById('file-display'); fileDisplay.innerHTML = '';

if (files.length === 0) {

fileDisplay.innerHTML = '<p>No files uploaded yet.</p>';

} else {

files.forEach(file => {

const card = document.createElement('div'); card.className = 'file-card';

const link = document.createElement('a'); link.href = file.filepath;

link.textContent = file.filename; link.target = '\_blank';

const qrButton = document.createElement('button');

qrButton.textContent = 'QR';

qrButton.onclick = () => window.open(file.qrCodePath, '\_blank'); // Open QR code image

const deleteButton = document.createElement('button'); deleteButton.textContent = 'Delete'; deleteButton.onclick = () => deleteFile(file.\_id);

card.appendChild(link); card.appendChild(qrButton); // Add QR button card.appendChild(deleteButton); fileDisplay.appendChild(card);

});

}

})

.catch(error => {

alert('Error fetching files. Please try again later.'); console.error('Error fetching files:', error);

});

// Function to delete a file function deleteFile(fileId) {

fetch(`/files/${fileId}`, {

method: 'DELETE',

})

.then(response => {

if (!response.ok) throw new Error('Network response was not

ok');

return response.text();

})

.then(message => { alert(message); location.reload();

})

.catch(error => console.error('Error deleting file:', error));

}

// Chatbot functionality let faqData = [];

// Fetch FAQ data from JSON file fetch('faq.json')

.then(response => response.json())

.then(data => { faqData = data; })

.catch(error => console.error('Error loading FAQ data:', error));

const chatMessages = document.getElementById('chat-messages'); const chatInput = document.getElementById('chat-input');

chatInput.addEventListener('keypress', function(event) { if (event.key === 'Enter') {

const userMessage = chatInput.value.trim(); if (userMessage) {

addMessage('You: ' + userMessage); chatInput.value = '';

// Find matching FAQ answer by keyword

const botResponse = getBotResponse(userMessage); addMessage('Bot: ' + botResponse);

}

}

});

function getBotResponse(userMessage) {

const lowerCaseMessage = userMessage.toLowerCase();

// Check for keywords in each FAQ entry for (const entry of faqData) {

if (entry.keywords.some(keyword => lowerCaseMessage.includes(keyword))) {

return entry.answer;

}

}

return "I'm sorry, I don't understand that. Try asking something

else!";

}

function addMessage(message) {

const messageDiv = document.createElement('div'); messageDiv.textContent = message; chatMessages.appendChild(messageDiv);

chatMessages.scrollTop = chatMessages.scrollHeight; // Auto- scroll to latest message

}

</script>

<!-- Add styles for chatbot -->

<style>

/\* Your existing styles for chatbot here \*/ #chatbot {

position: fixed; bottom: 20px; right: 20px; width: 300px;

border: 1px solid #ccc; border-radius: 8px; background-color: #fff;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1); display: flex;

flex-direction: column;

}

.chat-header {

background-color: #1a73e8; color: white;

padding: 10px; text-align: center;

border-top-left-radius: 8px; border-top-right-radius: 8px;

}

.chat-body { padding: 10px;

flex-grow: 1; /\* Allow body to grow \*/ overflow-y: auto; /\* Enable scrolling \*/

}

#chat-messages {

max-height: 200px; /\* Limit height \*/

}

#chat-input { border: none; padding: 10px;

width: calc(100% - 20px); /\* Full width minus padding \*/

}

</style>

</body>

</html>

# styless.css

/\* General styles \*/ html, body {

display: grid; height: 100%;

width: 100%;

place-items: center;

background: -webkit-linear-gradient(left, #003366, #004080, #0059b3, #0073e6);

}

/\* Container styles \*/

.container {

max-width: 900px; padding: 20px;

background: #ffffff; /\* White background for container \*/ border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

}

/\* Header styles \*/ header {

display: flex;

justify-content: space-between; align-items: center;

}

#logo {

max-width: 200px; /\* Adjust width as needed \*/ height: auto; /\* Maintain aspect ratio \*/

margin-bottom: 20px; /\* Space below logo \*/

}

/\* Heading styles \*/ h1, h2, h3 {

color: #202124; /\* Darker text for better readability \*/

}

/\* Button styles \*/ button {

background-color: #1a73e8; /\* Google Drive blue \*/ color: #fff;

border: none; border-radius: 4px;

padding: 10px 15px; cursor: pointer;

font-size: 14px;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #155ab6; /\* Darker blue on hover \*/

}

/\* Form styles \*/ form {

margin-top: 20px;

}

/\* File list styles \*/ #file-list {

margin-top: 20px;

}

#file-display { display: flex;

flex-wrap: wrap; /\* Allow cards to wrap \*/

}

.file-card {

background-color: #f9f9f9; /\* Light card background \*/ border-radius: 4px;

padding: 15px; margin: 10px;

box-shadow: 0 1px 3px rgba(0, 0, 0, 0.2);

width: calc(33% - 20px); /\* Three cards per row with spacing \*/

}

.file-card a {

display: block; /\* Makes the link fill the card area \*/ color: #1a73e8; /\* Link color similar to Google Drive \*/

}

.file-card button {

margin-top: 10px; /\* Space between link and button \*/

}

/\* Additional styles for QR button \*/

.file-card button:nth-child(2) { /\* Targeting the QR button specifically \*/ margin-left: 10px; /\* Space between filename and QR button \*/

}

# script.js

const loginText = document.querySelector(".title-text .login"); const loginForm = document.querySelector("form.login"); const loginBtn = document.querySelector("label.login"); const signupBtn = document.querySelector("label.signup");

const signupLink = document.querySelector("form .signup-link a"); signupBtn.onclick = (()=>{

loginForm.style.marginLeft = "-50%";

loginText.style.marginLeft = "-50%";

});

loginBtn.onclick = (()=>{ loginForm.style.marginLeft = "0%";

loginText.style.marginLeft = "0%";

});

signupLink.onclick = (()=>{ signupBtn.click();

return false;

});