!DOCTYPE html>

<html lang="en">

<head><link rel="stylesheet" href="project.css">

<meta charset="UTF-8">

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

<title>Resume Builder</title>

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

</head>

<body><script src="project.ts"></script>

<div class="container">

<h1>Resume Builder</h1>

<form id="resume-form">

<label for="name">Name:</label>

<input type="text" id="name" required>

<

<label for="email">Email:</label>

<input type="email" id="email" required>

<label for="experience">Experience:</label>

<textarea id="experience" required></textarea>

<button type="submit">Generate Resume</button>

</form>

<div id="resume-output"></div>

</div>

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

</body>

</html>

const express = require('express');

const mysql = require('mysql2');

const cors = require('cors');

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

const app = express();

const PORT = 52969; // Changed port number to 3000

const SECRET\_KEY = 'your\_secret\_key';

app.use(cors());

app.use(express.json());

// MySQL Database Connection

const db = mysql.createConnection({

host: 'localhost',

user: 'root',

password: 'password',

database: 'resume\_db',

port: 3300

});

db.connect(err => {

if (err) {

console.error('Database connection failed:', err);

} else {

console.log('Connected to MySQL Database');

}

});

// User Registration

app.post('/register', async (req, res) => {

const { username, email, password } = req.body;

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

db.query('INSERT INTO users (username, email, password) VALUES (?, ?, ?)',

[username, email, hashedPassword],

(err, result) => {

if (err) return res.status(500).json({ error: err });

res.json({ message: 'User registered successfully' });

}

);

});

// User Login

app.post('/login', (req, res) => {

const { email, password } = req.body;

db.query('SELECT \* FROM users WHERE email = ?', [email], async (err, results) => {

if (err) return res.status(500).json({ error: err });

if (results.length === 0) return res.status(401).json({ error: 'Invalid credentials' });

const user = results[0];

const passwordMatch = await bcrypt.compare(password, user.password);

if (!passwordMatch) return res.status(401).json({ error: 'Invalid credentials' });

const token = jwt.sign({ id: user.id }, SECRET\_KEY, { expiresIn: '1h' });

res.json({ token });

});

});

// Create Resume

app.post('/resume', (req, res) => {

const { userId, name, email, phone, skills, experience, education } = req.body;

db.query('INSERT INTO resumes (userId, name, email, phone, skills, experience, education) VALUES (?, ?, ?, ?, ?, ?, ?)',

[userId, name, email, phone, skills, experience, education],

(err, result) => {

if (err) return res.status(500).json({ error: err });

res.json({ message: 'Resume created successfully' });

}

);

});

// Get Resume

app.get('/resume/:userId', (req, res) => {

const { userId } = req.params;

db.query('SELECT \* FROM resumes WHERE userId = ?', [userId], (err, results) => {

if (err) return res.status(500).json({ error: err });

res.json(results);

});

});

app.listen(PORT, () => {

console.log(Server running on port ${PORT});

});