const express = require('express');

const puppeteer = require('puppeteer');

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

app.use(cors());

app.use(bodyParser.json());

// Resume templates

const templates = {

"template1": `

<html>

<head><title>Resume</title></head>

<body>

<h1>{{name}}</h1>

<p>{{jobTitle}}</p>

<p>{{email}} | {{phone}}</p>

</body>

</html>

`,

"template2": `

<html>

<head><title>Resume</title></head>

<body>

<h2>Professional Resume</h2>

<h1>{{name}}</h1>

<h3>{{jobTitle}}</h3>

<p>Contact: {{email}} | {{phone}}</p>

</body>

</html>

`

};

app.post('/generate-pdf', async (req, res) => {

try {

const { templateType, name, jobTitle, email, phone } = req.body;

const selectedTemplate = templates[templateType] || templates["template1"];

// Replace placeholders with user data

const htmlContent = selectedTemplate

.replace("{{name}}", name)

.replace("{{jobTitle}}", jobTitle)

.replace("{{email}}", email)

.replace("{{phone}}", phone);

// Launch Puppeteer

const browser = await puppeteer.launch({ args: ['--no-sandbox', '--disable-setuid-sandbox'], headless: true });

const page = await browser.newPage();

await page.setContent(htmlContent);

// Generate PDF

const pdfBuffer = await page.pdf({ format: 'A4' });

await browser.close();

// Send the PDF as a response

res.setHeader('Content-Type', 'application/pdf');

res.setHeader('Content-Disposition', 'attachment; filename="generated.pdf"');

res.end(pdfBuffer);

} catch (error) {

console.error("PDF generation error:", error);

res.status(500).send("Failed to generate PDF");

}

});

const PORT = 3000;

app.listen(PORT, () => console.log(`Server running on [http://localhost:${PORT}`)](http://localhost:$%7bPORT%7d%60)));

**How It Works:**

1. The server has multiple resume templates stored in an object.
2. The user sends a request with a templateType (e.g., "template1", "template2").
3. The selected template is filled with user data.
4. Puppeteer generates a PDF based on the selected template.
5. The generated PDF is sent as a response.