```
from zipfile import ZipFile
import os

# Define the project structure
project_files = {
    "server.js": """
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const path = require('path');
const app = express();

mongoose.connect('mongodb://localhost:27017/recruitmentDB', {
    useNewUrlParser: true,
    useUnifiedTopology: true
});

const CV = require('./models/CV');
const Vacancy = require('./models/Vacancy');
const Inquiry = require('./models/Inquiry');

app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.json());
app.use(express.static(path.join(__dirname, 'public')));

app.post('/submit-cv', async (req, res) => {
    const newCV = new CV(req.body);
    await newCV.save();
    res.send({ success: true });
});

app.get('/vacancies', async (req, res) => {
    const vacancies = await Vacancy.find({ isFilled: false });
    res.send(vacancies);
});

app.post('/apply/:id', async (req, res) => {
    await Vacancy.findByIdAndUpdate(req.params.id, { isFilled: true });
    res.send({ success: true });
});

app.post('/client-inquiry', async (req, res) => {
    const inquiry = new Inquiry(req.body);
    await inquiry.save();
    res.send({ success: true });
});
```

```
const PORT = 3000;
app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));
""",

    "models/CV.js": """
const mongoose = require('mongoose');
const cvSchema = new mongoose.Schema({
    name: String,
    email: String,
    skills: String
});
module.exports = mongoose.model('CV', cvSchema);
""",

    "models/Vacancy.js": """
const mongoose = require('mongoose');
const vacancySchema = new mongoose.Schema({
    title: String,
    description: String,
    isFilled: { type: Boolean, default: false }
});
module.exports = mongoose.model('Vacancy', vacancySchema);
""",

    "models/Inquiry.js": """
const mongoose = require('mongoose');
const inquirySchema = new mongoose.Schema({
    companyName: String,
    message: String
});
module.exports = mongoose.model('Inquiry', inquirySchema);
""",

    "public/index.html": """
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Recruitment Portal</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>Recruitment Portal</h1>

    <section>
```

```
        <h2>Submit Your CV</h2>
        <form id="cvForm">
            <input type="text" name="name" placeholder="Your name" required>
            <input type="email" name="email" placeholder="Your email" required>
            <textarea name="skills" placeholder="Your skills" required></textarea>
            <button type="submit">Submit CV</button>
        </form>
    </section>

    <section>
        <h2>Open Vacancies</h2>
        <div id="vacancyList"></div>
    </section>

    <section>
        <h2>Client Hiring Inquiry</h2>
        <form id="inquiryForm">
            <input type="text" name="companyName" placeholder="Company
name" required>
            <textarea name="message" placeholder="Your message" required></
textarea>
            <button type="submit">Submit Inquiry</button>
        </form>
    </section>

    <script src="script.js"></script>
</body>
</html>
""",

    "public/styles.css": """
body {
    font-family: Arial;
    margin: 20px;
}
form, section {
    margin-bottom: 30px;
}
input, textarea {
    display: block;
    margin: 10px 0;
    padding: 8px;
    width: 100%;
    max-width: 400px;
}
button {
    padding: 10px 20px;
```

```
}
""",

    "public/script.js": """
document.getElementById('cvForm').addEventListener('submit', async function(e) {
    e.preventDefault();
    const data = Object.fromEntries(new FormData(this));
    const res = await fetch('/submit-cv', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(data)
    });
    if (res.ok) alert('CV Submitted!');
});

async function loadVacancies() {
    const res = await fetch('/vacancies');
    const vacancies = await res.json();
    const container = document.getElementById('vacancyList');
    container.innerHTML = '';
    vacancies.forEach(v => {
        const div = document.createElement('div');
        div.innerHTML = `<h3>${v.title}</h3><p>${v.description}</p><button onclick="apply('${v._id}')">Apply</button>`;
        container.appendChild(div);
    });
}
loadVacancies();

async function apply(id) {
    const res = await fetch(`/apply/${id}`, { method: 'POST' });
    if (res.ok) {
        alert('Applied successfully!');
        loadVacancies();
    }
}

document.getElementById('inquiryForm').addEventListener('submit', async function(e) {
    e.preventDefault();
    const data = Object.fromEntries(new FormData(this));
    const res = await fetch('/client-inquiry', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(data)
    });
    if (res.ok) alert('Inquiry submitted!');
});
"""
}

# Create the ZIP file
zip_filename = "/mnt/data/recruitment_portal.zip"
with ZipFile(zip_filename, 'w') as zipf:
    for filepath, content in project_files.items():
        full_path = os.path.join("/tmp", filepath)
        os.makedirs(os.path.dirname(full_path), exist_ok=True)
        with open(full_path, 'w') as f:
            f.write(content)
        zipf.write(full_path, filepath)
```