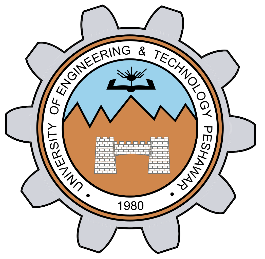
**PROJECT REPORT**

**JOB PORTAL**

**Spring 2025**

Submitted by: **Mohsin Sajjad, Muhammad Hassan, Muhammad Afnan Khan**

Registration No: **22pwsce2149,22pwcse2105,22pwcse2155**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr. Summeya Salahudin**

Month Day, Year (04 07 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Project Title**

**JOB SPHERE – Laravel 12 Based Job Portal**

# **Objective:**

To develop a fully functional and user-friendly job portal using Laravel 12 that connects job seekers with employers, allowing job applications, job listings, resume uploads, and admin-level employer verification.

# **Tools & Technologies:**

- Framework: Laravel 12 (PHP)  
- Frontend: Blade Templating, Bootstrap 5, HTML/CSS  
- Database: MariaDB/MySQL  
- Server: Localhost (XAMPP)  
- Version Control: GitHub  
- Deployment Platform: Railway (Free Hosting)

- Free hosting (aiven.io)

**1. Entity Descriptions:**

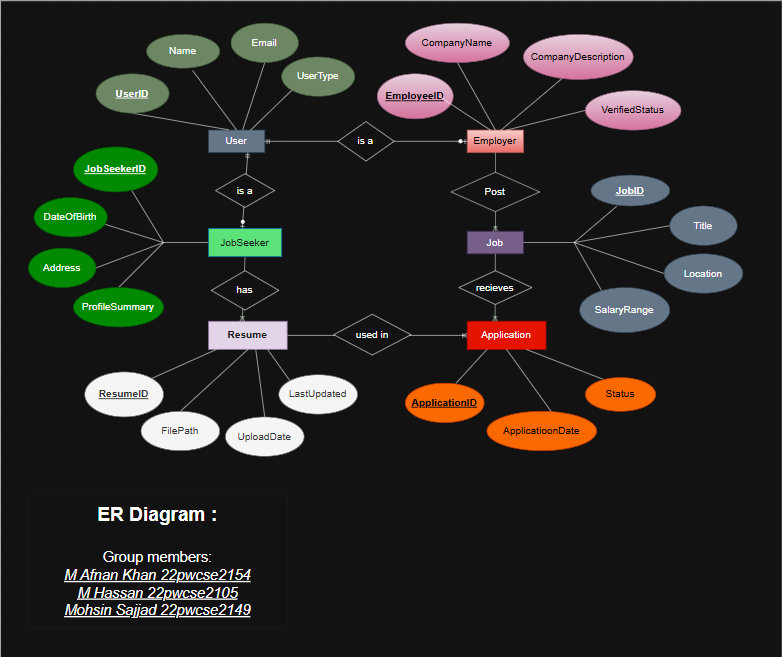
Below is a summary of the core entities, their primary keys, and main attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Primary Key** | **Key Attributes** | **Description** |
| **User** | UserID | Name, Email, Password, UserType | Stores common account data for both JobSeekers and Employers. |
| **JobSeeker** | SeekerID | UserID (FK), DateOfBirth, Address, ProfileSummary | Subtype of User; holds seeker-specific profile details. |
| **Employer** | EmployerID | UserID (FK), CompanyName, CompanyDescription, CompanyWebsite, VerifiedStatus | Subtype of User; holds employer-specific company details. |
| **Resume** | ResumeID | SeekerID (FK), FilePath, UploadDate, LastUpdated | Stores uploaded resumes for JobSeekers. |
| **Job** | JobID | EmployerID (FK), Title, Description, Requirements, Location, SalaryRange, PostedDate | Represents job listings posted by Employers. |
| **Application** | ApplicationID | JobID (FK), SeekerID (FK), ResumeID (FK), AppliedDate, Status | Captures a JobSeeker’s application to a specific Job with one resume. |

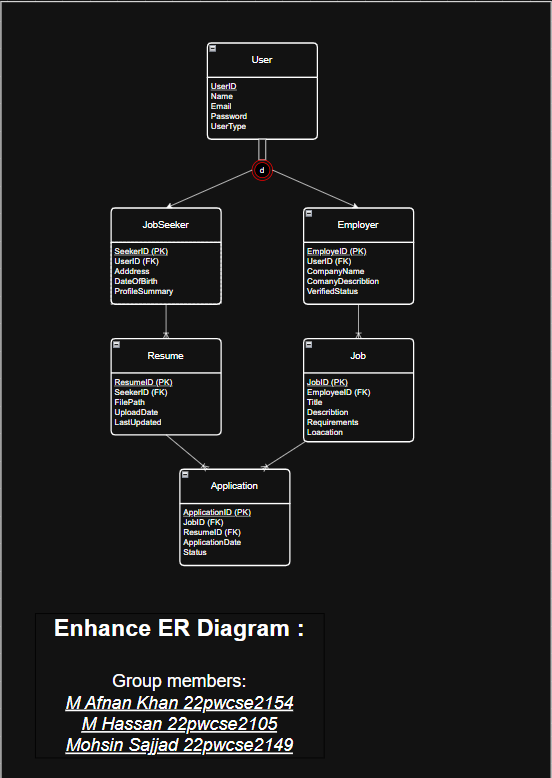
**2. Detailed Business Rules:**

1. **User registration**: Every account must register as either a JobSeeker or an Employer; the UserType attribute enforces this.
2. **Unique email**: The Email in **User** must be unique and validated before account activation.
3. **Email verification**: Only Employers with *VerifiedStatus = TRUE* may post Job listings.
4. **Job postings**: An Employer may post zero or more Jobs; each Job must reference exactly one Employer.
5. **Resume upload**: A JobSeeker may upload multiple Resumes, but only one active Resume may be associated per Application.
6. **Profile completeness**: JobSeekers must complete mandatory fields (*DateOfBirth, Address, Profile Summary*) before applying to any Job.
7. **Application constraint**: A JobSeeker may apply to the same Job at most once; duplicate applications are disallowed.
8. **Application status**: Each Application’s Status must be one of: Pending, Reviewed, Interview, Offered, Rejected.
9. **Posting requirements**: Each Job must include at minimum: Title, Description, Requirements, Location, and Posted Date.
10. **Automatic timestamps**: Applied Date (*Application*) and Posted Date (*Job*) default to the current system date when the record is created.
11. **Resume updates**: When a JobSeeker updates a Resume, the old version may be retained for audit but only the latest may be linked to new Applications.
12. **Deletion rules**: Deleting a User cascades to delete associated JobSeeker/Employer sub-records; deleting an Employer prevents deletion if open Job listings exist.

**3. Entity–Relationship Diagram (ERD):**

****

**4. Enhanced ER Diagram (EERD):**

****

**Relational Schema:**

**USER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User\_ID | Name | Email | Password | UserType |

**Employer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Employe\_ID | User\_ID | Company\_Name | Company\_Discription | Company\_website | Verified\_status |

**Jobseekers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Seeker\_ID | User\_ID | Date\_Of\_birth | Address | Profile\_Summary |

**Job**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Job\_ID | Employe\_ID | Title | Description | Requirement | Location | Salary\_range | Posted\_Date |

**Resume**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resume\_ID | Seeker\_ID | File\_Path | Upload\_Date | Last\_Updated |

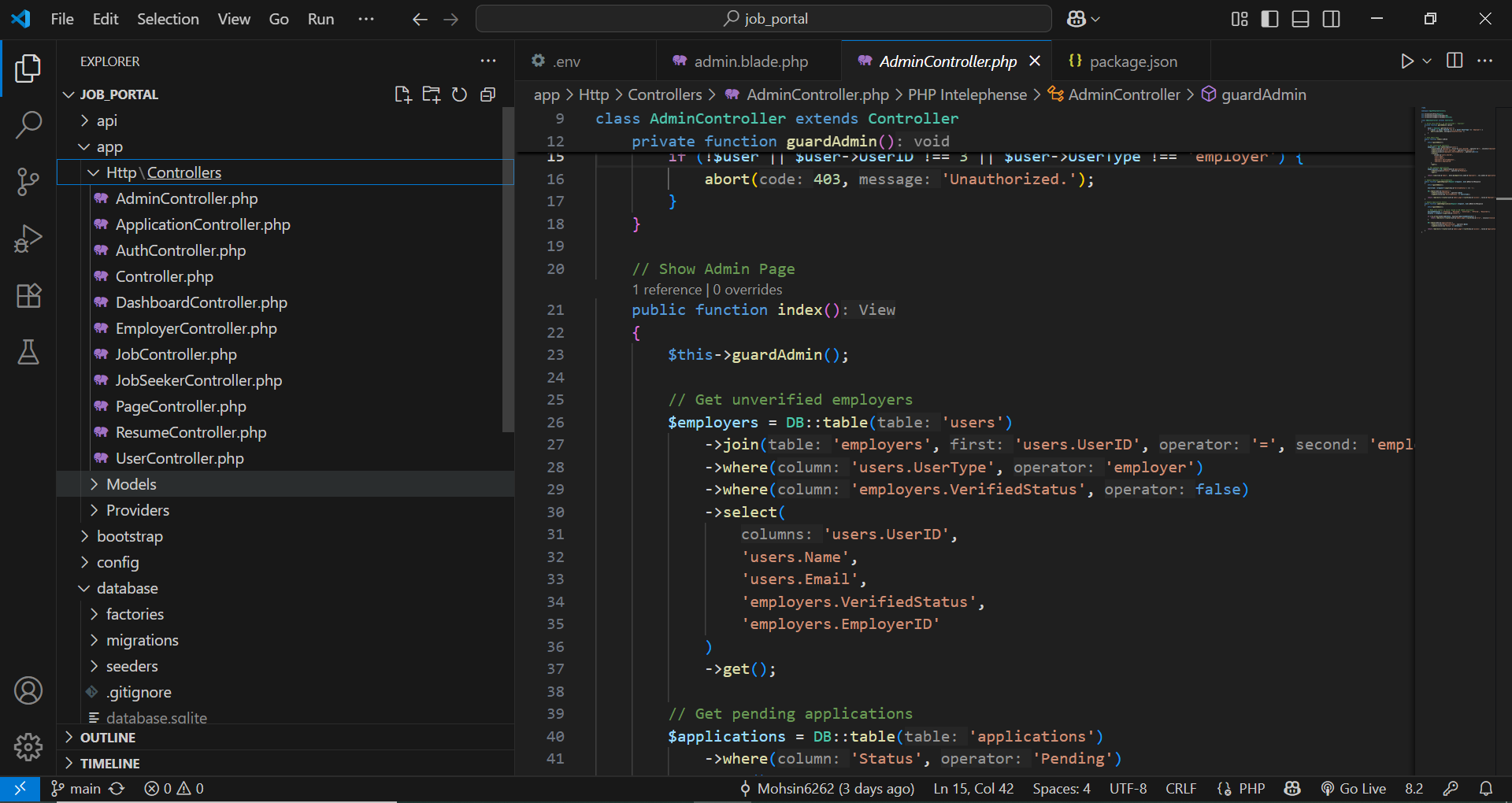
**Application**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Application\_ID | Job\_ID | Seeker\_ID | Resume\_ID | Applied\_Date | Status |

**Implementation on laravel:**

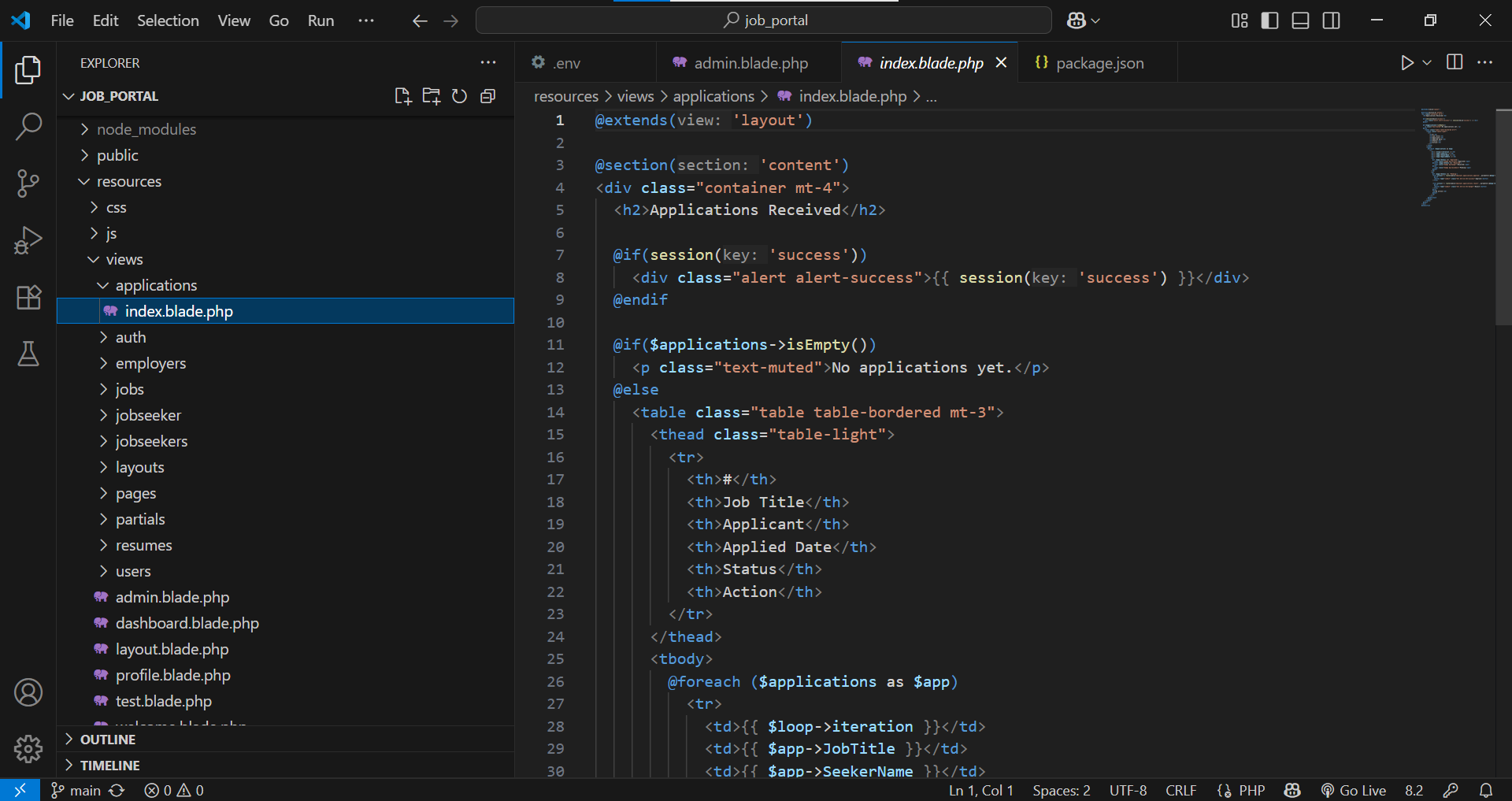
**Controller:**

Backend of our project

****

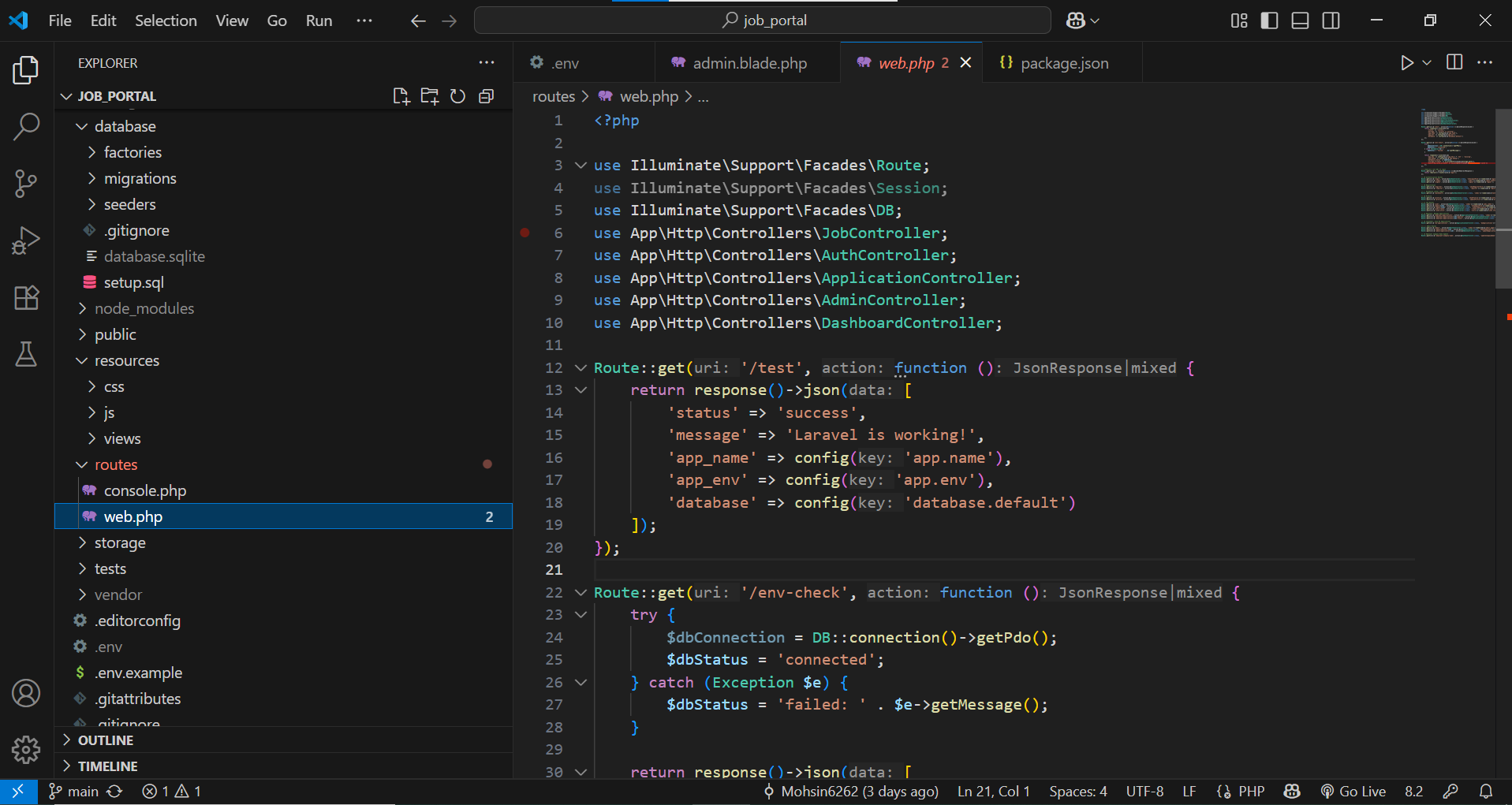
**View:**

Frontend HTML

****

**Routes:**

Maps URLs to controller functions

****

**Deployment:**

**Step 1: Push the project on GitHub:**

cd C:\xampp\htdocs\job\_portal

git init

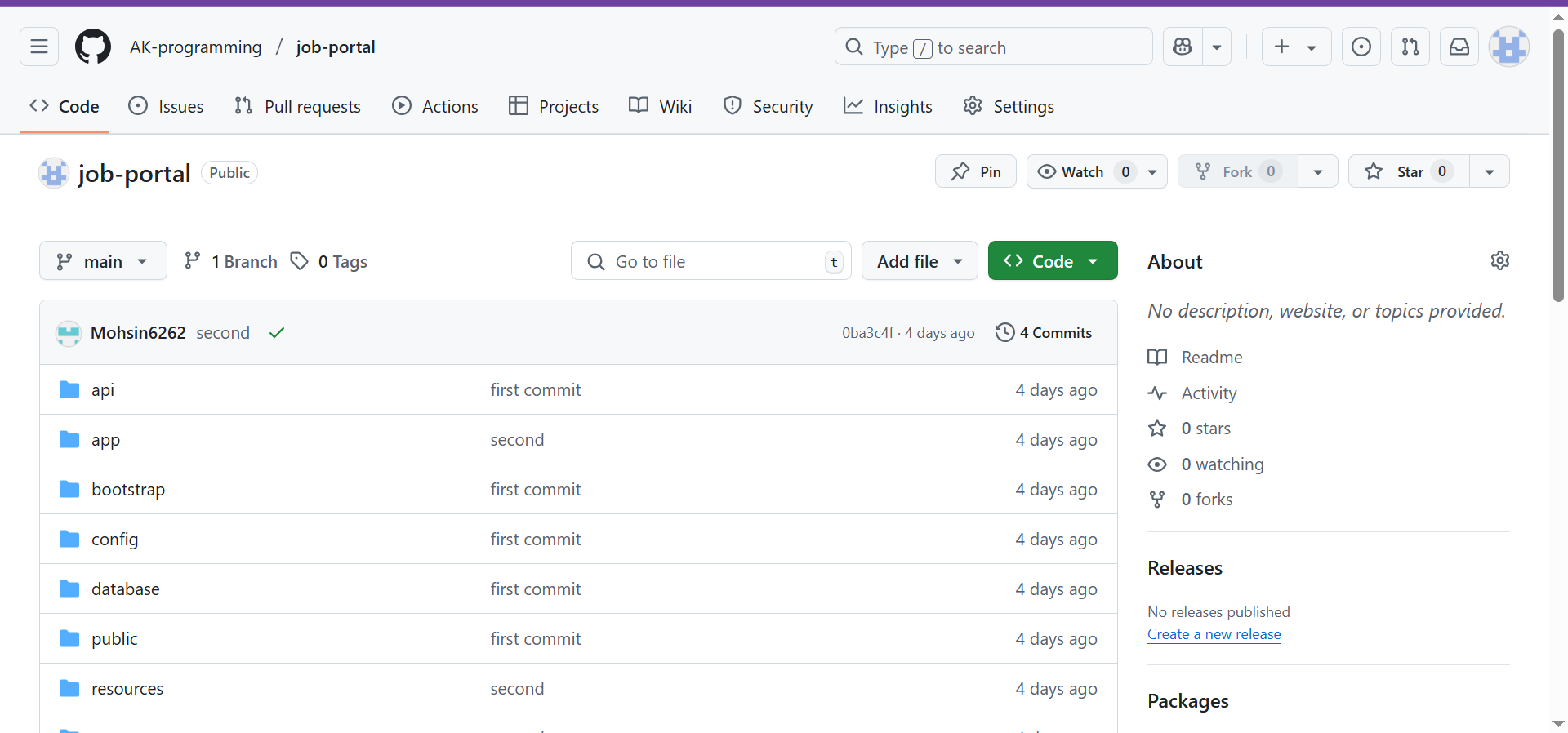
git remote add origin https://github.com/Ak-programming/job\_portal.git

git add .

git commit -m "Pushing project"

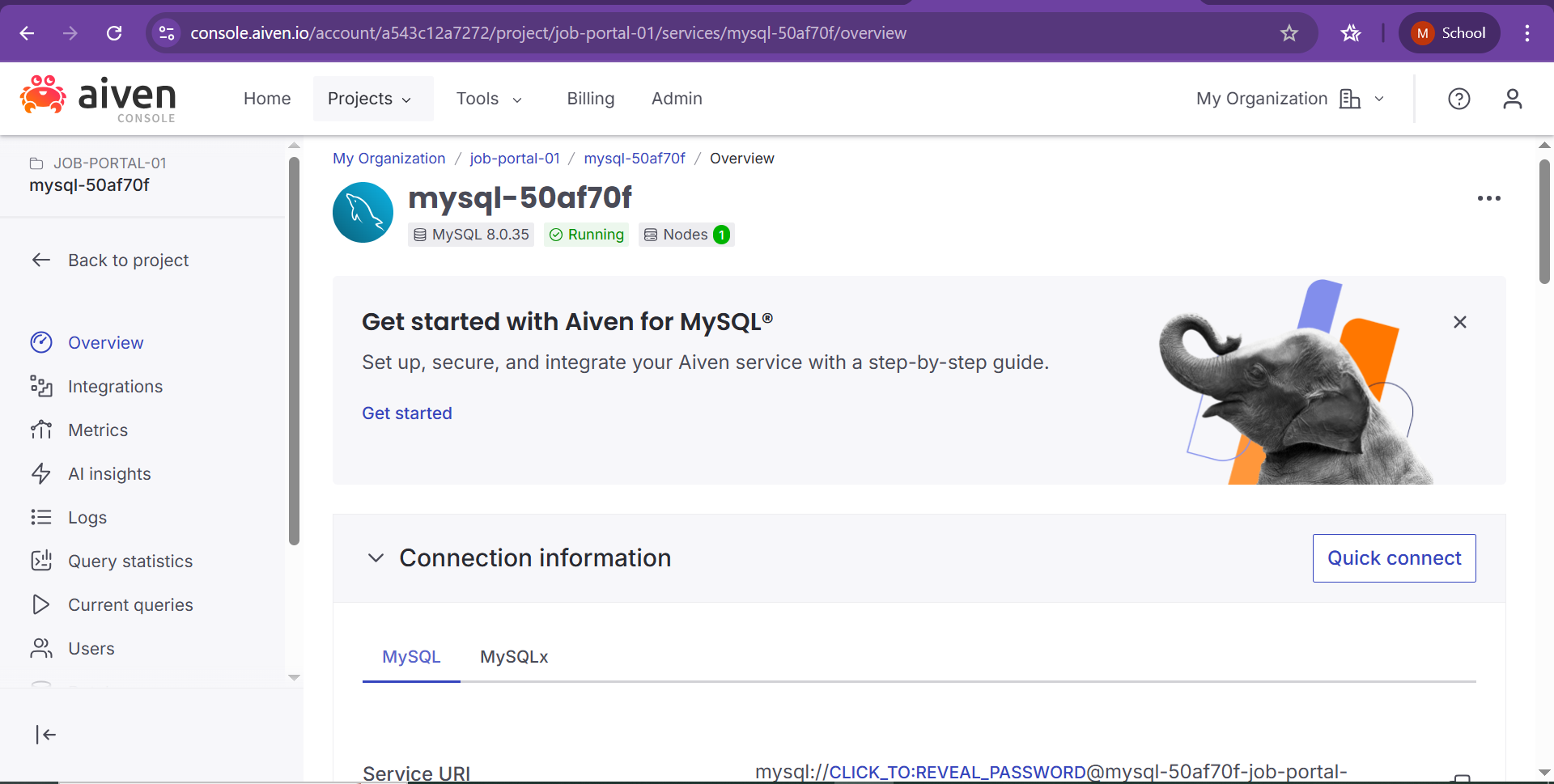
git branch -M main

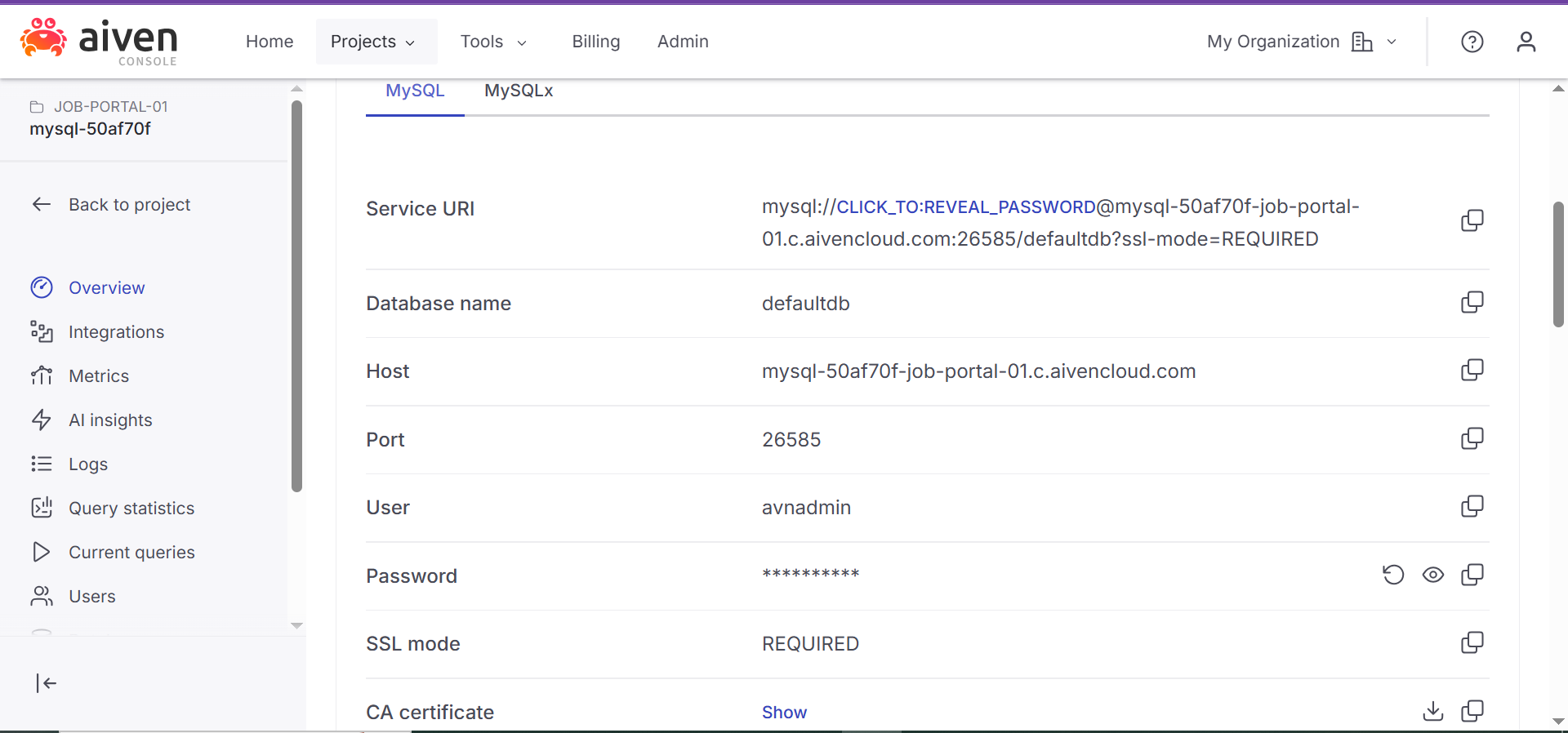
git push -u origin main

****

**Step 2: Create MySQL Database on Aiven.io**

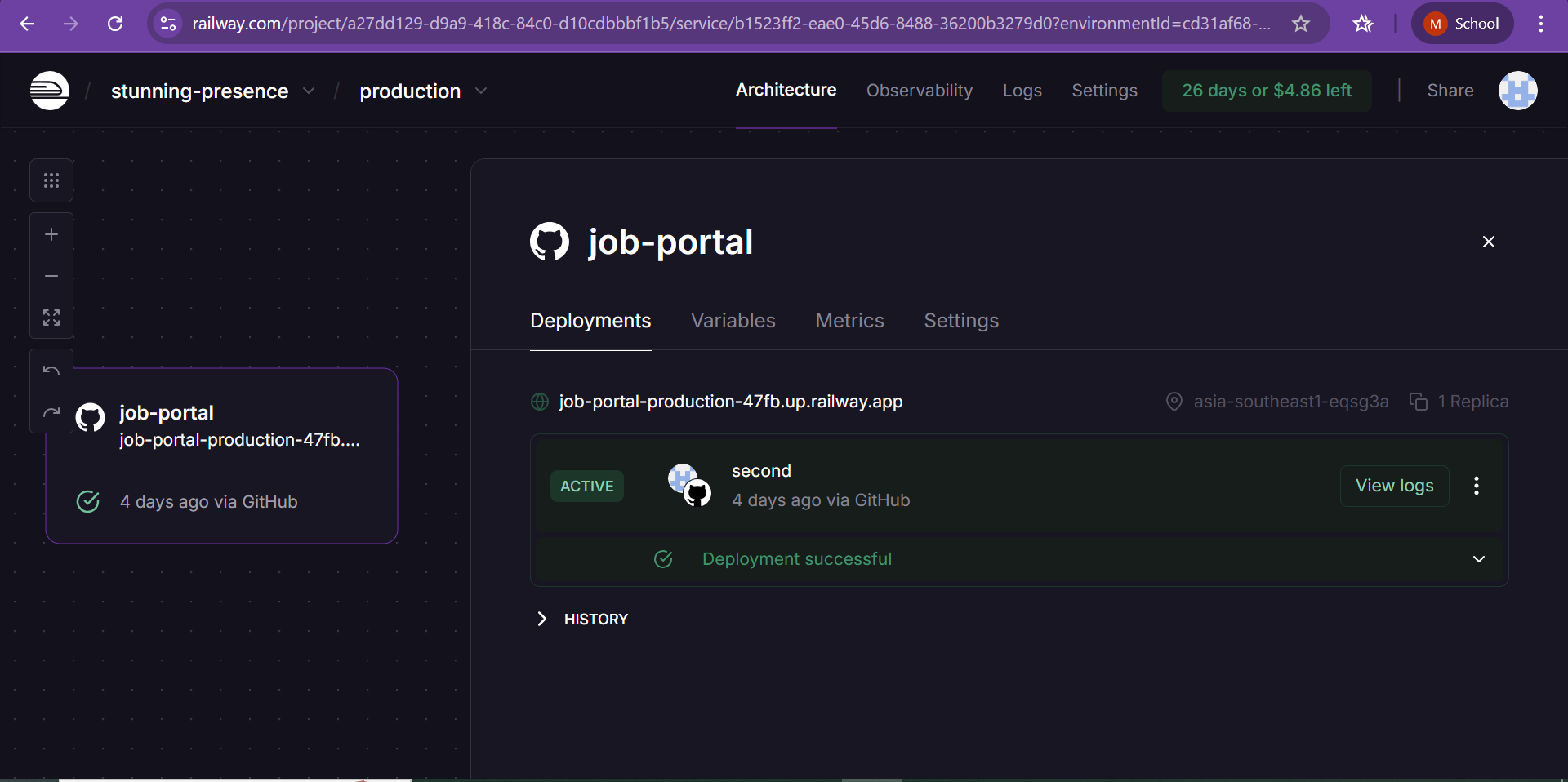
• Sign in to Aiven.io and create a new MySQL service.  
• After creation, note the following credentials:  
 - DB\_HOST  
 - DB\_PORT  
 - DB\_DATABASE  
 - DB\_USERNAME  
 - DB\_PASSWORD

****

****

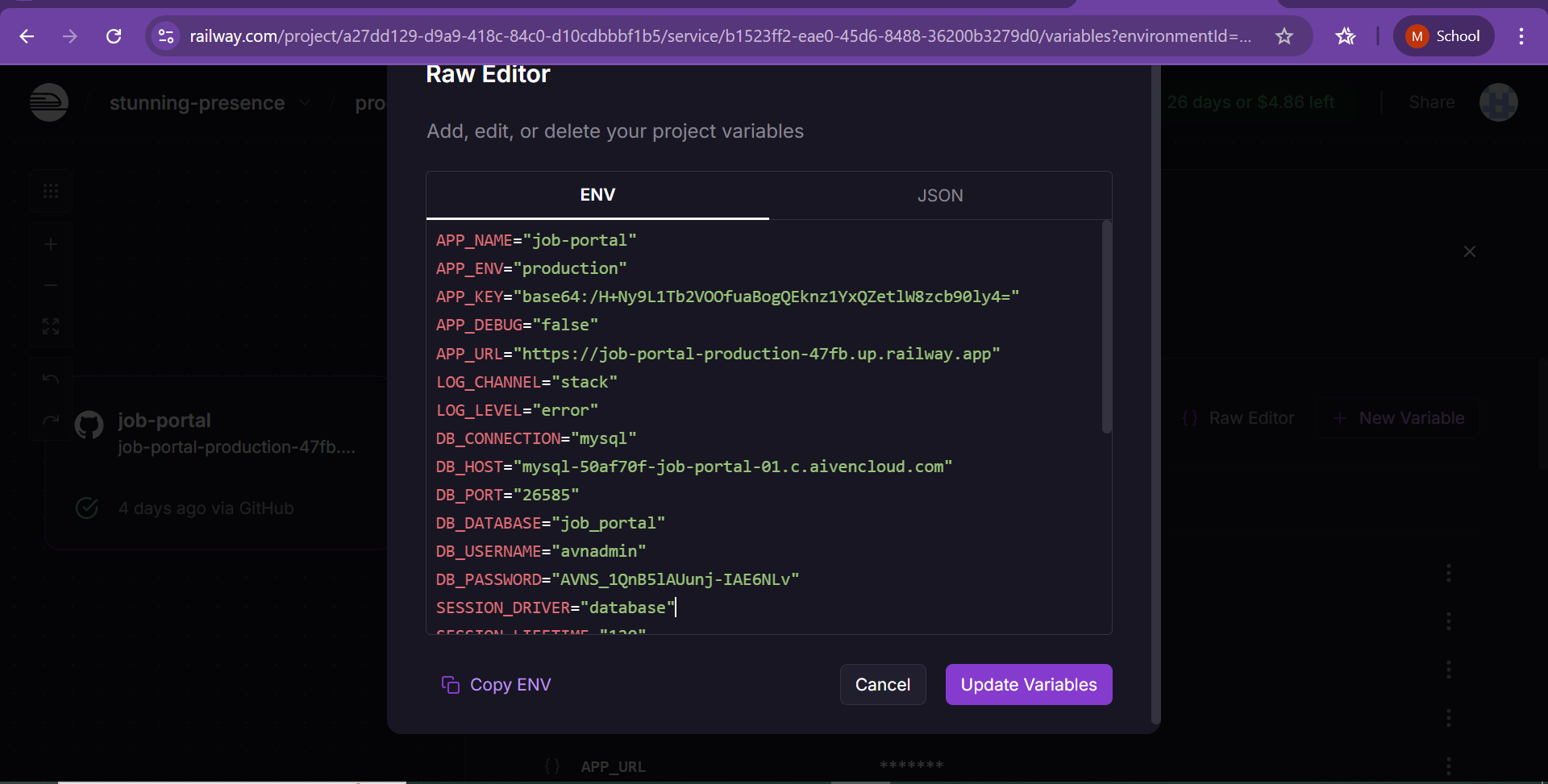
# **Step 3: Deploy Backend to Railway**

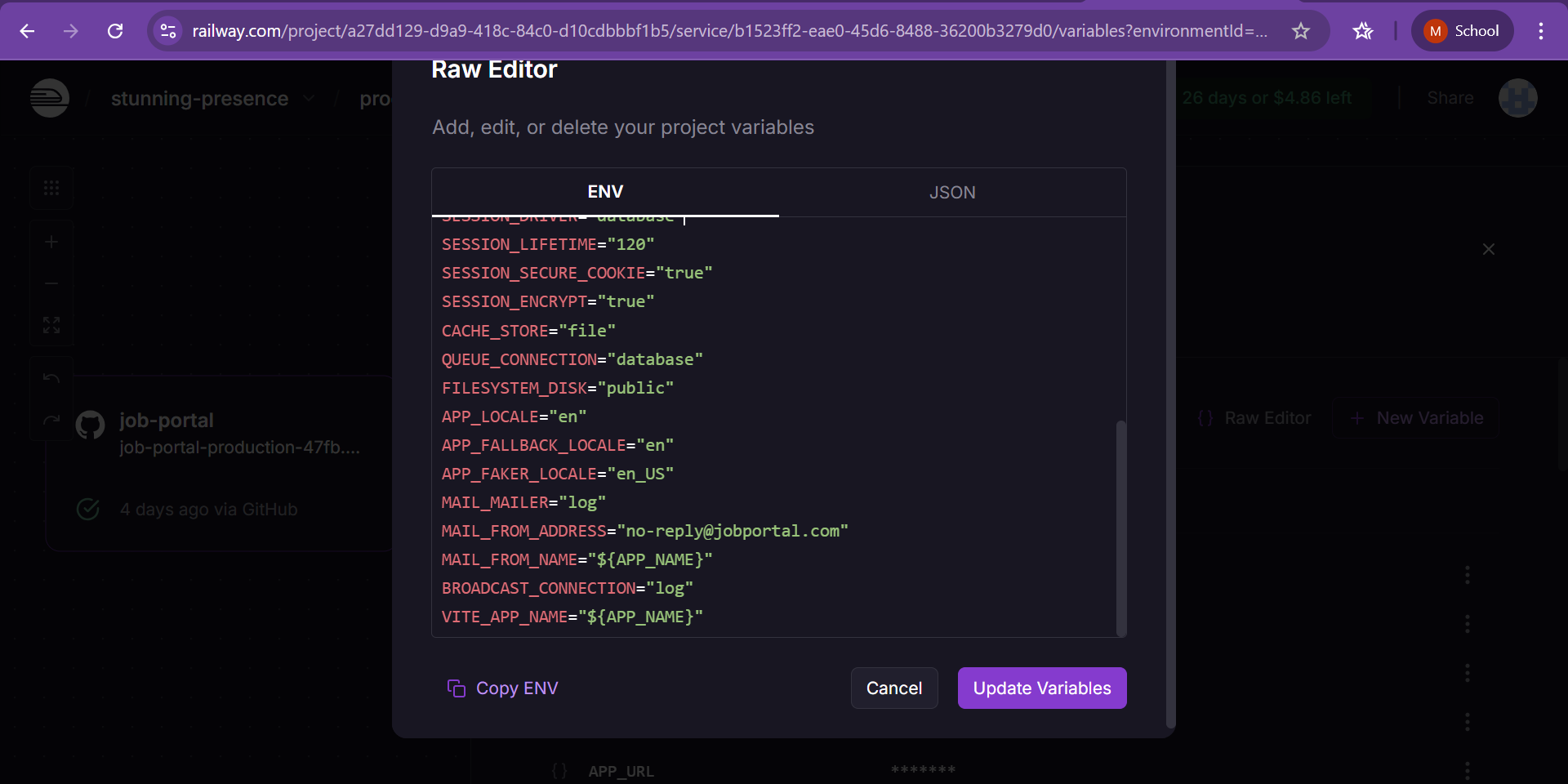
• Go to Railway.app.  
• Click on “New Project” → “Deploy from GitHub”.  
• Select your Laravel project repository



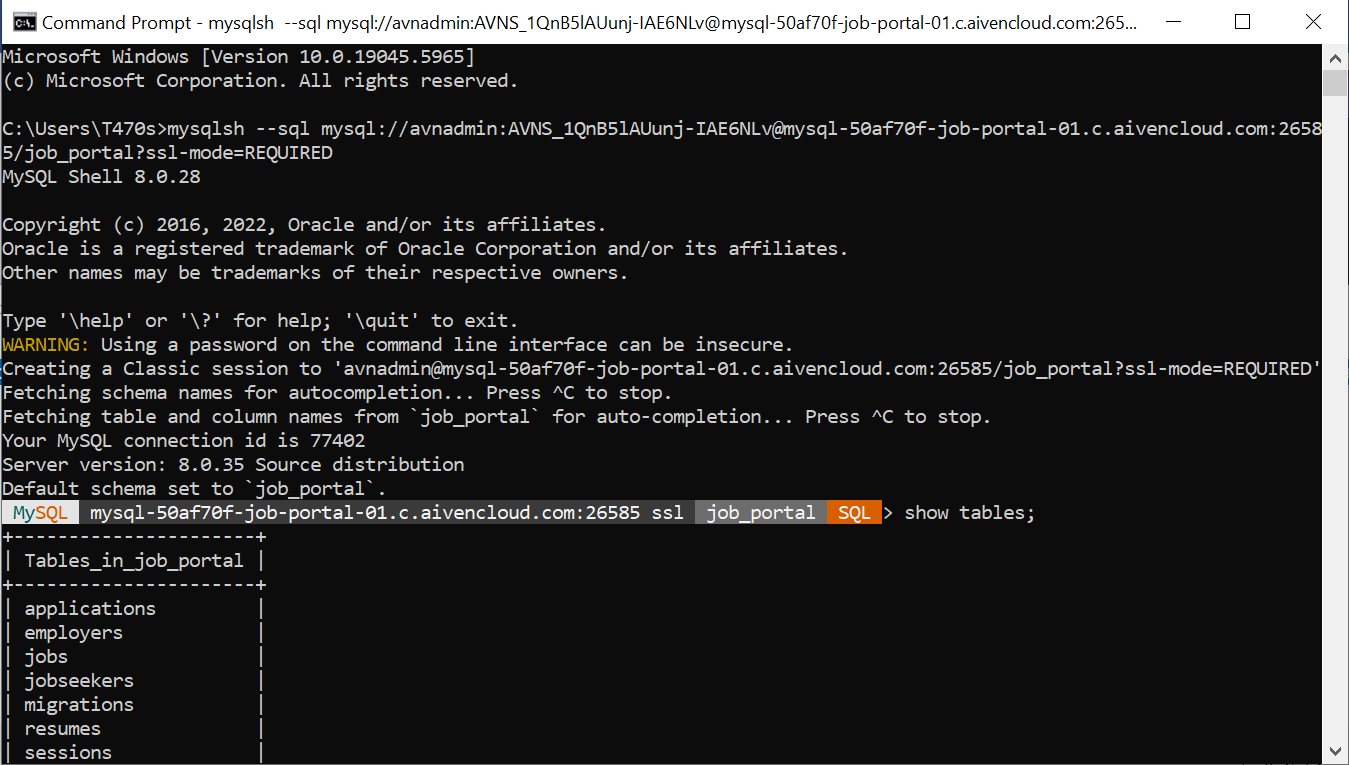
**STEP 4: Configuring Variable**

• After deployment, open the “Variables” tab in Railway.  
• Add the following environment variables (from your .env file):





**STEP 5: ACCESS USING MYSQL CLIENT:**



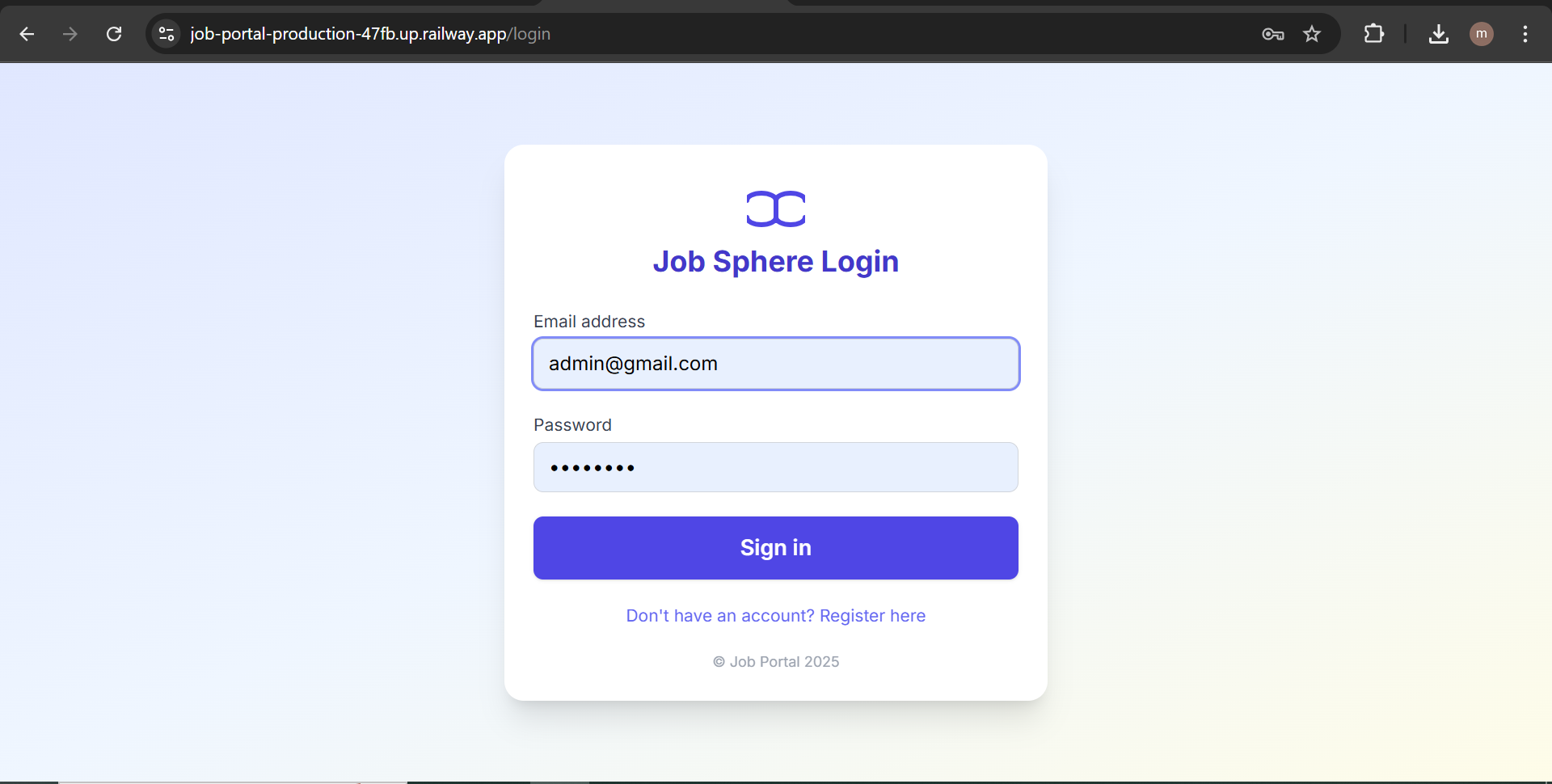
**Deployment link:**

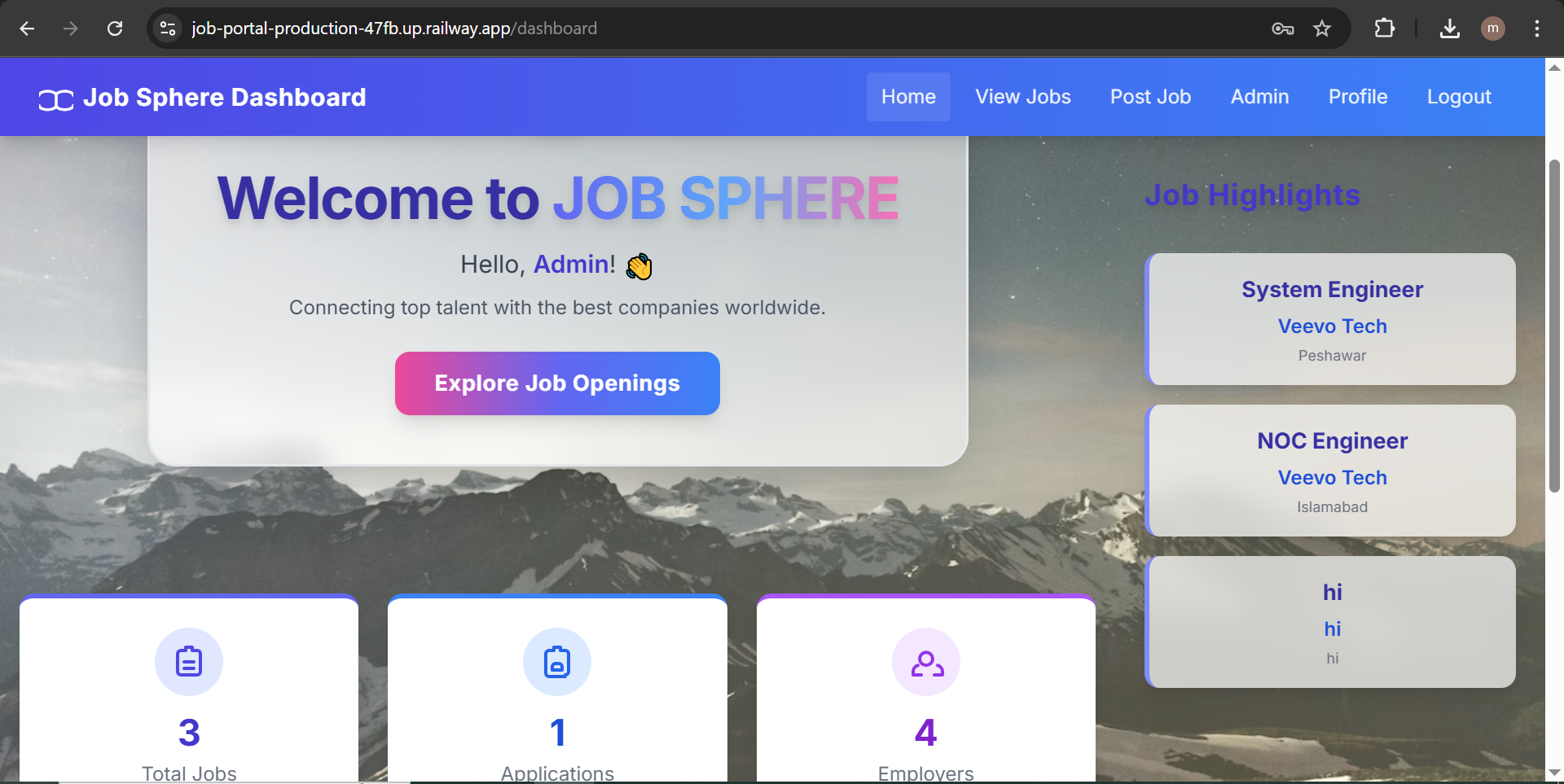
<https://job-portal-production-47fb.up.railway.app>

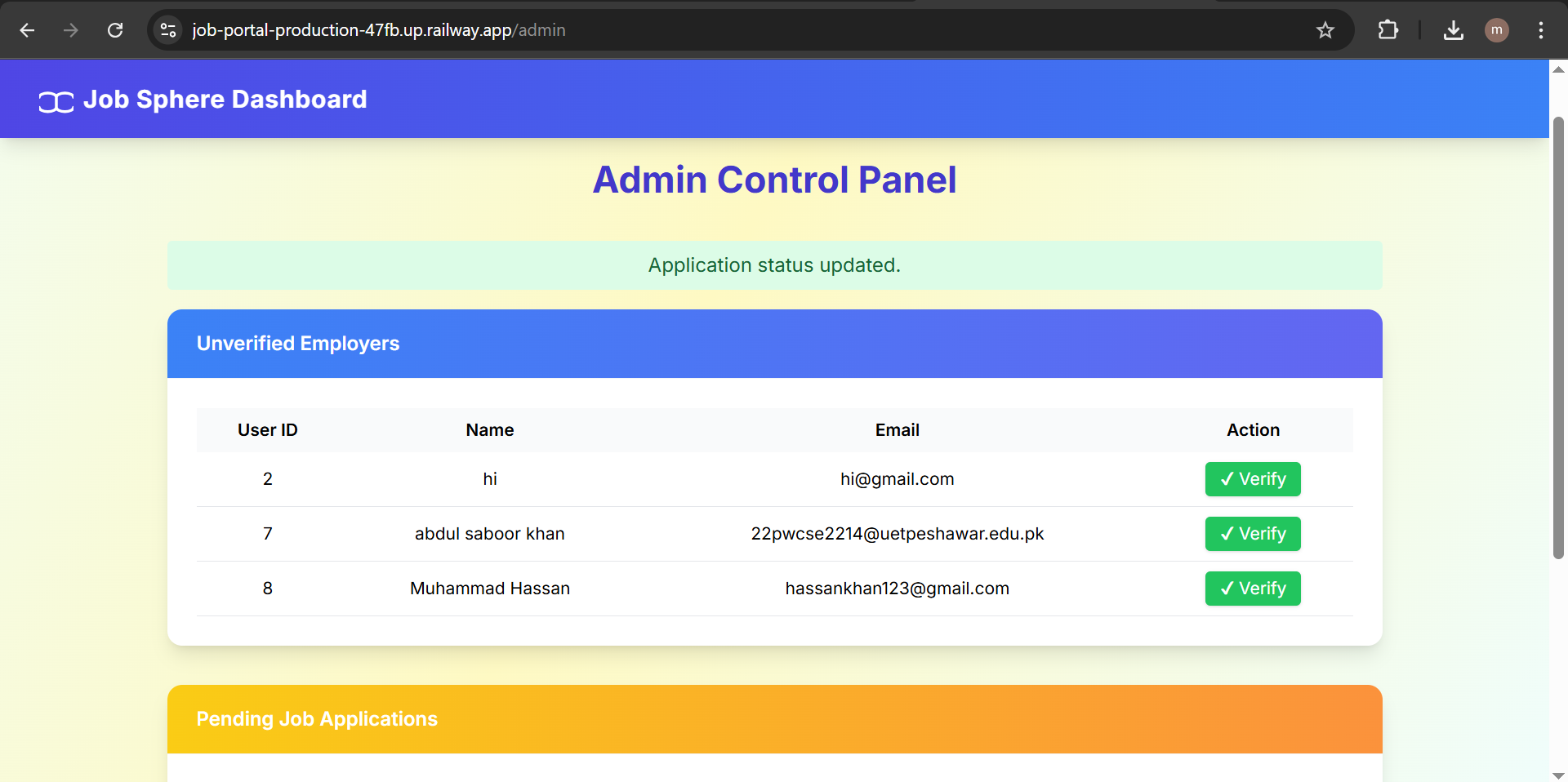
**Function:**

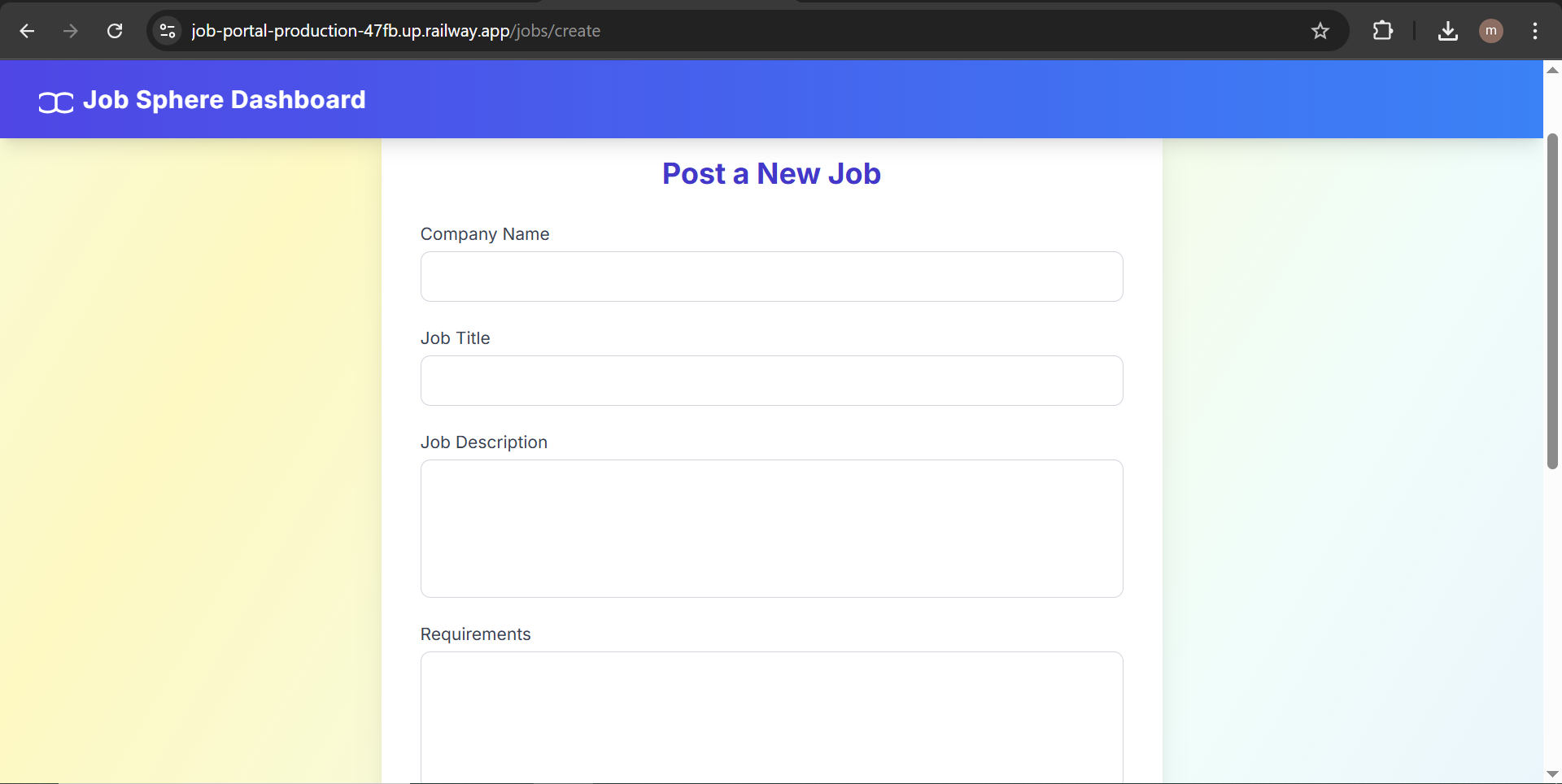
**Admin:**

* Admin can review or reject the applications.
* Admin can verify the employer
* Can post new job.



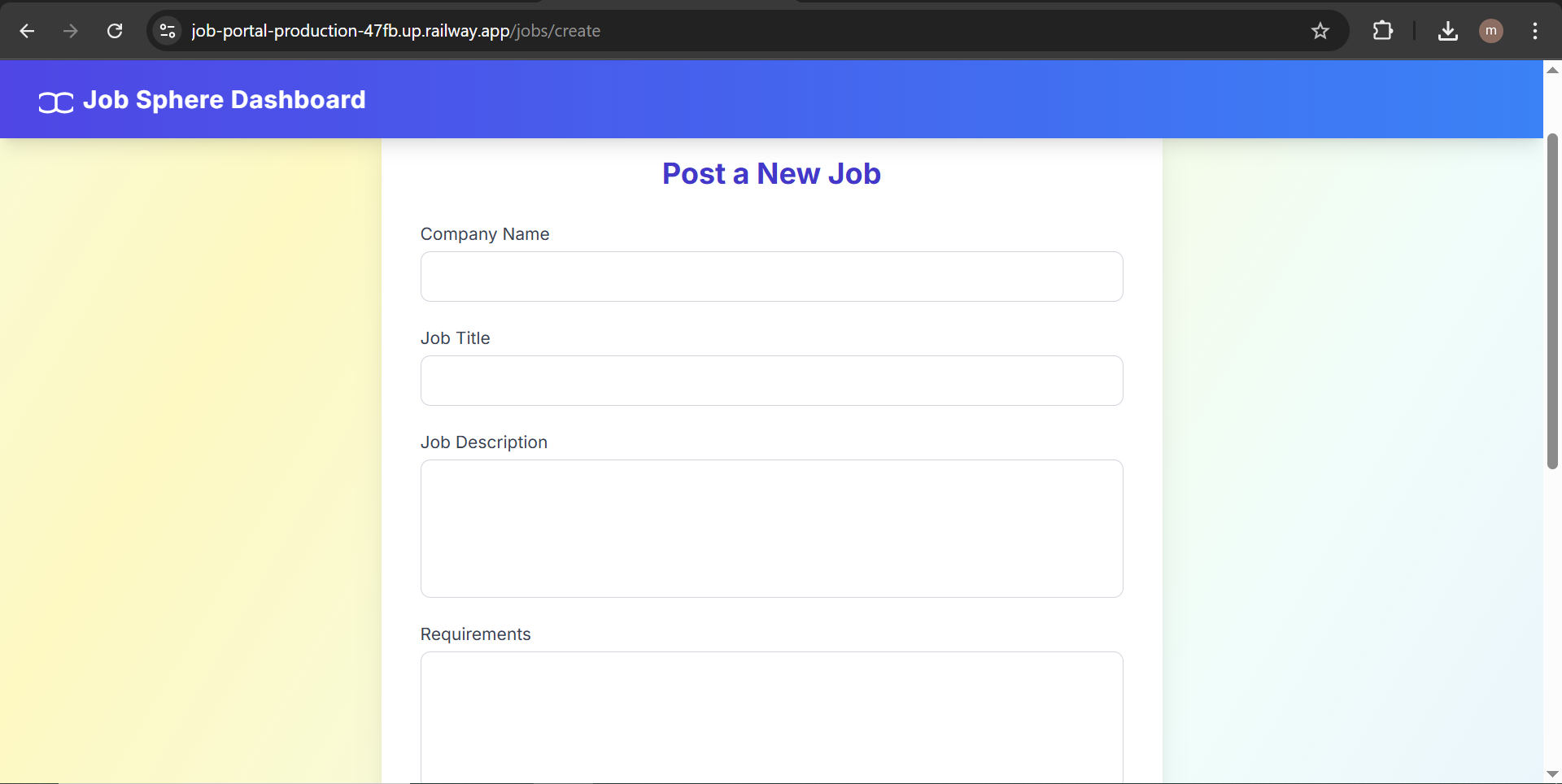


****



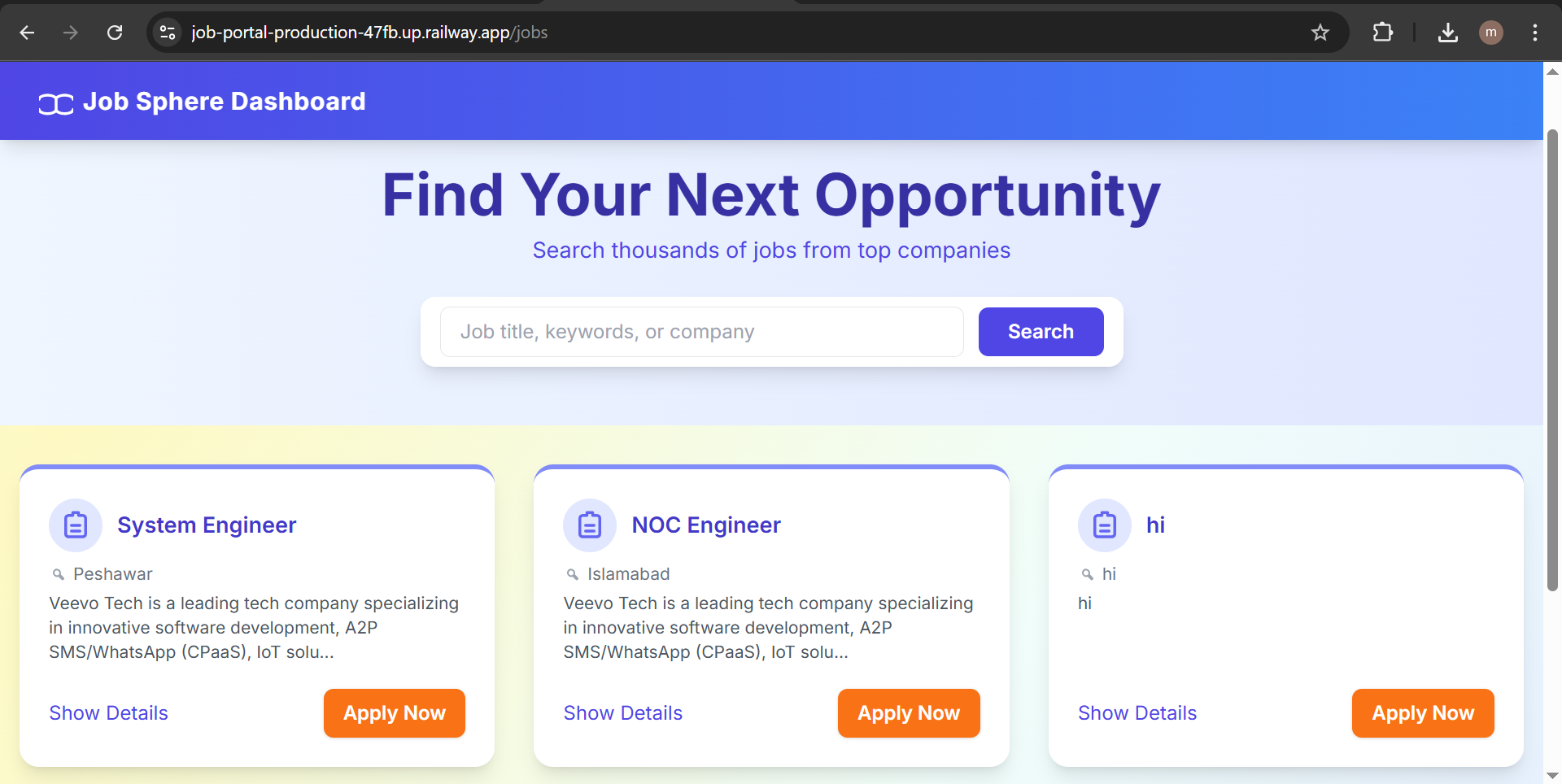
**Employer:**

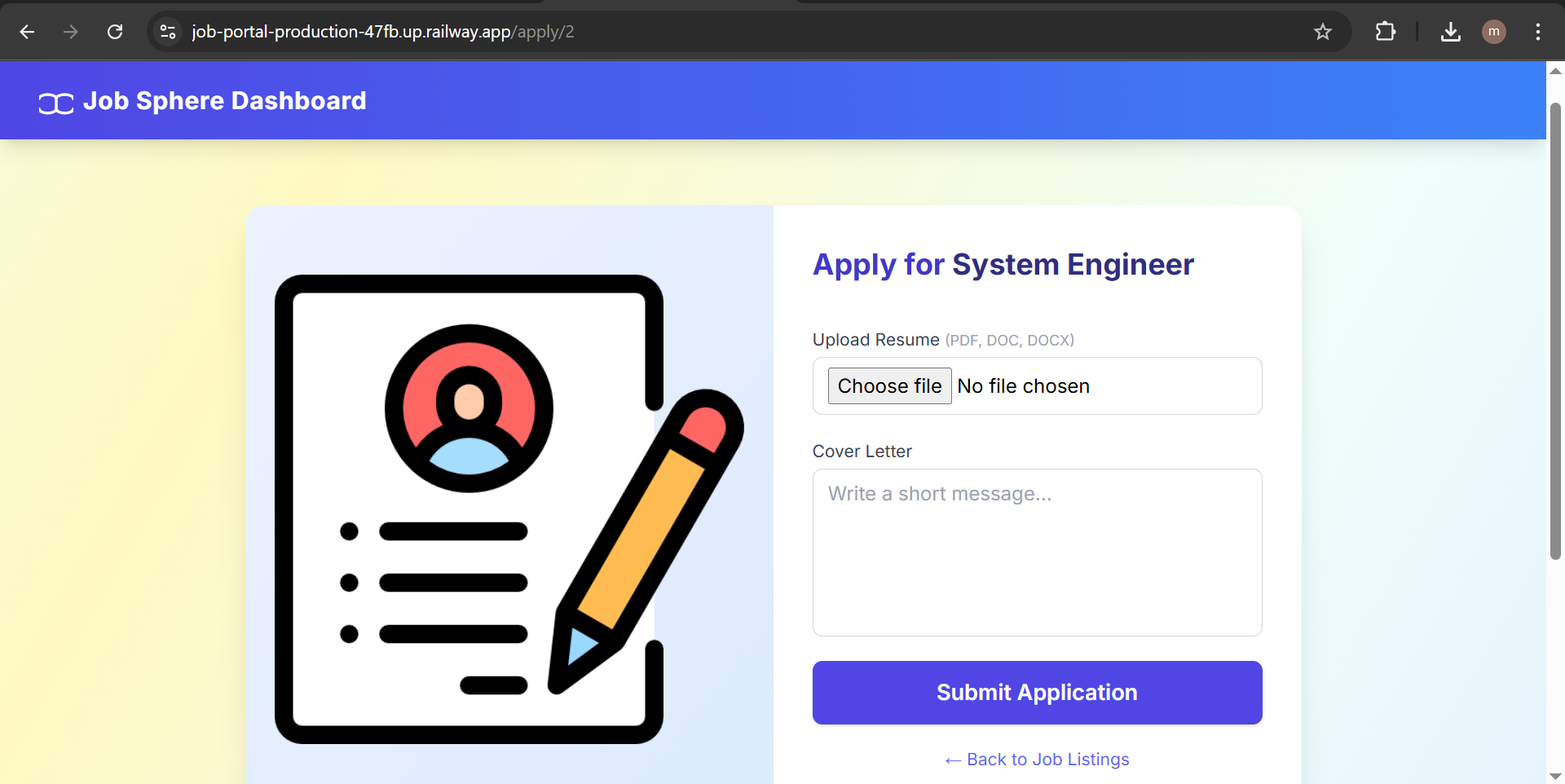
* Employer can post new job.
* Employer itself can apply for new job.
* Can Change the profile info.
* View the job page

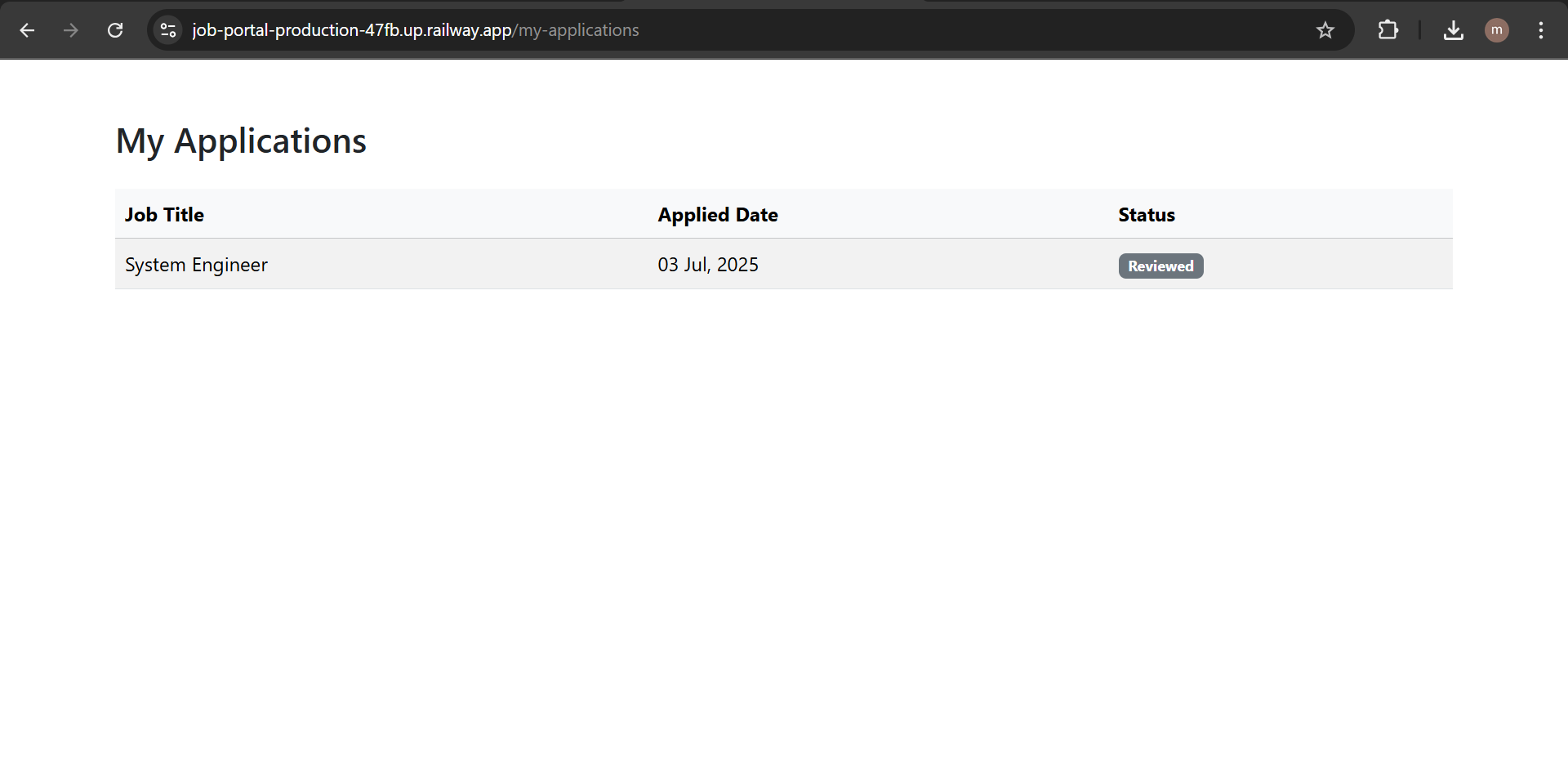


**JOB SEEKER:**

* Job seeker can submit the application for job.
* View the job page.
* Check the application status.
* Change the profile info.





****

### ****Conclusion****

This project provided practical experience with full-stack development using Laravel. The system successfully connects employers and jobseekers with essential features and role-based access. It is scalable, easy to manage, and user-friendly.

**Deployment link:**

<https://job-portal-production-47fb.up.railway.app>

**References**

* OpenAI. (2024). *ChatGPT conversations for code assistance and debugging*. Retrieved from [https://chat.openai.com](https://chat.openai.com/)
* diagrams.net. (n.d.). *Free diagram software*. Retrieved from [https://www.diagrams.net](https://www.diagrams.net/)
* Railway. (n.d.). *Railway hosting documentation*. Retrieved from [https://docs.railway.app](https://docs.railway.app/)
* Aiven. (n.d.). *Database service dashboard for job-portal-01 project*. Retrieved from <https://console.aiven.io/account/a543c12a7272/project/job-portal-01/services>