**VJIT Student Certifications Portal**

**Overview**

The **VJIT Student Certifications Portal** is a web-based application developed using **Python Flask**, designed to manage, track, and analyze students' certification activities department-wise, course-wise, and year-wise. The system enables admins and faculty to upload certifications, generate reports, apply filters, and view analytics.

**Features**

* User Authentication and Session Management
* Role-Based Dashboards (Admin/Faculty)
* Add, Edit, Delete Student Certifications
* Import/Export via CSV and Excel
* Year-wise, Course-wise, Department-wise, Academic Year-wise Filters
* Dashboard Statistics (Total, Completed, Not Completed)
* Visual Analytics via Pie, Bar, and Line Charts using Chart.js
* PDF Report Download
* Hosted Online via Render.com

**Technologies Used**

* **Backend:** Python Flask
* **Frontend:** HTML, CSS, JavaScript (Chart.js)
* **Database:** SQLite
* **Hosting:** Render.com
* **Version Control:** Git, GitHub

**Project Structure**

VJIT\_Student\_Certifications\_Portal/

├── app.py

├── requirements.txt

├── templates/

│ ├── index.html

│ └── login.html

├── static/

│ ├── css/style.css

│ └── Logo.jpg

├── database/

│ └── certifications.db

**Setup Instructions**

**1. Clone the Project**

git clone https://github.com/eswarbanala/VJIT\_Student\_Certifications\_Portal.git

cd VJIT\_Student\_Certifications\_Portal

**2. Install Dependencies (Windows)**

Make sure Python 3.11+ is installed. Then:

pip install -r requirements.txt

**3. Run the Application**

python app.py

Navigate to <http://127.0.0.1:8080/login> in your browser.

**Deployment (Render.com)**

**1. Create a GitHub Repository**

Push your local project code to a public GitHub repo.

**2. Create requirements.txt**

pip freeze > requirements.txt

**3. Create render.yaml (optional for auto deploys)**

**4. Deploy to Render.com**

* Go to [Render.com](https://render.com/)
* Create a new Web Service
* Connect your GitHub repo
* Set build command: pip install -r requirements.txt
* Set start command: python app.py
* Environment: Python 3.11

**Application Screens**

* **Login Page:** User enters credentials.
* **Dashboard:** View summary cards and analytics.
* **Filters:** Select Year, Academic Year, Course, Department.
* **Table View:** Displays all certification entries.
* **Actions:** Edit/Delete certifications.

**Notes**

* The field **Verified** is renamed to **Completed**.
* Charts update dynamically based on filters.
* PDF download option is available.

**Credits**

Developed and maintained by **Eswar Banala**, Associate Professor, VJIT.

For any enhancements or support, contact: [[eswarbanala@gmail.com](mailto:eswarbanala@gmail.com)]

### 🚀 Functional Enhancements

#### 1. **Search Bar**

* Add a global search (name, roll number, course, etc.)
* Live filtering using JavaScript.

#### 2. **Pagination**

* Split the table into pages (e.g., 10 records/page)
* Faster loading for large data sets.

#### 3. **File Upload Viewer**

* Allow uploaded certificate PDFs/images to be stored and **viewed inline or in a modal**.

#### 4. **Multi-Select Filters**

* Instead of single dropdowns, allow multi-select for departments, courses, years.

#### 5. **Notifications / Alerts**

* Notify admin when new certifications are added and need review.
* Display status badges (e.g., ✅ Completed / ❌ Not Completed).

#### 6. **Auto-Verification via OCR/AI**

* Integrate OCR to extract certificate data and auto-verify against inputs.

#### 7. **Student Login Dashboard**

* Allow students to **view their own certifications**, add new ones, and track progress.

### 📊 Dashboard & Analytics Enhancements

#### 1. **Drill-down Charts**

* Click on a department → see students in it with completion rates.

#### 2. **Monthly Trends**

* Track certifications submitted per month.
* Line chart with options to filter by department/course/year.

#### 3. **Admin Heatmap**

* View peak certification submission periods across the year.

### 🛠️ Technical Upgrades

#### 1. **Responsive UI**

* Make it mobile-friendly using CSS flex/grid or frameworks like Bootstrap or Tailwind.

#### 2. **Role Management System**

* Add more roles: Faculty, HODs, Students — with permissions.

#### 3. **Dark Mode Toggle**

* A small toggle switch to switch between light and dark UI.

#### 4. **Export Logs**

* Track who exported data and when — useful for audits.

### 🗃️ Data Management Features

#### 1. **Backup & Restore**

* Option for admin to back up and restore the entire certification dataset.

#### 2. **Change Log / History**

* Record of edits (who edited what and when).

#### 3. **Email Reports**

* Schedule a report to be emailed to HODs or admin weekly/monthly.