# Student Portal Web App Documentation

**Project Overview**

This project is a simple **Student Portal Web Application** built using:

* **Frontend**: HTML, CSS, JavaScript
* **Backend**: Node.js with Express
* **Database**: PostgreSQL (hosted on Render)
* **Hosting**: Render (for both Web Service and Database)

Users can:

* Sign up for an account
* Log in to their account
* Submit student enquiry forms

**Folder Structure**

Web\_App/

|-- Public/ # Static frontend files

| |-- index.html # Login page

| |-- signup.html # Signup page

| |-- form.html # Enquiry form page

|-- server.js # Express backend server

|-- package.json # Node.js dependencies

**Step-by-Step Instructions**

**1. Frontend Pages**

**index.html (Login Page)**

* A form that takes email and password
* Submits to /login endpoint using POST method

**signup.html**

* A form that collects fullname, email, and password
* Submits to /signup endpoint

**form.html**

* A detailed enquiry form capturing multiple fields like name, contact info, education, etc.
* Submits to /submit-enquiry

**2. Backend - server.js**

* Sets up Express server
* Connects to PostgreSQL using pg package
* Handles 3 routes:
  + POST /signup: Register user
  + POST /login: Authenticate user
  + POST /submit-enquiry: Store enquiry data

**Environment Variables Required:**

DATABASE\_URL=<your PostgreSQL connection URL from Render>

**3. Database Setup on Render**

* Go to [https://render.com](https://render.com/)
* Create a new PostgreSQL database
* Note the DATABASE\_URL string

**Tables to Create**

CREATE TABLE users (

id SERIAL PRIMARY KEY,

fullname VARCHAR(255),

email VARCHAR(255) UNIQUE,

password VARCHAR(255)

);

CREATE TABLE enquiries (

id SERIAL PRIMARY KEY,

fullname VARCHAR(255),

"phoneNumber" VARCHAR(20),

email VARCHAR(255),

dob DATE,

course VARCHAR(100),

source VARCHAR(100),

education VARCHAR(100),

"passedYear" VARCHAR(10),

about TEXT,

mode VARCHAR(50),

batch\_time VARCHAR(50),

language VARCHAR(50),

"demoStatus" VARCHAR(50),

comment TEXT

);

**4. Deployment on Render**

**Web Service:**

1. Create a new Web Service from your GitHub repo.
2. Set build command to:
3. npm install
4. Set start command:
5. node server.js
6. Add environment variable:
7. DATABASE\_URL=<your Render PostgreSQL URL>
8. Deploy and check logs for any issues.

**Static Files:**

* express.static(path.join(\_\_dirname, 'Public')) serves frontend pages.
* Make sure your folder name is Public, not public in Render.

**Sample URL**

Once deployed, you can access your app at:

https://student-portal-<your-app>.onrender.com

**Final Notes**

* Always commit changes to GitHub before triggering deploy
* Check Render -> Logs tab if anything breaks
* Keep your database URL private

**✅ Project Completed and Live!**