Project Name: -Athena

Job: University Management system

To Build App Using Node.js, MongoDB, EJS Template, RESTAPIs and JWT

Background:

Student Portal is a web application which provide web-based student activity management system for all college/school students in order to introduce more reliability and transparency in information exchange activities e.g., exam notification, gate pass details and many more. It has separate admin page to manage the data. as a Node.js Developer, our job is to develop the web application as per the requirement. This application will be implemented as MERN/Full Stack application followed by MVC architecture. This will be a RESTful web service.

Web Application Requirements:

1) ***Frontend***: JavaScript, HTML, CSS and React, Redux and Axios to call API.
2) ***Backend***: Node.js, Express Framework
3) ***Database***: MongoDB ,Mongoosh
4) ***Deployment Tools***: AWS EC2
5) **IDE:** Visual Studio Code/ WebStorm
6) **Code repository:** Git Hub

Goal:

The goal of this project is to develop web application and there will be three user Admin, student, Faculty.

- **Admin**

  At Admin page, admin should login to website using their credentials, which are authenticated through JWT. Admin can add, edit and delete news from the database. Create an API which can be consumed in Customer facing web application.

Finally, both web applications will be deployed in separate docker containers and incoming traffic will be managed by a reverse proxy server Nginx.

The Application is designed to provide user specific functionality. There will be two users of this application:

1) Admin

- At Admin page, admin should enroll the students with a unique registration number.
- Admin Can send email to all the students with their registration numbers to register on the portal.
- Admin can post/publish any *notification, exam timetable, change in faculty, circular on the portal*.
- Admin can block the account of any student.
- Admin can an approve or reject the leave application which is applied by student
- Admin can change or reset user password.

2) Student

- Can register themselves using registration number provided by admin Can change or reset their password Can apply for the leave.
- Can update their profile details.
- Can download *exam timetable / exam gate pass*
- *Can see result term wise as well as grade wise*
- *Can Log complaint and see complaint status*
- *Can upload/download assignment*

3) Faculty

- Can register themselves using registration number provided by admin
- Can change or reset their password Can apply for the leave.
- Can update their profile details.
- Can add CA marks
- Can see Student result as well as marks
- Can Upload /download assignment
- Can provide message regarding subject or section

PROCESS FLOW: -

1. Application Deployment

The Code for the client and the server app will be maintained in the git repository. For each small functionality, a new pull request will be opened and once that pull request is merged to the master repository, latest code will be automatically deployed to the cloud. The Automatic deployment from git will be configured.

The URL of the application will be shared, once the development being finished.

2. Application Testing

When the application will be hosted on the cloud with all functionality, a basic testing will be carried out by postman to check each functionality. Postman will be used for this testing, especially the CRUD operations.

TASKS

Following are the tasks, which need to be developed while executing the project:

1. Client and Server end code through git repository.
2. MongoDB free account credentials.
3. Functionality document with screenshot. (This will be optional)

Web Application Pages: -

1) Login

- **a) User ID (Input Type=" Textbox")**
- **b) Password (Input Type=" Password")**
- **c) Student Email**
- **d) Faculty Email**
- **e) Forgot Password**

2) Ums Navigation

- **a) Examination System**
  - **i) Backlog Registration**
  - **ii) Exam attendance**
  - **iii) Exam Registration**
  - **iv) Result**
- **b) Learning Management System**
  - **i) Assignment**
  - **ii) Makeup Adjustment**
  - **iii) view Timetable**
  - **iv) Attendance**

c) **Finance Management System**
   - i) **Fees Status**
   - ii) **Confirmation fess deposits**

d) **Relationship Management system**
   - i) **Log Request**
   - ii) **View Request status**

3) Import Links

a) **Apply Duplicate Id Card**

b) **Apply RFID**

c) **view Academic calendar**

d) **view My Message**

4) My Profile

5) Change / Reset Password

Admin Application:

It should include following sections –

- ♦ REGISTER
- ♦ Add / Modify Data
- ♦ data should be stored and fetched through MongoDB using admin dashboard (managed by Admin).