

**RAJEEV GANDHI MEMORIAL
COLLEGE OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)**



(ESTD -1995)

RGM

“CODEQUEST²⁴”

Hack Fest

**by
CSE (DATA SCIENCE)**

Accredited by NAAC of UGC, New Delhi with ‘A+’ Grade

Accredited by NBA, Approved by AICTE, Affiliated to

J.N.T. University Anantapur,

Nandyal-518501

Team members: CSE (Cyber Security)

1. Pippala Bharath Kumar Reddy
2. Boya Narasimhulu
3. Thalaricheruvu Shahidabhanu
4. Mohammad Arshad

**RGM COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
NANDYAL**

**RGM CODEQUEST 24 - 18th, 19th October 2024
Hackathon Problem Statement**

Statement:

Course Selection and Teacher Assignment System

Description:

Design and Develop and develop a comprehensive Course Selection and Teacher Assignment System for a university. In this system, each student must choose 4 theory courses and 2 lab courses per semester, selecting a teacher for each course from an available pool. The system should feature a user-friendly interface and a robust database to manage course selections, teacher assignments, and student preferences. Students will be able to view detailed teacher profiles, including ratings from the last four years, research projects, patents, and academic background. The platform should also allow students to submit feedback on their experience with each teacher. The goal is to create a seamless and data-driven platform that optimizes course-teacher matching, enhancing the academic experience by providing students with the necessary tools to make informed decisions based on teacher performance and expertise.

Development Tools:

1. Frontend Development:

- **Frameworks/Libraries:**
 - React, Angular, or Vue.js for building the user interface.
- **Styling:**
 - CSS frameworks like Bootstrap or Tailwind CSS for responsive design.

2. Backend Development:

- **Frameworks:**
 - Node.js with Express, Django (Python), or Flask (Python) for handling server-side logic.
- **Database:**
 - PostgreSQL or MongoDB for data storage and management.

3. API Development:

- **RESTful or GraphQL API** to facilitate communication between the frontend and backend.

Database Management

• Database Management Tools:

- pgAdmin (for PostgreSQL) or MongoDB Compass for database administration and query testing.

Development Environment

• Code Editor/IDE:

- Visual Studio Code, IntelliJ IDEA, or PyCharm for writing and managing code.

• Version Control:

- Git for version control, along with platforms like GitHub or GitLab for collaboration and code hosting.

Testing and Quality Assurance

• Testing Frameworks:

- Jest or Mocha for JavaScript testing.
- Pytest for Python-based applications.

• UI Testing:

- Selenium or Cypress for end-to-end testing of the user interface.

Step by step procedure to execute the submitted solution

Step 1: Set Up Your Development Environment

1. Install Prerequisites:

- Ensure you have Node.js and npm installed (for backend).
- Install Python if you're using Django/Flask for the backend.
- Install a code editor (e.g., Visual Studio Code).

2. Version Control:

- Install Git to manage your code repository.
- Create a new repository on GitHub or GitLab for version control.

Step 2: Initialize Your Project

1. Create the Project Structure:

- Create folders for the frontend and backend.

```
mkdir course-selection-system
cd course-selection-system
mkdir frontend backend
```

2. Initialize Backend:

- Navigate to the backend folder and initialize a new Node.js project:

```
cd backend
npm init -y
```

3. Install Backend Dependencies:

- Install necessary packages (Express, body-parser, etc.):

```
npm install express mongoose cors dotenv
```

4. Initialize Frontend:

- Navigate to the frontend folder and create a React app:

```
cd ../frontend
npx create-react-app .
```

Step 3: Develop the Application

1. Backend Development:

- Create your server file (e.g., server.js).
- Set up Express and MongoDB connection.

- Create routes for courses, teachers, and feedback.
- Implement authentication if necessary.

2. **Frontend Development:**

- Design the user interface using components.
- Use React Router for navigation between pages (course selection, teacher profiles, feedback).
- Fetch data from the backend using Axios or Fetch API.

Step 4: Testing the Application Locally

1. **Run the Backend Server:**

- In the backend directory, start the server:
`node server.js`

2. **Run the Frontend:**

- In the frontend directory, start the React app:
`npm start`

3. **Testing:**

- Test the functionality of course selection, teacher assignment, and feedback submission in your browser.

Step 5: Prepare for Deployment

1. **Build the Frontend:**

- Create a production build of the React app:
`cd frontend`
`npm run build`

2. **Configure Environment Variables:**

- Create a .env file in the backend folder for database connections and other configurations.

Step 6: Deploy the Application

1. **Choose a Hosting Service:**

- For example, use Heroku or AWS.

2. **Deploy the Backend:**

- **Heroku:**
 - Install the Heroku CLI and log in:
`heroku login`
 - Create a new app:
`heroku create your-app-name`

- Push your backend code:

```
git add
```

```
git commit -m "Deploying backend"
```

```
git push heroku main
```

- **AWS:**

- Set up an EC2 instance and SSH into it.
- Clone your repository and install dependencies as mentioned earlier.

3. **Deploy the Frontend:**

- **Heroku:** You can also deploy the frontend in the same app or use a separate one:

```
heroku open
```

- **AWS:** Serve static files using S3 or set up Nginx to serve your frontend from the same EC2 instance.

Step 7: Configure Domain and SSL (Optional)

1. **Domain Setup:**

- Point your domain to the server's IP address or Heroku's domain.

2. **SSL Configuration:**

- Use Let's Encrypt or your hosting provider's SSL options to secure your app.

Step 8: Monitor and Maintain

1. **Logging and Monitoring:**

- Set up logging in your application for error tracking.
- Use monitoring tools (like New Relic) to track performance.

2. **User Feedback:**

- Collect user feedback and iterate on features based on user experience.