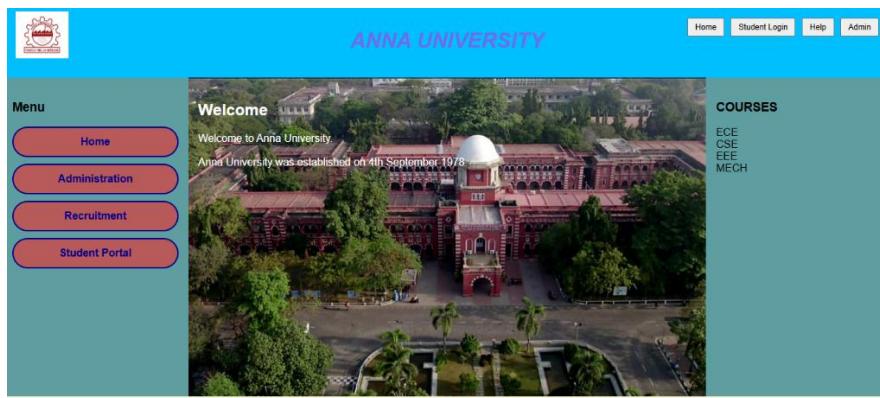


1. Title : Anna University Website Design

Subtitle: Simple HTML & CSS Project

Your Name / Class / Subject



Introduction

- This is a static website created using HTML and CSS.
- The website is designed for Anna University.
- It contains header, navigation menu, content section, and course list.
- The purpose is to create a university homepage layout.

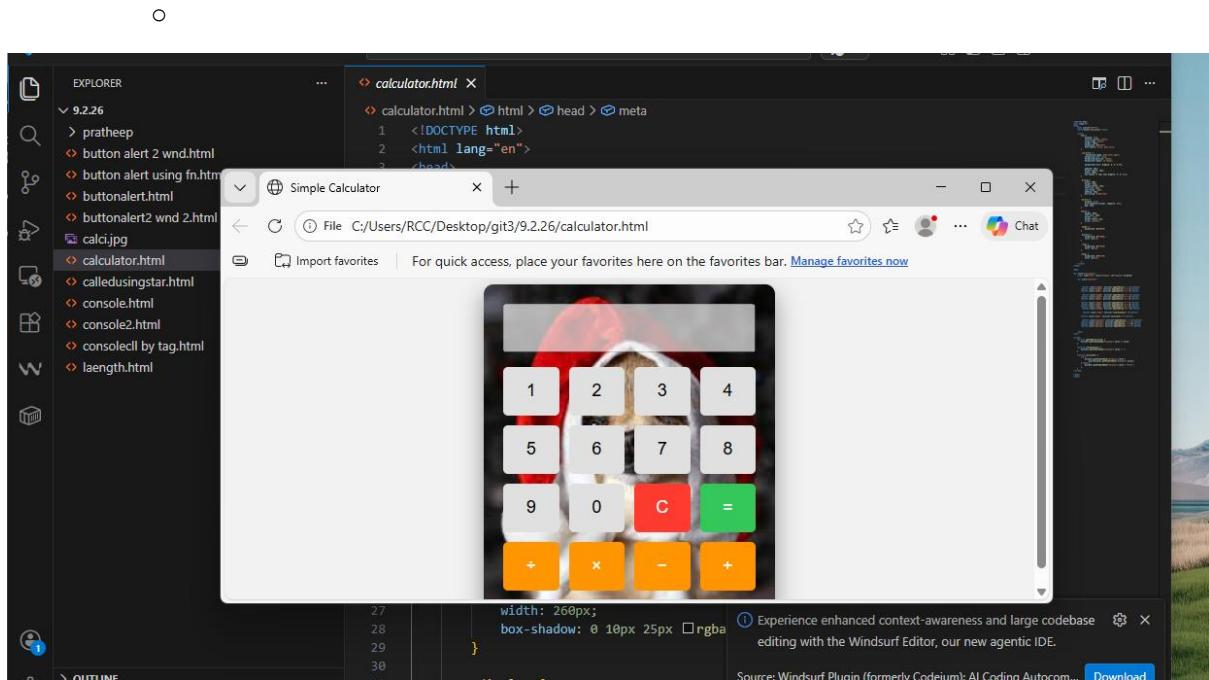
Technologies Used

- HTML – Structure of the webpage
- CSS – Styling and design
- Basic layout using div and flex concepts
- Images for background and logo

Conclusion

- Successfully created a university homepage.
- Learned HTML structure and CSS styling.
- Can improve by adding:
 - Responsive design
 - Login functionality
 - Database connection

2. Title :Simple Calculator using HTML, CSS & JavaScript



Project Overview

- This is a basic working calculator.
- It performs arithmetic operations:
 - Addition (+)
 - Subtraction (-)
 - Multiplication (×)
 - Division (÷)
- Designed using HTML, CSS, and JavaScript.
- Added a background image to improve visual appearance.

Technologies Used

- **HTML** → Structure of calculator
- **CSS** → Styling and layout design
- **JavaScript** → Calculator logic and functionality

Title : Group Chat Application Developed Using Angular



Project Overview

- This is a simple group chat application.
- Built using Angular framework.
- Multiple users can send messages.
- Messages are displayed in a common group message area.
- It simulates a real-time chat interface.

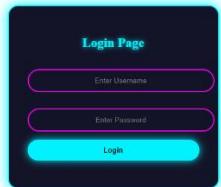


Technologies Used

- Angular → Frontend framework
- TypeScript → Logic implementation
- HTML → Structure
- CSS → Styling and layout

3. Project Title :Simple Login Page with Username & Password Validation

Developed using HTML, CSS & JavaScript



Project Overview

- This project is a basic login authentication page.
- User enters:
 - Username
 - Password
- System checks whether credentials are correct.
- If correct → Login Successful message displayed.
- If wrong → Error message shown.

Technologies Used

- HTML → Structure of the form
- CSS → Styling and design
- JavaScript → Validation logic

What I Learned

- Form handling
- Basic authentication logic
- CSS glow effects
- User interaction handling

Project Title :Responsive Navigation Bar using Flexbox



Project Overview

- This project is a simple navigation bar.
- Created using HTML and CSS.
- Used Flexbox for alignment.
- Contains:
 - Website Logo/Title (MySite)
 - Navigation Links (Home, About, Service, Contact)

Technologies Used

- HTML → Structure
- CSS → Styling
- Flexbox → Layout alignment

Slide 5 – Flexbox Concept Used

Main CSS property:

```
.navbar {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
}
```

Explanation:

- display: flex; → Activates flexbox layout
- justify-content: space-between; → Logo left, menu right
- align-items: center; → Vertical center alignment

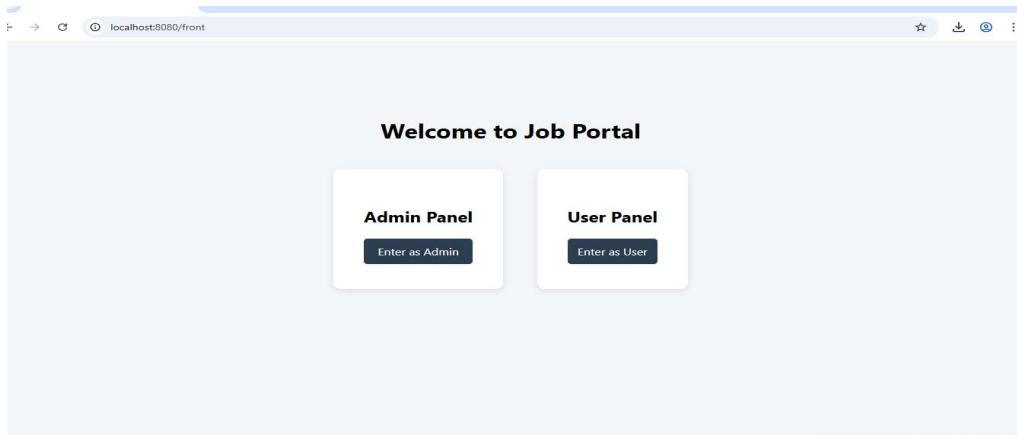
4. TITLE : SIMPLE JOB PORTAL

I built a simple job portal application using Spring Boot, Thymeleaf (HTML templates), and MySQL as the database.

The application has two types of users: Admin and User.

Admin can add, edit, delete jobs, while User can view jobs and apply.

The application is fully functional with a clean frontend design using HTML and CSS.



Key Components

Entities

Job: Represents a job listing with fields: id, title, company, location, description, and postedBy.

User: Represents a user with fields: id, username, password, role, and list of jobs posted.

Repositories

JobRepository — Handles database operations for jobs.

UserRepository — Handles database operations for users.

Controllers

HomeController — Serves HTML pages (Thymeleaf) like home, add-job, edit-job.

JobRestController — Provides REST API endpoints to add or fetch jobs.

UserRestController — Handles user registration and fetching users.

The screenshot shows a web application titled "Job Portal". At the top right, there are "Admin" and "Switch Role" buttons. Below the header, there is a green button labeled "+ Add Job". The main content area displays three job listings in separate boxes:

- Java Developer**
TCS | Chennai
Spring Boot Developer Required
Edit Delete
- Developer**
TCS | Chennai
Spring Boot Developer Required
Edit Delete
- Developer12**
TCS12 | Chennai12
Spring Boot Developer Required
Edit Delete

The Admin can:

Add new jobs

Edit existing jobs

Delete jobs

Frontend Pages

front.html – Allows selection between Admin and User.

index.html – Shows list of jobs. Admin sees Add/Edit/Delete buttons, User sees Apply button.

add-job.html – Form to add a new job.

edit-job.html – Form to edit an existing job.

The screenshot shows a browser window with the URL "localhost:8080/add/job". The title bar says "Add Job" and there is a "Back" button. The main content is a form with four input fields: "Title", "Company", "Location", and "Description". Below the form is a green "Save Job" button. At the bottom center of the page is a small "Word" button.

Application Flow

User visits the application → front page shows Admin/User options.

Admin login (for simplicity, role is selected manually) → can add new jobs.

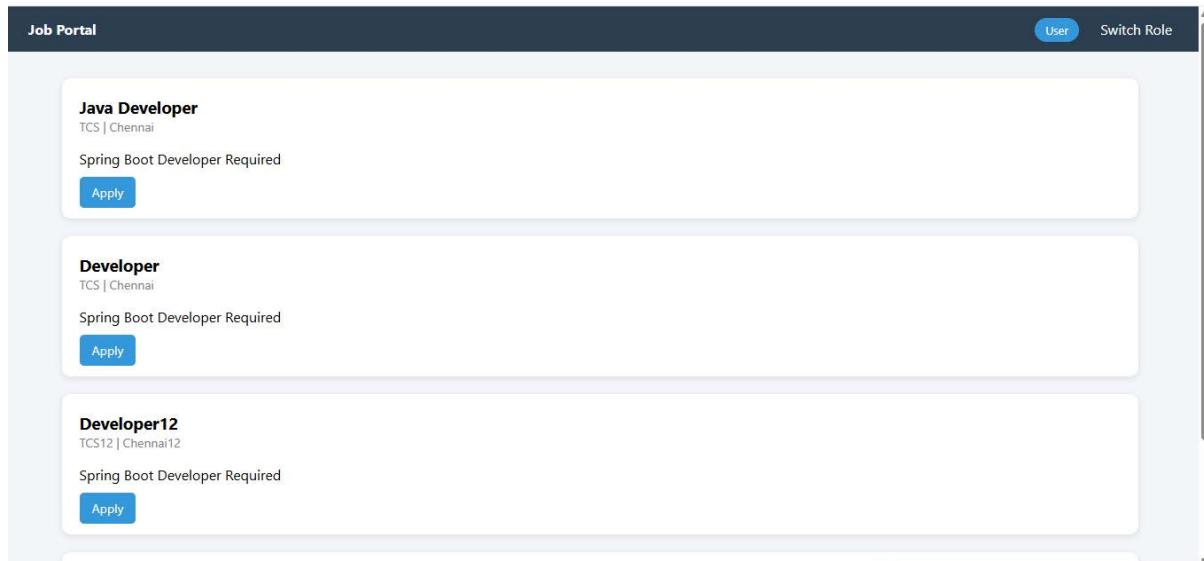
Jobs are displayed on the homepage:

Admin view: Add/Edit/Delete buttons visible.

User view: Only Apply button visible.

Apply button triggers a simple JavaScript alert:
“You applied for this job”.

All data is stored in MySQL via JPA.



The User can:

View all available jobs

Click the Apply button, which currently shows a confirmation alert

In future improvements, I plan to add Spring Security for proper authentication and store applied jobs in the database.

Overall, this project demonstrates a complete full-stack workflow from frontend to backend to database.