FullStack Developer Problem Statements

CMS APP

Brief:

- As a user, I need a CMS application to publish and share my content on a daily basis.

Problem statement:

CMS should have the following capability.

- Develop backend APIS and integrate it into the frontend application.
- Develop an admin view to manage users / posts / comments
- Develop a front-end app with the following features:

  1. User can view all blog posts.
  2. Users can read the full blog post.
  3. Users can provide feedback on blog posts.

Module 1#

- Create a database with the required table structure (use any database, e.g. MySQL, Mongo)
- Use any backend framework to develop the REST API (e.g. Django, Flask, Express Js).
- Secure the REST API with JWT or OAUTH tokens

Module 2#

- Use any front-end framework to develop an application front-end (e.g. React Js, Angular).
- Create a login page to authenticate the administrator and users
- Develop the REST API for users
- Create list / create / update / delete view for users

Module 3#

- Create an HTML template for application front-end pages (blog post list page, blog post detail page).

Module 4#

- Create list / create / update / delete views for blog entries at the end of the administrator

- Provide an option to upload images and files as attachments for blog posts
- Media files should be stored in the cloud storage (e.g. S3, Azure-Blob).

Module 5#

- Create a list view to list all blog posts with images retrieved from database or API
- Create a detailed view page to read the post

Module 6#

- Provide the users with the ability to enter the comments on each blog post
- Create a list view for comments with approval / block options at the end of the admin

What are we looking for:

- Use of clean code, SOLID principles and coding standards, code documentation
- Code structure and folder organization.
- Knowledge of Docker and Git repository
- Use of cloud services (AWS, Azure)
- Knowledge of object-oriented programming and implementation of design patterns.
- Understanding of the REST API, JavaScript or Python, HTML, JSON and CSS.

Brownie Points:

- Creativity in terms of look and feel (UI & smooth transitions) of the page.