**E-Learning Platform**

**Project Goal**

**The project aims to develop a computerized E-Learning platform to manage online courses. It is divided into two main modules Admin (Instructor) and Student.**

**Software Processes using the MVP approach**

**Here is an explanation of the questions in the image and how it applies to your project.**

**State your team around the project (Roles and Responsibilities).**

**Ensures the project meets deadlines and goals.**

**In your previous project this was fulfilled by yourself.**

**Frontend**

**Designs and implements the user interface (UI) and user experience (UX).**

**Creates interactive web pages and ensures responsiveness.**

**Backend**

**Designs and implements the server-side logic (database, APIs).**

**Manages the models, views, and business logic.**

**In your previous project this was fulfilled by yourself.**

**User Authentication**

**Users (both Students and Instructors) should be able to register and log in securely.**

**Course Management**

**Instructors able to create, edit, and delete courses and lessons (Admin Module).**

**Users able to view a list of available courses.**

**Users able to see a detail of a course.**

**Users able to see the content of the lessons.**

**Student Enrollment**

**Students able to enroll in courses (Student Module).**

**Students should be able to see a list of enrolled courses.**

**Core Features**

**User Authentication (Signup and Signin).**

**Course Listing A page to view all available courses.**

**Course Detail A page showing details about a specific course.**

**Lesson Detail A page to see the lesson content.**

**Enrollment Ability to enroll in courses.**

**My Courses Displays the courses the user is enrolled in.**

**Course Creation Ability for a staff user to create courses.**

**Example**

**Users should be able to create a new course using a course form, and they should be able to create a first lesson when creating a new course.**

**Users should be able to see a list of courses.**

**State a straightforward design that makes your design remarkable.**

**Clear Layout**

**Use a simple and intuitive navigation bar.**

**Organize content using the main container and section.**

**Ensure a clear distinction between navigation elements and content with different background colors.**

**Consistent Visuals**

**Use the same styling for all pages and form elements by using a CSS file, and using inline styles to customize specific elements.**

**Use consistent fonts, colors, and typography.**

**Use rounded borders for elements to create a cleaner look and feel.**

**Responsiveness**

**website is responsive for different screen sizes.**

**Color Palette**

**Using a light yellowish color as background color with a transparent white background for the main containers.**

**Apply one design process of the Malu’s story.**

**Malu's Design Thinking Framework**

**This is a framework used to focus the design on the user and not in the business logic. The idea is to first understand the user, then generate ideas, then prototyping and finally testing the product with the users. In this case the idea is to focus on making a clear, understandable and useful interface for the user.**

**Define backend technologies.**

**Language Python, due to the use of Django.**

**Framework Django (for handling the models, views, APIs, authentication).**

**Database SQLite (for simple data storage and development purposes).**

**Example from the project**

**models.py (Django models),views.py (Django views).**

**Define frontend technologies.**

**HTML For structure and content.**

**CSS For styling and layout.**

**Use of Bootstrap for handling most of the styling, and a css custom file to override some Bootstrap elements and for custom styling.**

**JavaScript For basic interactivity and to make dropdown menus work correctly.**

**Example from the project**

**The templates used for the webpages, CSS file (styles.css), Bootstrap.**

**State how to contract between frontend and backend.**

**Templates Django templates are used to dynamically render HTML pages. The front end will be rendered using HTML and will call the backend to retrieve data using Django template tags.**

**Forms Django's forms (using a class structure) are used to handle user input on forms and sent to the backend.**

**URLs and Views URL patterns defined in urls.py are mapped to Django view functions in views.py. The frontend is linked to the backend through url routing.**

**State different types of testing you will do.**

**Unit Testing**

**Test individual functions, models and views in isolation, in the backend.**

**Integration Testing**

**Verify the interaction between the different modules.**