Course Name: Software Development Project
Course No: CSE 3106

Submitted to :
Dr. Amit Kumar Mondal
Associate Professor
Computer Science & Engineering Discipline,
Khulna University, Khulna.

Submitted by:
Name: Md Anamul Islam
Student ID: 210219
Name: Atikullah Atik
Student ID: 210220

Proposal: Layered Architecture for HTML Learner Desktop Application

Introduction:

In this proposal, we outline the design and architecture for an HTML Learner desktop application. The application aims to provide users with an interactive platform to learn HTML concepts, syntax, and best practices through a user-friendly interface.

Reasons for Selecting Layered Architecture:

Layered architecture has been chosen for the following reasons:

1. Modularity and Separation of Concerns: Layered architecture promotes modularity by dividing the application into distinct layers, each responsible for specific functionalities. This separation of concerns facilitates easier maintenance, testing, and understanding of the application's components.

2. Scalability: The layered architecture allows for scalability by enabling individual layers to be scaled independently. This flexibility is crucial as the application evolves and grows in complexity over time.

3. Reusability: By encapsulating functionalities within separate layers, components can be reused across different parts of the application or in future projects. This promotes code reusability and reduces redundancy, leading to more efficient development processes.

4. Flexibility and Adaptability: Layered architecture provides flexibility to adapt to changes in requirements or technology. Each layer can be modified or replaced without significantly impacting other parts of the application, making it easier to incorporate new features or integrate with external systems.

Proposed Layered Architecture

1. Presentation Layer:
   View: Responsible for rendering the user interface and capturing user interactions.
   Presenter/Controller: Mediates between the view and the underlying business logic, handling user inputs and updating the view accordingly.

2. Business Logic Layer:
- Service Layer: Implements the application's business logic, including HTML learning modules, quizzes, and progress tracking.

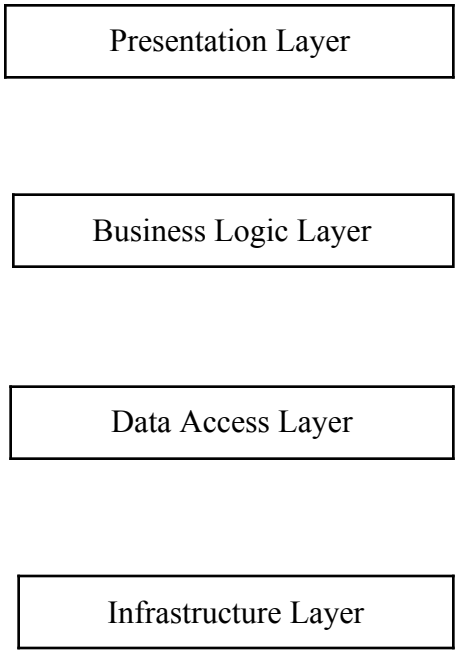
Domain Model: Defines the core entities and their behaviors, encapsulating business rules and data access logic.

3. Data Access Layer:
Repository/DAO: Manages interactions with the underlying data storage, such as a local database or file system.

4. Infrastructure Layer:
- External Services: Interfaces with external services or APIs for additional functionalities, such as fetching HTML resources or accessing online tutorials.
- Utilities: Contains reusable utility functions or helper classes utilized across the application.

Conclusion:

The proposed layered architecture provides a robust foundation for the HTML Learner desktop application, offering modularity, scalability, reusability, and flexibility. By adopting this architecture, we aim to deliver a high-quality learning experience for users while ensuring the maintainability and extensibility of the application.


Presentation Layer
Business Logic Layer
Data Access Layer
Infrastructure Layer