The MVC Pattern Architecture

Lamia Binta Latif

23 September 2024

Introduction

The Model-View-Controller (MVC) framework is an architectural design pattern that separates an application into three main logical components: Model, View, and Controller. Each component is designed to handle specific development aspects of an application. It isolates the business logic and presentation layer from each other. Initially used for desktop graphical user interfaces (GUIs), MVC is now widely employed as an industry-standard web development framework to create scalable and extensible projects. Additionally, MVC is used in designing mobile applications.

Features of MVC

- Clear separation of business logic, UI logic, and input logic.
- Full control over HTML and URLs, making web application architecture easier to design.
- A powerful URL-mapping component, enabling the creation of comprehensible and searchable URLs.
- Supports Test Driven Development (TDD).

Components of MVC

The MVC framework consists of three main components:

- Controller
- Model
- View

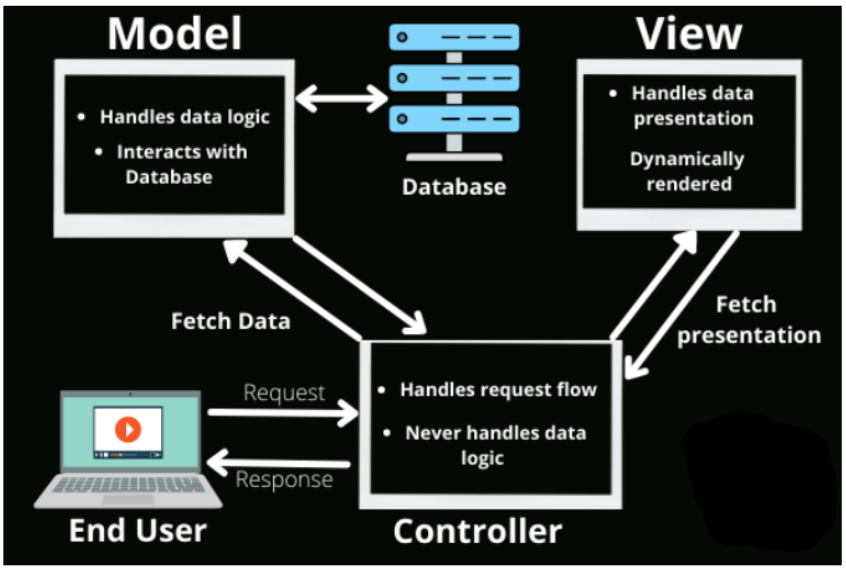

Figure 1: MVC Architecture

Controller

The controller acts as the intermediary between the View and Model, processing all business logic and incoming requests. It manipulates data using the Model component and interacts with the View to render the final output.

Responsibilities:

- Receiving user input and interpreting it.
- Updating the Model based on user actions.
- Selecting and displaying the appropriate View.

Example: In an online store project, *Ki-Kinbo!*, the Controller would handle actions such as searching for products, adding a product to the cart, or checking out.

View

The View component is responsible for all the UI logic in the application. It interacts with the Controller to gather the necessary data from the Model and renders the user interface accordingly.

Responsibilities:

- Rendering data to the user in a specific format.

- Displaying the user interface elements.
- Updating the display when the Model changes.

Example: In a bookstore application, the View would display the list of books, book details, and provide input fields for searching or filtering books.

Model

The Model corresponds to the data-related logic the user works with. It manages CRUD (Create, Read, Update, Delete) operations and interacts with the database to retrieve or store data. The Controller uses the Model to manipulate the data and returns the results to the View.

Responsibilities:

- Managing data (CRUD operations).
- Enforcing business rules.
- Notifying the View and Controller of state changes.

Example: In the *Ki-Kinbo!* project, the Model would handle data related to products, such as product title, description, price, and stock level.

Advantages of MVC

- Easy to maintain and extend.
- The MVC Model component can be tested independently.
- Components of MVC can be developed simultaneously.
- Reduces complexity by dividing an application into three components: Model, View, and Controller.
- Supports Test Driven Development (TDD).
- Works well for large web applications supported by teams of web designers and developers.
- Allows independent testing of components as all classes and objects are independent of each other.
- SEO-friendly architecture.

Disadvantages of MVC

- Increased complexity due to the separation of concerns.
- Not suitable for small applications.
- Inefficient data access in the View.
- Complex framework navigation due to new layers of abstraction.