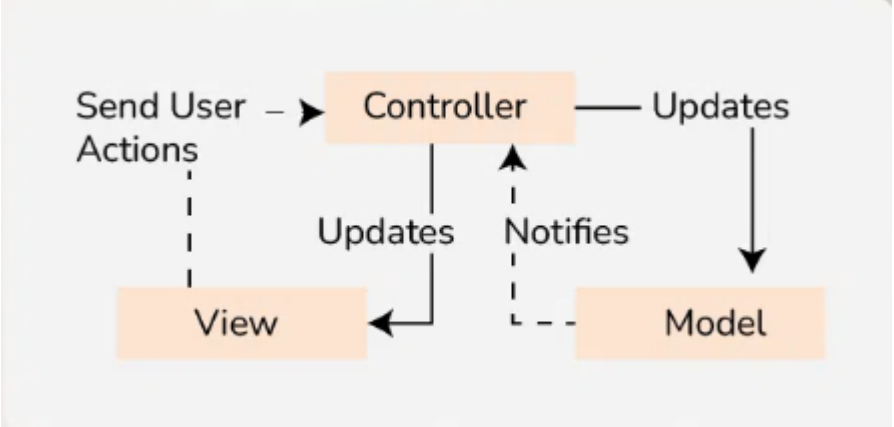
T.Ayush Raj
2141018011

Assignment 1: MVC Design Pattern and Variants

MVC (Model-View-Controller) separates an application into three interconnected components:
- Model: Manages the data and business logic.
- View: Handles the display and user interface.
- Controller: Processes user input and updates the Model and View accordingly.

Basic MVC Flow:



User Variants of MVC:
1. Model-View-Presenter (MVP)
- Presenter acts as intermediary between Model and View.
- View is passive and sends user inputs to the Presenter.
Flow: User --> View --> Presenter --> Model; Presenter --> View

2. Model-View-ViewModel (MVVM)
- Common in frameworks like Angular and WPF. - ViewModel binds the View and Model via two-way data binding.
Flow: User --> View <--> ViewModel <--> Model

When to Use:
- MVC: Simple web applications.
- MVP: Complex user interactions (e.g., desktop apps).
- MVVM: Real-time UI updates (e.g., mobile apps).

T.Ayush Raj
2141018011

Assignment 2: Microservices and Event-Driven Architecture in E-commerce

An e-commerce platform handles products, users, and orders.

Architectures:

- Microservices:
    - Services: Inventory, Orders, Payments, User Profiles, Notifications.
    - Communicate via lightweight APIs.

- Event-Driven Architecture:
    - Order Service emits "Order Placed".
    - Inventory Service listens and updates stock.
    - Notification Service sends email.

SOLID Principles:

- Single Responsibility: Each service has a clear purpose.
- Open/Closed: Extend services without altering existing code.
- Liskov Substitution: Swappable payment implementations.
- Interface Segregation: Fine-grained service interfaces.
- Dependency Inversion: Use abstractions over concrete dependencies.

DRY (Don't Repeat Yourself):

- Centralized authentication service.
- Shared libraries for logging/monitoring.
- Shared API contracts (OpenAPI/Swagger) to avoid duplicate endpoint definitions.

KISS(Keep It Simple, Stupid):

- Simple REST APIs.
- Business logic handled in backend services.
- Minimal shared state—communication happens through well-defined events.

T.Ayush Raj
2141018011

Assignment 3: Trends and Cloud Services Overview

Benefits of Serverless Architecture

Serverless computing eliminates the need to manage server infrastructure, allowing developers to focus purely on writing code. It offers automatic scaling, reduces operational costs, and supports event-driven workloads, making it ideal for unpredictable or bursty traffic scenarios.

Concept of Progressive Web Apps (PWAs)

PWAs are web applications that provide a native-app-like experience on browsers. They support offline access, push notifications, and fast load times, making them a powerful choice for companies aiming to reach users across different devices without developing separate apps.

Role of AI and Machine Learning in Software Architecture

AI and ML influence architecture by enabling intelligent features like recommendation systems, fraud detection, and predictive analytics. Architectures now often include specialized components like ML model servers, data lakes, and pipelines to manage continuous training and deployment.

Cloud Computing Service Models

- **SaaS (Software as a Service)**: End-user applications hosted in the cloud (e.g., Google Docs, Salesforce).
- **PaaS (Platform as a Service)**: Provides platforms for building, testing, and deploying applications without managing underlying infrastructure (e.g., Heroku, AWS Elastic Beanstalk).
- **IaaS (Infrastructure as a Service)**: Offers virtualized computing resources over the internet (e.g., AWS EC2, Microsoft Azure).