1. What is the difference between a) MVC and 3-Tier Architecture? [2080 Baisakh QN. 5 a)]

Ans. The difference between MVC and 3-tier architecture is given below:

|  |  |  |
| --- | --- | --- |
| Aspect | MVC (Model View Architecture) | 3-tier Architecture |
| Purpose | Design pattern for organizing code in web applications. | Architectural pattern for organizing application layers. |
| Components / Tiers | Model, View, Controller | Presentation Tier, Application (Logic) Tier, Data Tier |
| Model / Data tier | Represents and manages data, logic, and rules. | Manages data storage and retrieval from the database. |
| View / Presentation Layer | Manages the user interface and presentation logic. | Displays data to the user and sends user requests to the logic tier. |
| Controller / Application Tier | Handles user input, manipulates the model, and updates the view. | Processes business logic and communicates between the presentation and data tiers. |
| Separation of concerns | Separates application into three interconnected components to improve modularity. | Separates application into distinct layers to enhance maintainability and scalability. |
| Use Cases | Primarily used in web applications and frameworks. | Used in enterprise applications and complex systems. |
| Examples | Web frameworks like Ruby on Rails, Django, ASP.NET MVC. | Applications built with technologies like Java EE, .NET, and PHP with layered architecture. |
| Interaction Flow | User interacts with View, Controller processes, and updates Model, then View updates. | User interacts with Presentation Tier, which calls Application Tier for processing, and interacts with Data Tier for data access. |
| Flexibility | High flexibility within the application components. | High flexibility in deploying and scaling each layer independently. |

1. What is Content Management System? Explain information system at different levels of management. [2080 Baisakh QN. 7]

Ans. A **Content Management System (CMS)** is software that allows users to create, manage, and modify digital content on a website or other digital platforms without needing specialized technical knowledge. It typically provides an interface to manage and publish content, handle various media types, and organize content in a structured manner. Examples: WordPress, Drupal, Joomla! etc.

Features:

* **User-Friendly Interface**: Allows users to easily create and manage content through a web-based interface.
* **Templates and Themes**: Provides customizable templates for designing the appearance of the website.
* **Content Organization**: Organizes content into categories and tags for better management and retrieval.
* **Access Control**: Manages user permissions to control who can create, edit, or publish content.
* **Media Management**: Handles uploading and managing various media types such as images, videos, and documents.
* **Workflow Management**: Supports content approval processes and version control.

Web information system, or web-based information system, is an information system that uses Internet web technologies to deliver information and services, to users or other information systems/applications. It is a software system whose primary aim is to publish and maintain data by using hypertext-based principles.

Web-based information displays many benefits of multimedia technology. Using today's fast broadband connection, it's possible to stream sophisticated content to a computer anywhere in the world. This is an advantage for many people as the information can be received and read wherever and whenever it is convenient for them, which can be a crucial factor for a busy executive. A significant amount of interactive multimedia content is now delivered via the Internet.

A web information system usually consists of one or more web applications, and specific functionality-oriented components, together with information components and other non-web components. A web browser is typically used as the front-end whereas databases as back-end.

Key features of web-based Information System**:**

Web-based information systems have evolved significantly over recent years with its improvement. Web-based applications have several advantages over traditional software-based applications. Some of the core features of web-based applications are given below:

1. Cross-platform compatibility:

Most web-based applications are compatible in different platforms than traditional installed software. The minimum requirement would be a web browser (Internet Explorer, Firefox, Netscape etc.). You can use different OS such as Windows, Linux or Mac to run the web applications.

1. More Manageable:

WBIS only needs to be installed on the server placing minimal requirements on the end user workstation, which makes the system easier to maintain and update as usually it can all be done on the server.

1. Multiple concurrent users:

Web-based applications can indeed be used by multiple users at the same time. It’s not necessary to share a screen or send a screenshot when multiple users see and even edit the same document at the same time. Web conferencing and online collaboration companies regulate some key transformations and users only explore what they really need to work effectively and co-edit documents together.

1. Reduced cost:

Web-based applications can reduce costs due to support and maintenance, lower requirements on the end user system, and simplified architecture. It doesn’t require any distribution or marketing infrastructure.

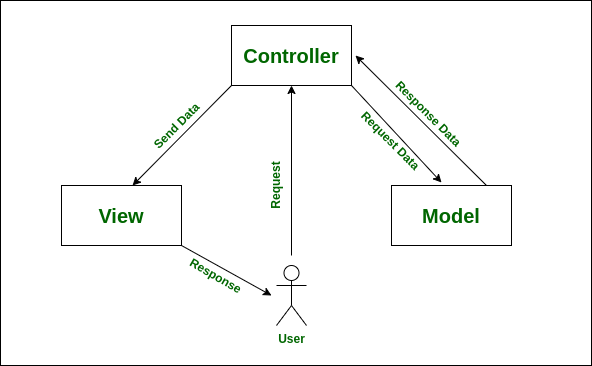
1. Secure live data:

These applications can decrease the risk of losing data due to an unexpected disk crash or computer virus. Companies of web-based applications provide extensive data backup services either as an integral part or a basic service or sometimes as a paid service.

1. Distinguish between Content Management System (CMS) and Web Application Framework. Describe the working of the MVC (Model View Controller) architecture. [2079 Bhadra QN. 5]

Ans. The difference between CMS and Web Application Framework is given below:

|  |  |  |
| --- | --- | --- |
| Aspect | Control Management System (CMS) | Web Application Framework |
| Purpose | Designed for creating, managing, and modifying digital content easily. | Provides a structured foundation for building web applications. |
| Primary Use | Content creation and management for websites. | Development of custom web applications and dynamic websites. |
| Examples | WordPress, Joomla, Drupal | Laravel, Django, Ruby on Rails |
| User Interface | Typically includes a user-friendly interface for non-technical users to manage content. | Focuses on developers, usually lacking built-in content management interfaces. |
| Customization | Often includes themes and plugins for extending functionality and changing appearance. | Customizable by writing code; provides tools and libraries to build features. |
| Flexibility | Limited to content management and site configuration. | Highly flexible, suitable for various types of web applications. |
| Development Skills | Minimal technical skills required for basic content management. | Requires programming knowledge for application development. |
| Functionality | Pre-built features for content management, user roles, and SEO. | Core functionalities are basic; additional features need to be coded. |
| Examples of Functionality | Content editing, media management, SEO tools, user management. | Routing, database access, templating, form handling. |
| Integration | Often integrates with other services via plugins or modules. | Integration done via coding and libraries, often requires additional setup. |
| Security | Security features are typically included but may need updates and plugins. | Security features need to be implemented by developers and maintained. |
| Development Speed | Faster setup for content-driven websites with minimal coding. | Requires more development time to build and customize applications from scratch. |

Fig: MVC architecture

1. **Model**    
   This level is considered the lowest level when compared with the View and Controller. It primarily represents the **data to the user** and defines the storage of all the application’s data objects. It also contains the application logic.
2. **Views**   
   This level is majorly associated with the **User Interface(UI)** and it is used to provide the visual representation of the MVC model. In simpler terms, this level deals with displaying the actual output to the user. It also handles the communication between the user (inputs, requests, etc.) and the controller.
3. **Controller**  
   This level takes care of the **request handler**. The controller completes the cycle of taking the user output, and converting it into desired messages.
4. What is Content Management System? Explain the Web application framework. [2078 Bhadra QN. 5]

Ans. A **Content Management System (CMS)** is software that enables users to create, manage, and modify digital content on a website or other digital platforms without requiring specialized technical skills. It provides a user-friendly interface for handling various aspects of content management and website administration.

A **Web Application Framework** is a software framework designed to aid the development of web applications, including web services, web resources, and web APIs. It provides a structured environment and set of tools to streamline the process of building and maintaining web applications. Examples: Laravel (PHP), Django (Python), Express.js (Node.js) etc.

Features:

* **Framework Structure**:
  + **Architecture**: Provides a standard way to organize code and separate concerns, often following design patterns like MVC (Model-View-Controller) or MVVM (Model-View-ViewModel).
  + **Conventions**: Follows conventions and best practices for structuring application code and directories.
* **Routing**: Manages how HTTP requests are mapped to application logic. Defines URL patterns and maps them to specific controllers or handlers.
* **Template Engine**: Facilitates dynamic content generation by allowing developers to use templates that can be rendered with data. This separates the presentation layer from the business logic.
* **Database Integration**: Provides tools and libraries for interacting with databases, including Object-Relational Mapping (ORM) for easier database management and querying.
* **Security Features**: Includes built-in security features such as protection against common vulnerabilities (e.g., SQL injection, Cross-Site Scripting), authentication, and authorization.

1. Explain the Model View Controller (MVC) pattern. [2076 Chaitra QN. 9]

Ans. *(Repeated Question)*

1. Define web application framework. List some advantages of the web application framework. Explain the concept of MVC (Model View Controller). [2074 Chaitra QN. 9]

Ans. *(Repeated)*