

LAB ASSIGNMENT

Qus.1 An Introduction of software product life cycle(SDLC models name Diagram, Advantage, Disadvantage).

Ans:1.1 Software product life cycle:- The software product life cycle (SDLC) is a structured process that guides the creation of software from initial planning to eventual maintenance and retirement, encompassing stages like planning, design, development, testing, deployment, and maintenance.

➤ **Phases of SDLC:-**

- Planning
- Requirement Analysis
- Design
- Implementation(Coding)
- Testing
- Deployment
- Maintenance

➤ **Advantages of SDLC:-**

- Clear project objectives and deliverable.
- Improves project planning and tracking.
- Ensures high-quality software with systematic testing.
- Reduces project risk and cost over time

➤ **Disadvantages of SDLC:-**

- Requires accurate requirement gathering early on.
- Time-consuming documentation.
- Model are too risky for complex projects.

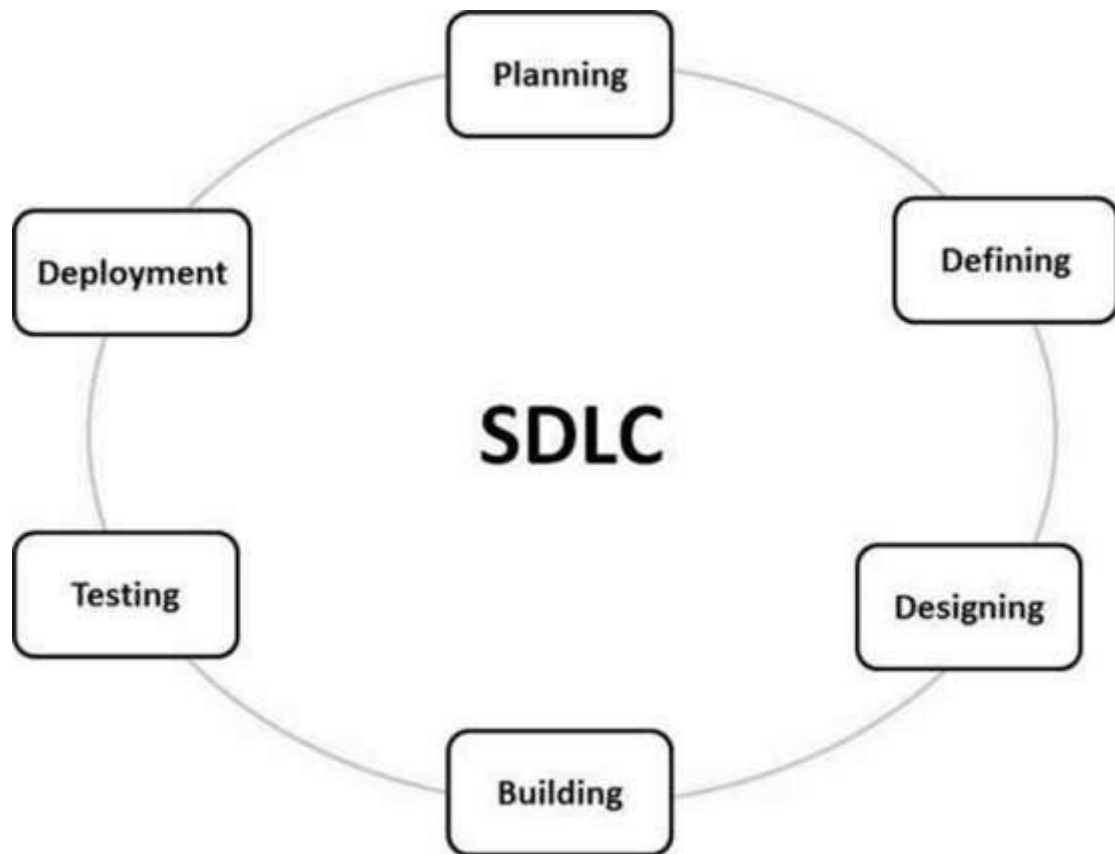


Diagram:-SDLC

1.2. Waterfall Model:-.

➤ Advantage of waterfall Model:-

- Simple and easy to understand and use.
- Phases are processed and completed one at a time.
- Clearly defined stages.
- Easy to arrange tasks.
- Process and results are well document.

➤ Disadvantages of waterfall Model:-

- Once a phase is completed, it's difficult to go back and make changes.
- Testing is done only after the development is complete, which means bugs and issues are detected late, making them harder and more expensive to fix.
- High Risk.

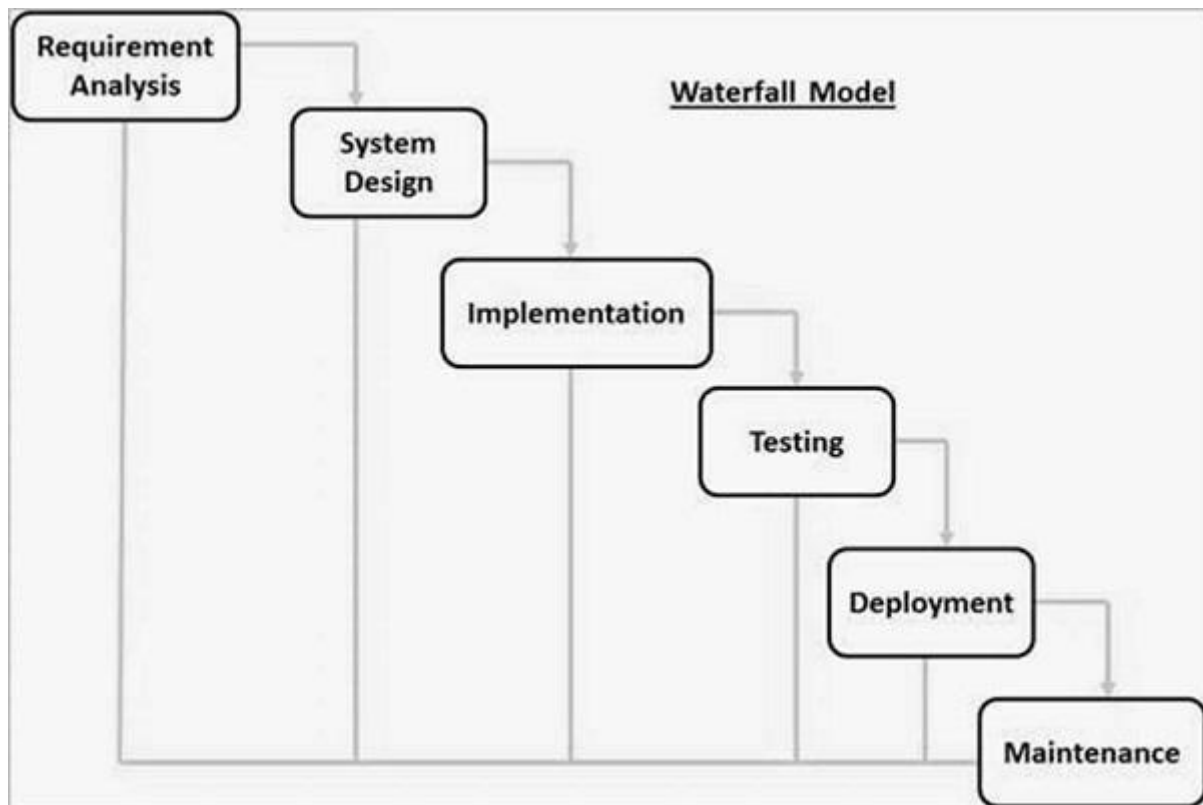


Diagram:-Waterfall Model

1.3. V-Model (Verification and Validation Model):-

➤ **Advantage of V-Model:-**

- This is a highly-disciplined model and Phases are completed one at a time.
- Simple and easy to understand and use.

➤ **Disadvantage of V-Model:-**

- High risk and uncertainty.
- Not a good model for complex and object-oriented projects.

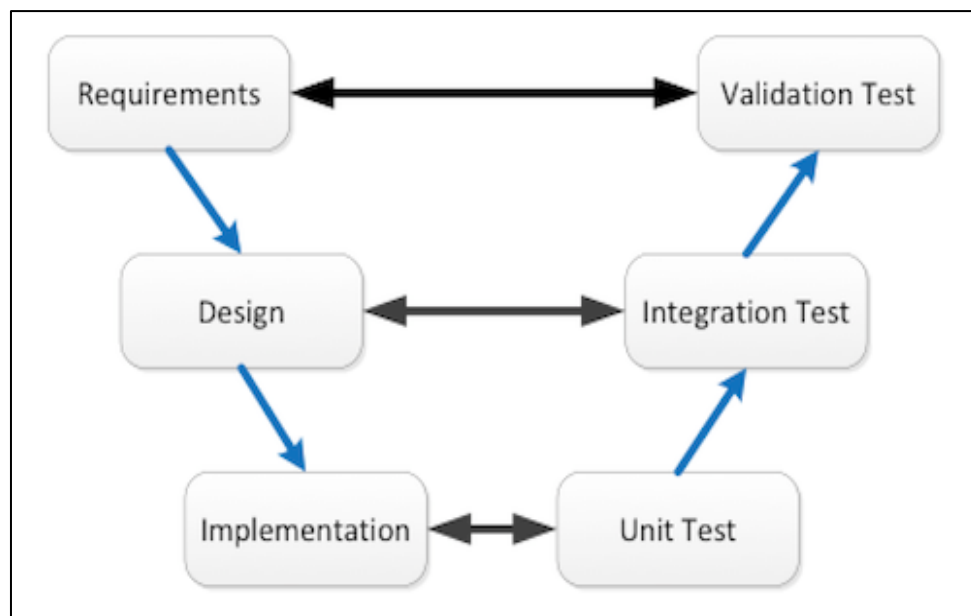


Diagram:-V-Model

1.4. Incremental Model:-

➤ **Advantage of increment Model:-**

- Errors are easy to be recognized.
- Easier to test and debug.
- More flexible.
- Simple to manage risk.

➤ **Disadvantage of increment Model:-**

- Need for good planning.
- Total Cost is high.

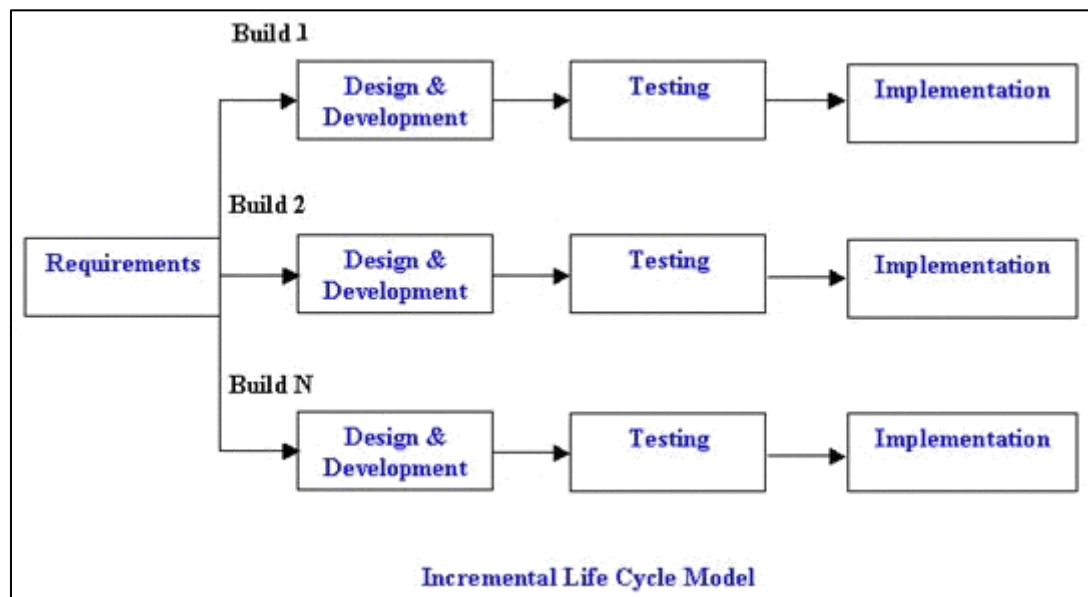


Diagram:-Increment Model

1.5. RAD Model (Rapid Application Development):-

➤ **Advantages:**

- Fast development
- Flexible changes
- User feedback
- Less planning

➤ **Disadvantages:**

- Not for big projects
- Needs active users
- Skilled team needed

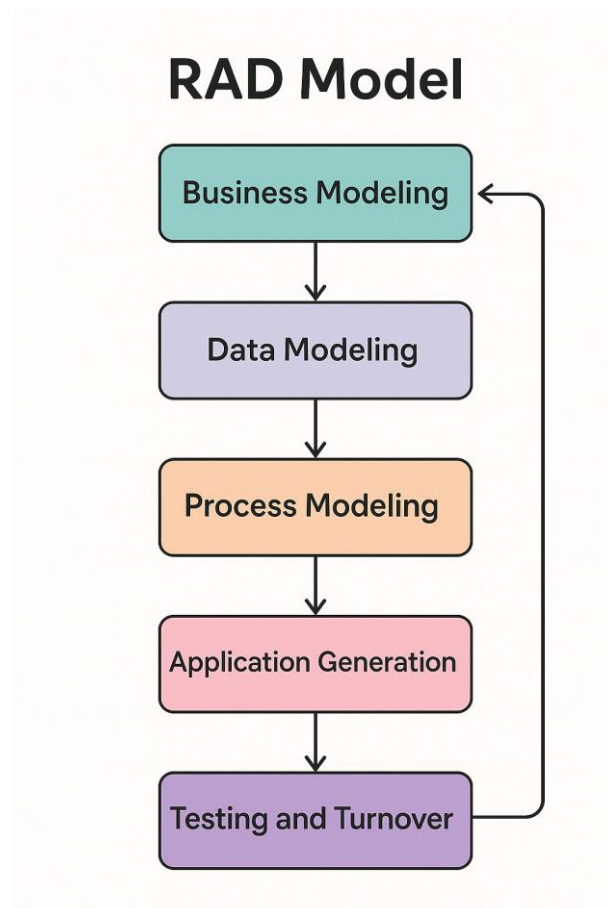


Diagram:-RAD Model

1.6. Agile Model:- Iterative and highly flexible. The Agile model is a project management and software development approach that emphasizes iterative development, flexibility, and collaboration, breaking down projects into smaller, manageable units of work and prioritizing continuous delivery and feedback.

➤ **Phases of Agile Model:-**

- Requirements gathering
- Design the requirements
- Construction/ iteration
- Testing/ Quality assurance
- Deployment
- Feedback

➤ **Features:-**

- Customer feedback.
- Fast delivery.
- Continuous.
- Improvement.

➤ **Advantages of Agile Model:-**

- **Customer Collaboration:-**Frequent feedback ensures that the final product meets user needs and expectations.
- **Faster Delivery:-**Working software is delivered in short sprints, giving customers quick access to usable features.
- **Improved Quality:-**Continuous testing and integration help detect and fix issues early.
- **Better Risk Management:-**Issues are identified and resolved early due to constant feedback and reviews.

➤ **Disadvantages of Agile Model:-**

- **Requires Experienced Team:-**Agile practices rely heavily on skilled and self-disciplined team members.
- **Difficult for Large Teams:-**Coordinating large or distributed teams can be challenging without proper structure.
- **Not Ideal for All Projects:-**Projects with fixed scope, cost, and timeline may not benefit from Agile's flexibility.

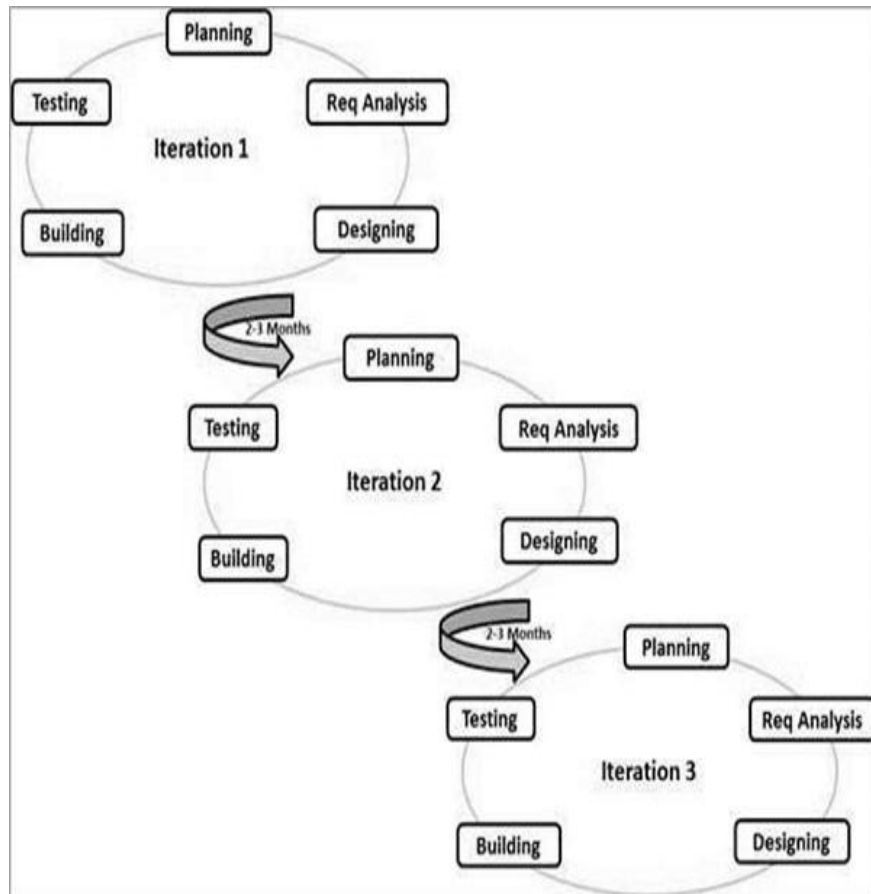


Diagram:-Agile Model

Qus .2 Case study of various software development trends. Theory, Advantages ,Disadvantages, Application and Diagram.

- 1. AI**
- 2. IOT**
- 3. Blockchain**
- 4. GenAI**

Ans :-2.1 Artificial Intelligence:-Artificial Intelligence is the simulation of human intelligence processes by machines. It includes learning (acquiring data), reasoning (using rules to reach conclusions), and self-correction. AI aims to create machines that can perform tasks requiring human intelligence, such as learning, problem-solving, and decision-making.

➤ **Advantages of AI:-**

- Automates tasks and reduces human error
- Enhances decision-making.
- Improves customer experience.
- Enables predictive analytics.

➤ **Disadvantages of AI:-**

- High development cost.
- Requires large data sets.
- Ethical concerns (bias, job loss).
- Difficult to interpret (black-box models).

➤ **Application of Artificial Intelligence:-**

- Healthcare: Diagnostic tools, patient monitoring.
 - Finance: Fraud detection, algorithmic trading.

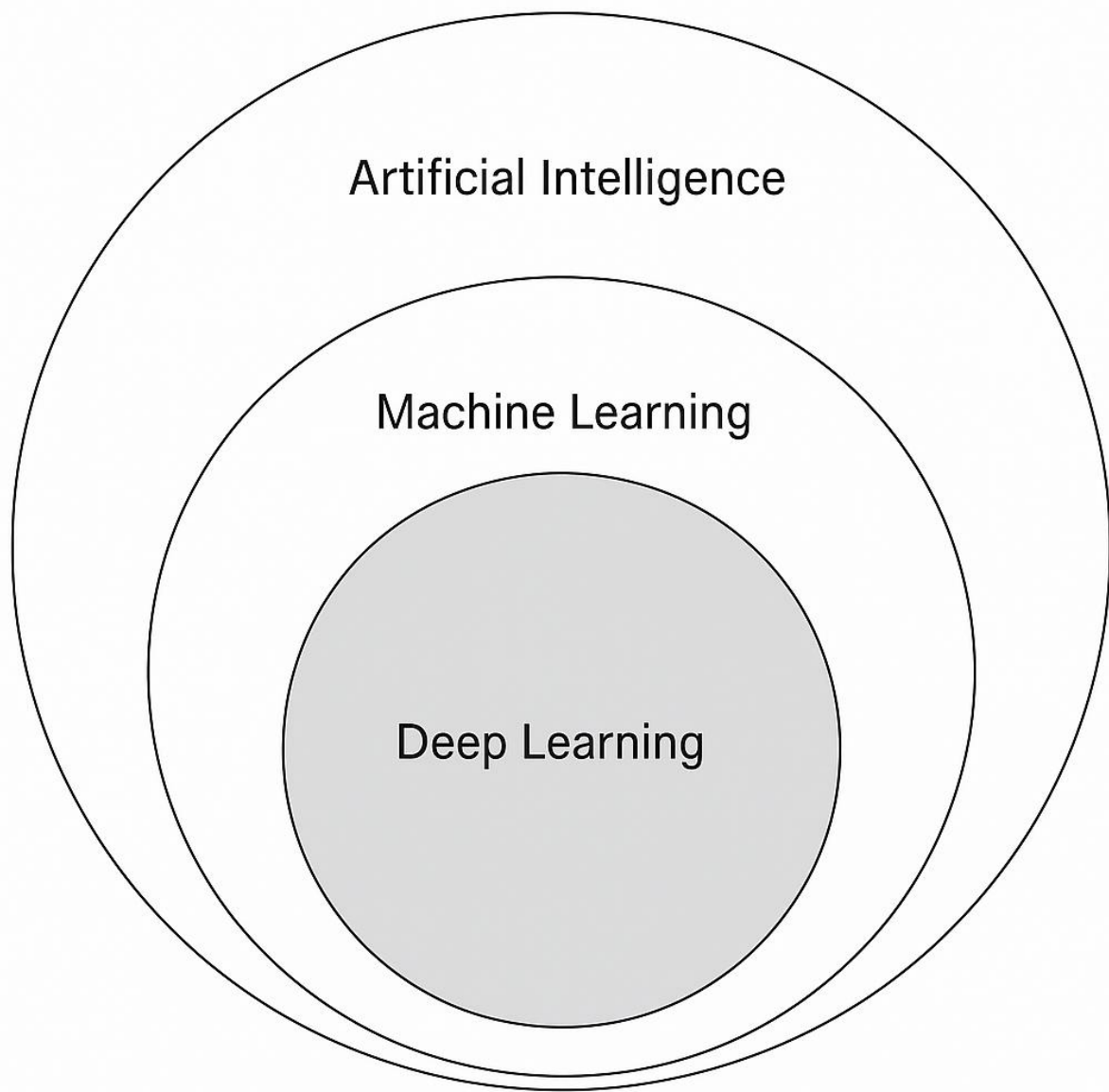


Diagram:-Artificial Intelligence

2.2 Internet of things(IOT):-The internet of Things(IoT) is a network of device that are connected to the internet and can communication with each other. These device are equipped with sensors and other technologies that allow them to collect and exchange data.

➤ **Advantages:-**

- Enables real-time monitoring and control.
- Improves operational efficiency.
- Automates complex processes.
- Enhances user convenience and safety.

➤ **Disadvantages:-**

- Security and privacy concerns.
- Scalability and compatibility issues.
- High initial setup cost.
- Network dependency.

➤ **Application of Internet of things:-**

- Smart homes: Thermostats, lights, security systems.
- Healthcare: Remote patient monitoring.
- Agriculture: Smart irrigation systems.

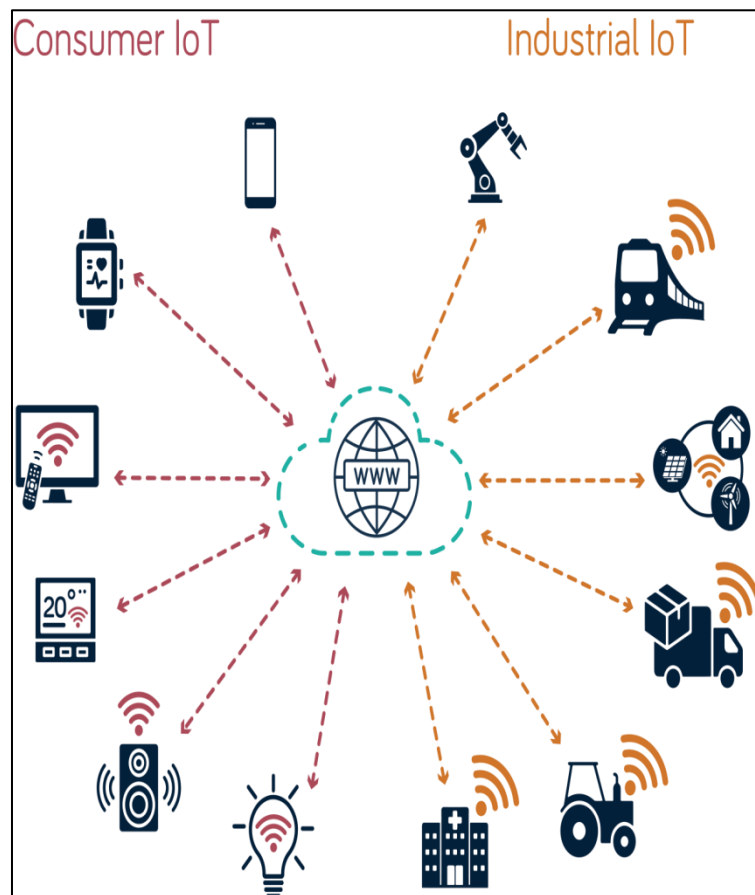
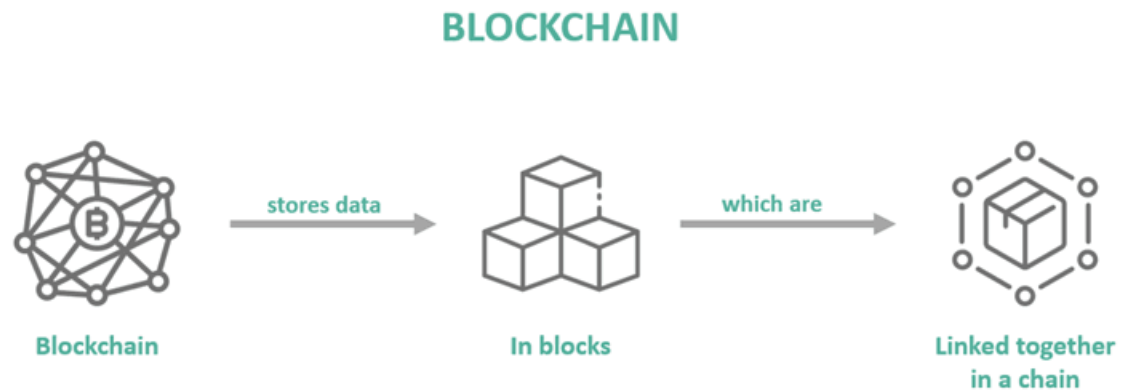


Diagram:-IOT

2.3.Blockchain:-Blockchain is a distributed ledger technology where data is stored in blocks linked chronologically. It ensures transparency, immutability, and decentralization, making it secure and reliable for transactions.



➤ **Advantages:-**

- Ensures data integrity and transparency.
- Eliminates need for central authority.
- Reduces fraud and cyber threats.
- Enables smart contracts and automation.

➤ **Disadvantages:-**

- High energy consumption .
- Complex implementation.
- Regulatory challenges.
- Limited scalability.

➤ **Applications:-**

- Supply Chain: Provenance tracking.
- Healthcare: Secure patient data sharing.
- Voting Systems: Tamper-proof election result

2.4. Generative AI (GenAI): GenAI refers to models that can create new content based on training data. These include language models (e.g., GPT), image generators (e.g., DALL·E), and code generators.

*4

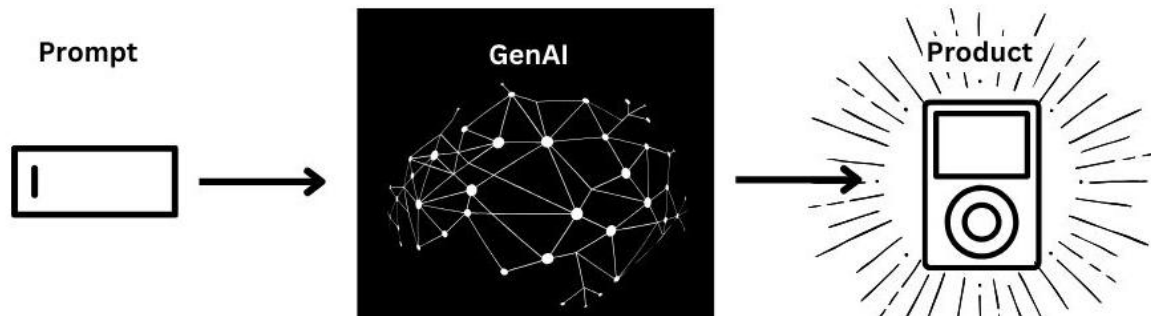


Diagram:-GenAI

➤ **Advantages:-**

- Speeds up content creation.
- Enhances creativity and innovation.
- Personalizes digital experiences.
- Assists in coding, writing, and designing.

➤ **Disadvantages:-**

- Risk of plagiarism and misinformation.
- Bias in generated content.
- Copyright and ethical issues.
- May require human review for accuracy.

➤ **Applications:-**

- Content generation: Blogs, social media posts.
- Design: Art, UI mockups.
- Education: Summarization, tutoring.
- Software development: Code generation, debugging.

Qus.3 Case study of Data Warehouse and Machine Learning.(Theory, Advantages, Disadvantages, Applications and Diagram).

Ans:-3.1.Data Warehouse:-A Data Warehouse (DWH) is a centralized system designed to store, process, and analyze large volumes of historical data from different sources. It is optimized for querying and reporting, rather than transaction processing. A data warehouse is a centralized repository that stores and organizes data from various sources, enabling businesses to perform in-depth analysis and reporting for better decision-making.

➤ **Advantages:-**

- High performance.
- Historical data analysis.
- Data integration.
- Time saving

➤ **Disadvantages**

- High initial cost and setup time.
- Complex maintenance and management.
- Not suitable for real-time data.

➤ **Application of data warehouse:-**

1.Banking and Finance:-

- Fraud detection
- Risk analysis
- Loan performance tracking

2.Retail and E-Commerce:-

- Customer behavior analysis
- Sales forecasting
- Personalized marketing

3.Healthcare:-

- Cost management.
- Clinical reporting.

4.Travel and Transportation:-

- Customer preferences tracking.
- Demand forecasting.

5. Government:-

- Public health trend monitoring.

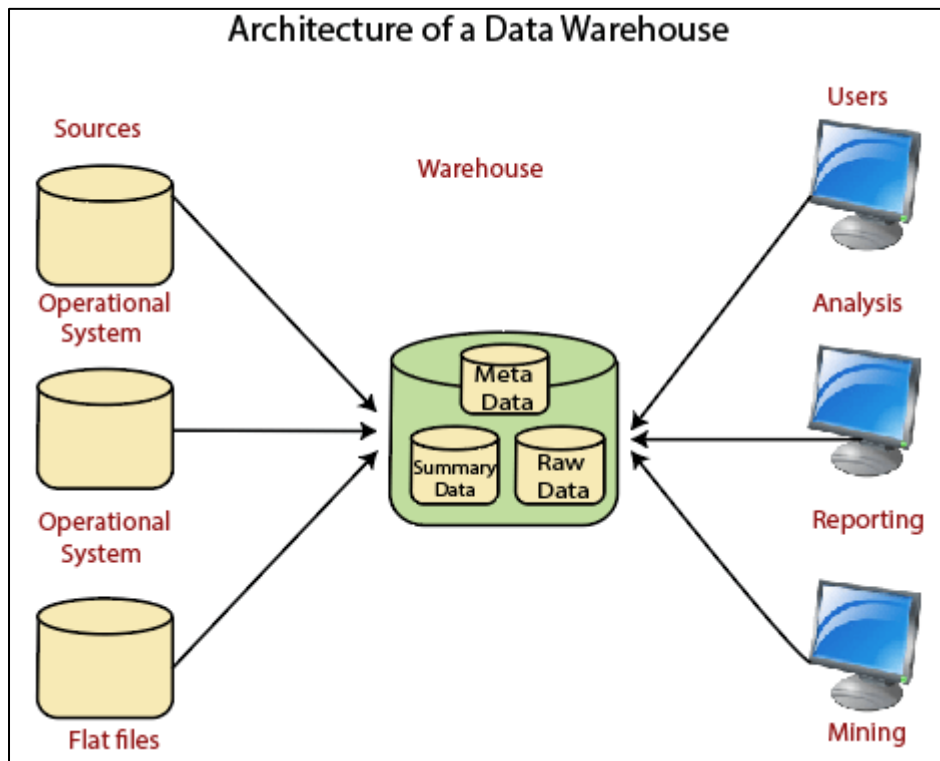


Diagram:-Data Warehouse

3.2.Machine Learning:-Machine learning is a type of artificial intelligence(AI) that allows computers to learn from data and improve their performance without being programmed. Machine learning main focused on developing algorithms that can learn from data and make prediction or decision without human intervention.

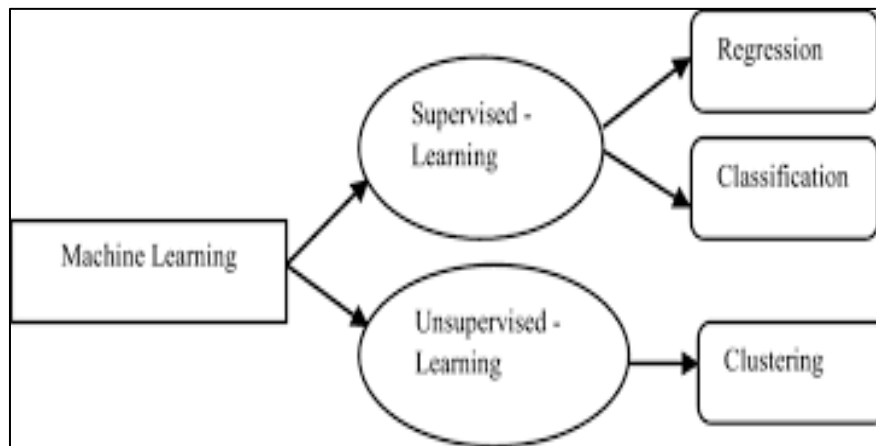


Diagram:-Machine Learning

➤ **Advantages of ML:-**

- **Automation:-** ML systems can automate repetitive tasks without human intervention.
- **Improves Over Time:-** Models learn and get better as more data is provided continuous improvement.
- **Handles Larges Data:-** amounts of data that humans cannot the read.
- **Pattern Recognition :-** Detects hidden patterns or trends in data, useful in prediction.

➤ **Disadvantages of ML:-**

- **Data Dependency :-** ML needs a large amount of quality data for training.
- **High Computational cost:-** Some ML models (like deep learning) require powerful hardware.
- **Over fitting:-** A model may perform well on training data but fail on new or unseen data.

➤ **Applications of Machine Learning:-**

- Healthcare.
- E-commerce .
- Finance & Banking.

Que. 4 To study various technical skills which are currently in demand for industry.

Ans:- 1.Programming Languages:-

- Python, JavaScript, Java.

2.Web Development:-

- HTML, CSS, JavaScript, React.js, Node.js.

3.Data Science & Machine Learning:-

- **Skills:** Python (NumPy, pandas), SQL, data visualization, ML algorithms, deep learning.
- **Tools:** Jupyter Notebook.

4.Cloud Computing:-

- **Platforms:** AWS, Azure, Google Cloud..

5.Cybersecurity:-

- **Skills:** Network security, ethical hacking, cryptography, incident response.

6.DevOps & Automation:-

- **Kills:** CI/CD pipelines, Git.
- **Cloud + Scripting:** Shell scripting, Python.
- **Tools:** GitHub Action.

7.UI/UX Design:-

- **Skills:** Wireframing, prototyping, user research.
- **Tools:** Figma, Sketch.

8.Database Management:-

- MySQL, MongoDB.

9.Mobile App Development:-

- Flutter, React Native.

10.Version Control:-

- Git & GitHub

Qus.5 An Introduction of various design patterns in software development(Theory of UML Diagram).

1. Structure
2. Behaviour

Ans:-UML:- UML (Unified Modeling Language) is a standardized modeling language used to visualize, specify, construct, and document the structure and behavior of software systems. It helps developers and stakeholders understand, design, and communicate software architecture effectively.

Purpose of UML:-

- **Visualization** of a system's architecture.
- **Specification** of software components and their interactions.
- **Construction** by providing a blueprint for coding.
- **Documentation** of design decisions and structures.

Types of UML :-

1.Structural :-

- **Class Diagram** – Shows classes, attributes, methods, and relationships.
- **Component Diagram** – Describes organization and dependencies among software components.
- **Deployment Diagram** – Describes the physical deployment of artifacts on nodes.
- **Composite Structure Diagram** – Shows internal structure of a class.

2.Behavioral :-

- **Use Case Diagram** – Shows system functionality and user interactions.
- **Sequence Diagram** – Shows object interactions over time.
- **Activity Diagram** – Describes workflows or processes.

Various Design pattern:-

1. Creational Patterns:- Concerned with object creation.

- **Singleton Pattern:-** Ensures only one instance of a class is created.
- **Factory Method Pattern:-** Creates objects without exposing the instantiation logic to the client.
- **Builder Pattern:-** Builds complex objects step-by-step.
- **Prototype Pattern:-** Creates new objects by copying an existing object.

- **Abstract Factory Pattern:-** Provides an interface for creating related objects without specifying their concrete classes.

2. Structural Patterns:- Concerned with how classes and objects are composed.

- Adapter Pattern:- Converts one interface into another that clients expect.
- Facade Pattern:- Provides a simplified interface to a complex subsystem.
- Decorator Pattern:- Adds functionality to an object dynamically.
- Proxy Pattern:- Provides a placeholder to control access.

3. Behavioral Patterns:- Concerned with communication and responsibility between objects.

- Observer Pattern:- One object notifies others about state changes.
- Strategy Pattern:- Defines a family of algorithms and lets you choose one at runtime.
- Command Pattern:- Encapsulates a request as an object.
- State Pattern:- Alters behavior when the object's state changes.
- Chain of Responsibility Pattern:- Passes a request along a chain until it's handled.

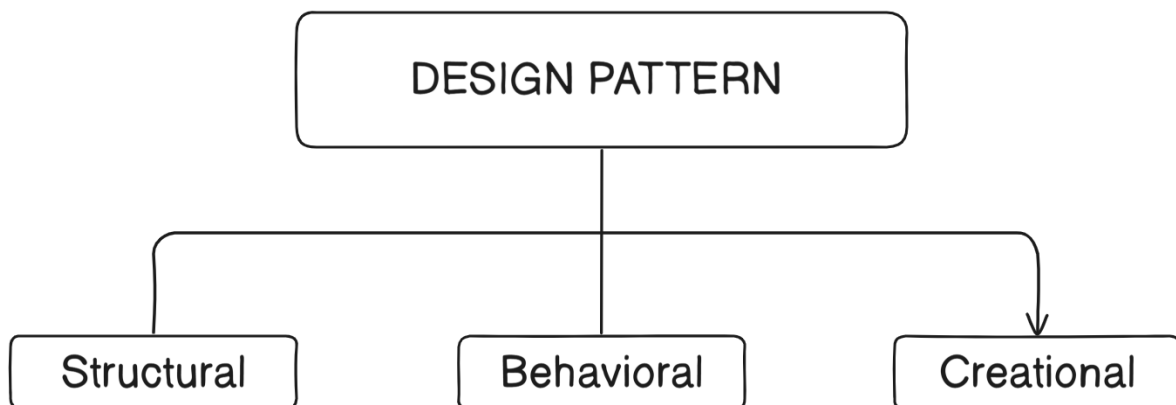


Diagram:-Design Pattern

Qus .6 To study various software development standards.

Ans:- 6.1. Software Life Cycle Processes:-

- Covers all phases: requirement analysis, design, coding, testing, deployment, and maintenance.
- Ensures a structured approach to software development.

6.2. Software Requirements Specification (SRS):-

- Provides a structure for writing clear, complete, and consistent requirements.
- Helps bridge the gap between clients and developers.

6.3. Test Documentation Standard:-

- Describes standard formats for test plans, test cases, test logs, and test reports.
- Useful for ensuring systematic testing.

6.4. Software Quality Model:-

- Defines eight quality attributes:
- Functional Suitability, Performance Efficiency, Security, Maintainability, Compatibility.

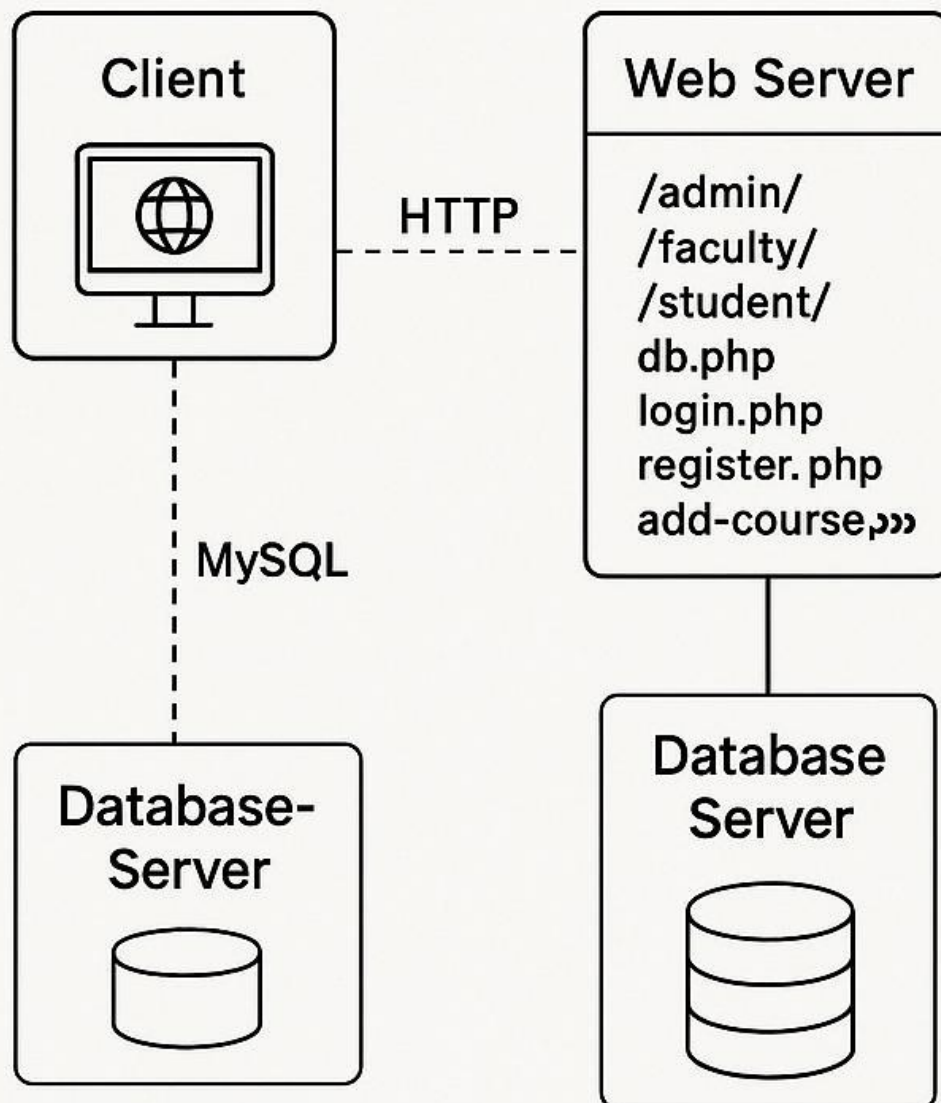
6.5.Information Security:-

- Focuses on developing software that protects data and prevents breaches.
- Essential in sectors like banking, education, and healthcare.

Qus.7 To study and Draw Deployment Diagram and ER Diagram for our project.

Deployment Diagram is a type of UML diagram that shows the hardware (nodes) and how software components are deployed on those nodes.

Deployment Diagram



Deployment Diagram

Fig.-Deployment Diagram.

ER Diagram:- An **ER Diagram** is a type of flowchart that shows the **entities** (things, objects, or concepts) involved in a system and the **relationships** between them.

Key Components of an ER Diagram:-

1. Entity:

- A real-world object or concept that can have data stored about it.
- Represented by a **rectangle**.
- Example: Student, Course, Employee.

2. Attributes:

- Properties or characteristics of an entity.
- Represented by an **ellipse**.
- Example: Name, ID, Email.

3. Primary Key:

- A unique identifier for each entity instance.
- Underlined in the diagram.

4. Relationship:

- Association between entities.
- Represented by a **diamond**.
- Example: Enrolled, Works_For, Teaches.

5. Cardinality:

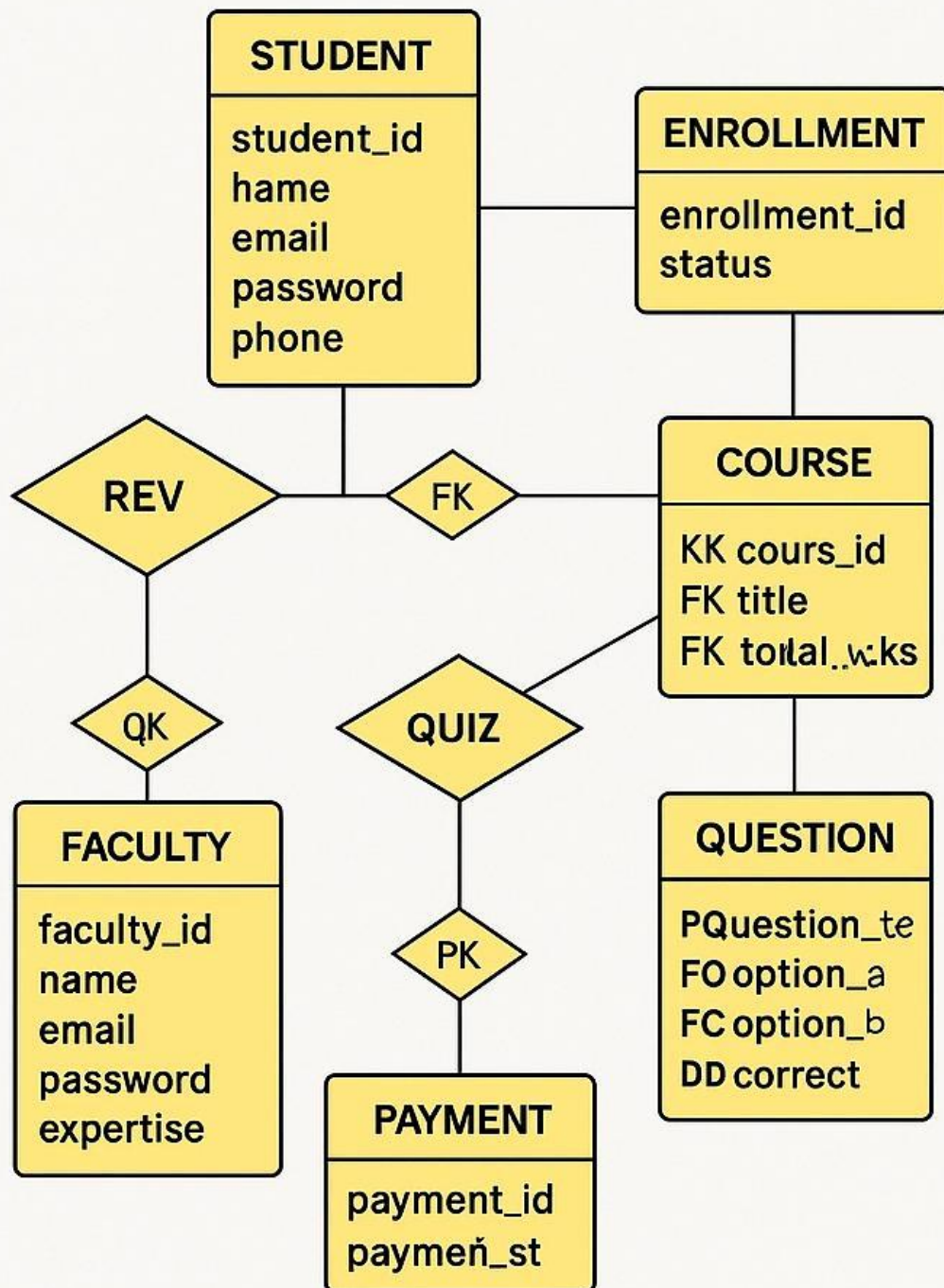
- Describes the number of entity instances that can be associated.
- Example: One-to-One, One-to-Many, Many-to-Many.

6. Weak Entity:

- An entity that cannot exist without another entity.
- Represented by a **double rectangle**.

7. Multivalued Attribute:

- An attribute that can have multiple values.
- Represented by a **double ellipse**.



ER Diagram

Fig.:-ER diagram

Qus.8 To study component diagram and Draw component diagram for our project.

A **Component Diagram** in UML (Unified Modeling Language) shows how a system is divided into components and how they interact through interfaces. It's used mainly in the design phase to represent the physical structure of the system.

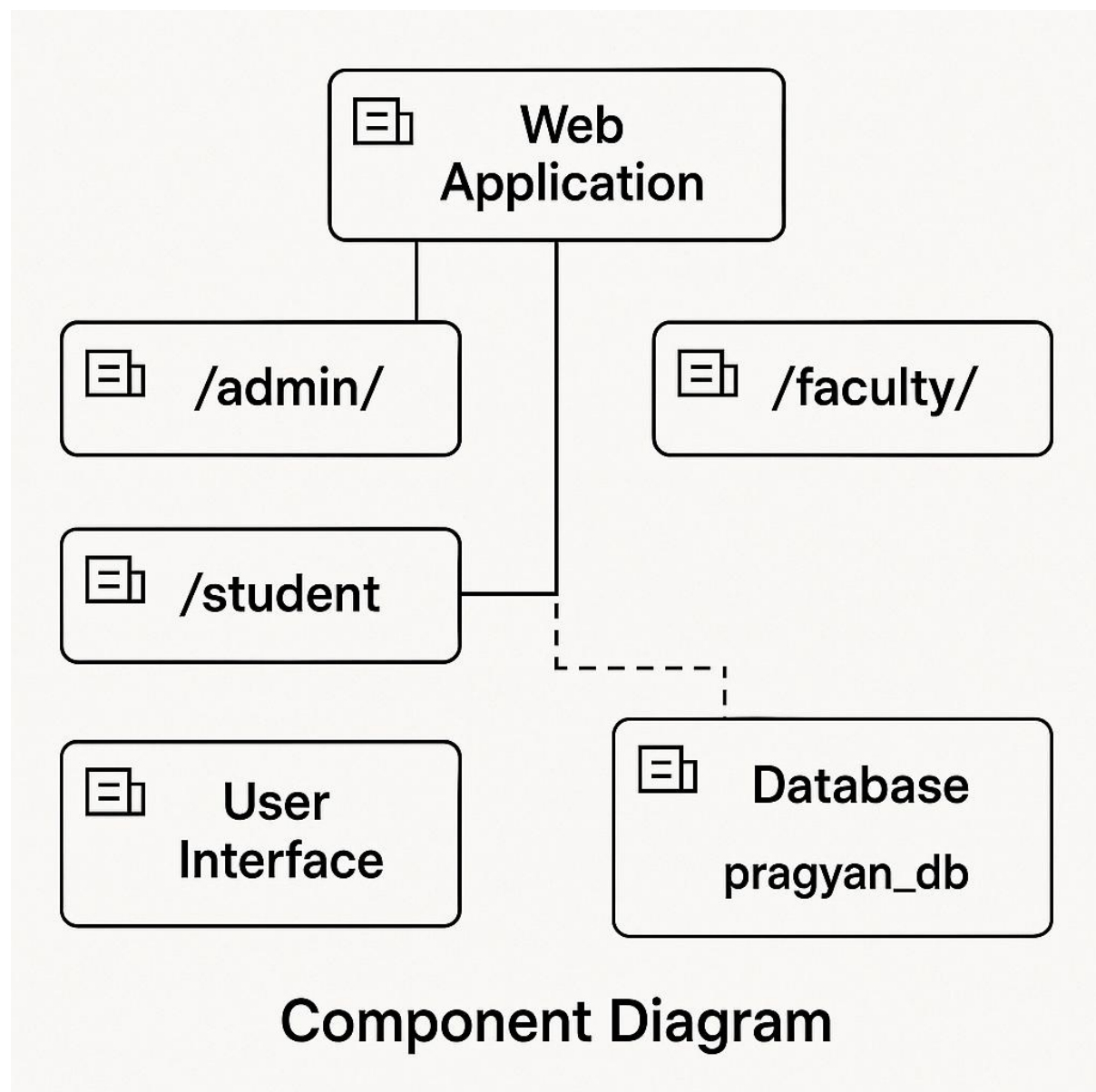
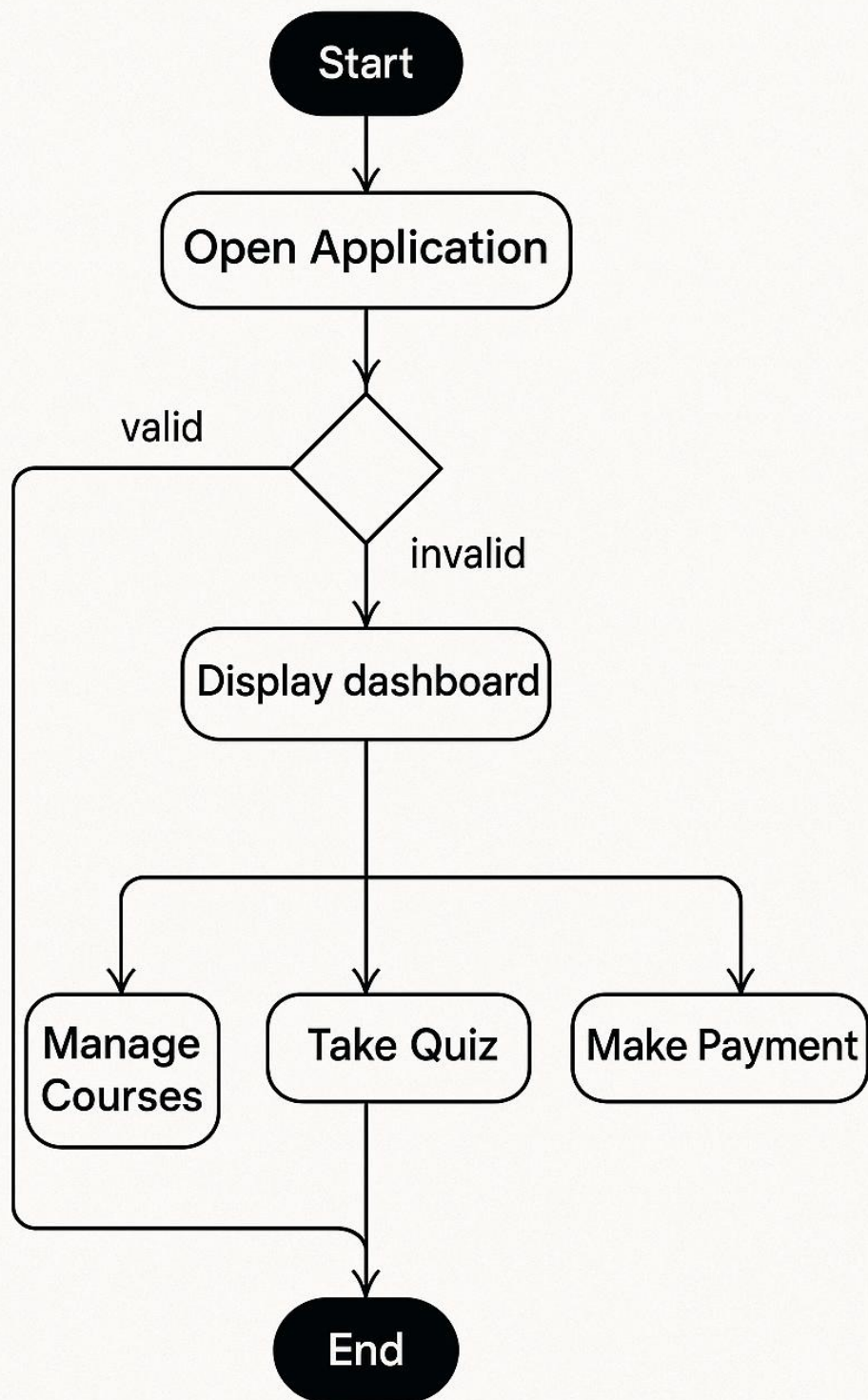


Fig.:-Component Diagram

Qus.9 To study Activity diagram and Draw Activity diagram for your project.

An **Activity Diagram** is a type of behavioral diagram in UML (Unified Modeling Language) that visually represents the flow of activities or actions in a system or process.



Activity Diagram

Qus.10 To study use case diagram and Draw use case diagram for your project.

A **use case diagram** visually represents the interactions between a system and its users (actors), illustrating how users interact with the system to achieve specific goals. These diagrams are a key part of the Unified Modeling Language (UML) and help stakeholders understand the system's functionality and requirements. They show the various use cases (sets of actions) the system performs in response to user interactions.

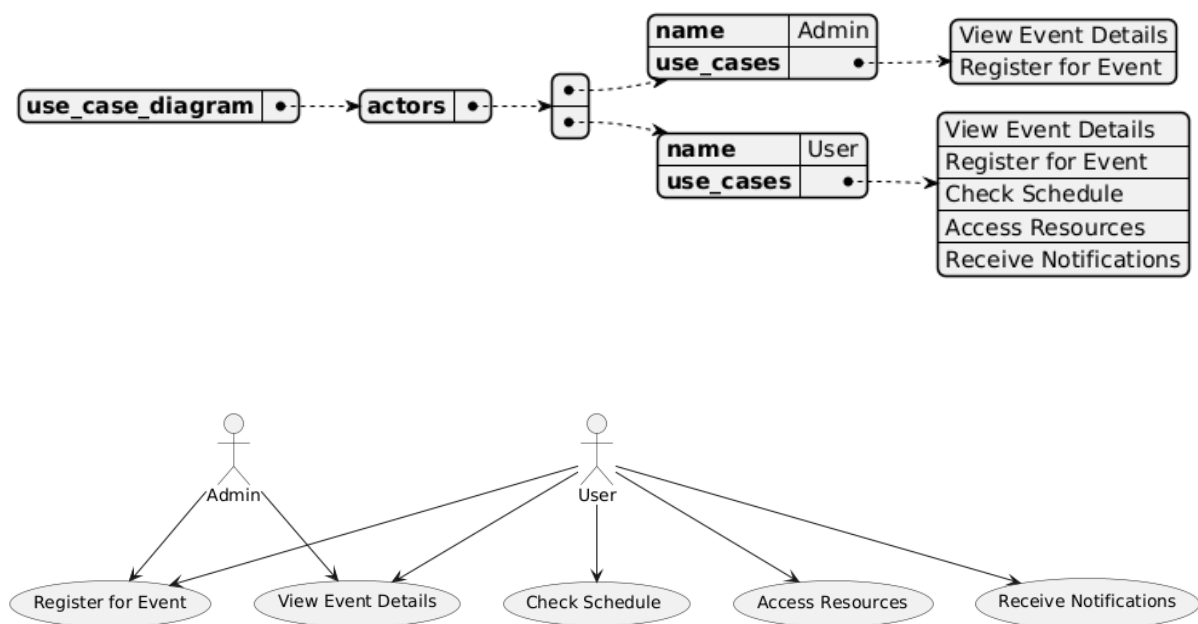


Fig.:-Use case diagram

