ARCHITECTURAL STYLES AND PATTERNS IN SOFTWARE ENGINEERING

PRESENTED BY:

NAME : A.SANTHOSH

ROLL NO : 24671A6701

BRANCH : CSE(DS)

INTRODUCTION

- SOFTWARE ARCHITECTURE: THE HIGH-LEVEL STRUCTURE AND ORGANIZATION OF A SOFTWARE SYSTEM.

- IMPORTANCE: DETERMINES COMMUNICATION, SCALABILITY, MAINTAINABILITY, AND SYSTEM PERFORMANCE.

ARCHITECTURAL STYLES – OVERVIEW.

- ARCHITECTURAL STYLE: A COLLECTION OF DESIGN DECISIONS SHAPING COMPONENT STRUCTURE AND INTERACTIONS.

- ROLE: SETS THE FOUNDATION FOR RELIABILITY, FLEXIBILITY, AND SYSTEM EVOLUTION.

COMMON ARCHITECTURAL STYLES .

- **LAYERED: SEPARATE RESPONSIBILITIES ACROSS TIERS**

- **CLIENT-SERVER: CLIENTS INTERACT WITH CENTRAL SERVER**

- **EVENT-DRIVEN: COMMUNICATION VIA EVENTS**

- **MICROSERVICES: INDEPENDENT, LOOSELY-COUPLED SERVICE**

LAYERED ARCHITECTURE.

- **THREE LAYERS: PRESENTATION, BUSINESS LOGIC, DATA.**

- **PROS: EASY MAINTENANCE, CLEAR RESPONSIBILITIES.**

- **CONS: POTENTIAL SLOWDOWNS, LESS FLEXIBILITY.**

CLIENT-SERVER ARCHITECTURE.

- MULTIPLE CLIENTS CONNECT TO SINGLE SERVER.

- PROS: CENTRALIZED MANAGEMENT, EASY SCALING.

- CONS: SERVER BOTTLENECK, SINGLE POINT OF FAILURE.

MICROSERVICES ARCHITECTURE.

- SERVICES INTERACT VIA APIS, EACH WITH OWN DATA STORE.

- ADVANTAGES: SCALABILITY, FAULT ISOLATION, TECH DIVERSITY.

- CHALLENGES: COMPLEX DEPLOYMENT, COMMUNICATION OVERHEAD.

- KEY FEATURES: INDEPENDENT DEPLOYMENT, DIVERSE TECHNOLOGY STACKS.

ARCHITECTURAL PATTERNS – OVERVIEW.

- ARCHITECTURAL PATTERN: A REUSABLE SOLUTION TO A COMMON STRUCTURAL SYSTEM PROBLEM.

- DIFFERENCE FROM STYLES: PATTERNS SOLVE A SPECIFIC CHALLENGE; STYLES ARE THE BIG-PICTURE BLUEPRINT.

COMMON ARCHITECTURAL PATTERNS.

- MODEL-VIEW-CONTROLLER (MVC): SEPARATES DATA, USER INTERFACE, AND LOGIC.

- PUBLISH-SUBSCRIBE: COMPONENTS BROADCAST OR RECEIVE MESSAGES BASED ON INTEREST.

- BROKER PATTERN: A BROKER MEDIATES MESSAGES BETWEEN CLIENTS AND PROVIDERS.

- PIPES & FILTERS: DATA PROCESSED IN DISCRETE STEPS THROUGH A SEQUENCE OF MODULES

COMPARING STYLES VS PATTERNS.

- **STYLES: MACRO/SYSTEM STRUCTURE (E.G., LAYERED, MICROSERVICES).**

- **PATTERNS: SYSTEM-LEVEL SOLUTIONS (E.G., MVC, BROKER).**

- **DESIGN PATTERNS: CODE-LEVEL BEST PRACTICES (E.G., FACTORY, SINGLETON).**

CONCLUSION.

- ARCHITECTURE IMPACTS SYSTEM SUSTAINABILITY AND EFFICIENCY.

- CHOICES AFFECT WEB, MOBILE, AND ENTERPRISE SOLUTIONS.

- REFERENCE KEY TEXTS, TRUSTED WEBSITES, CLASS NOTES.