

System Architecture

- Siva Gopalan , Sindhuja

Agenda

- Introduction
- System Architecture Design
- Front End (React)
- Backend (Go)

Agenda

- Database (MongoDB)
- Deployment and Scalability
- Security and Best Practices
- Conclusion
- Q & A

Introduction

- Overview:
 - Briefly introduce the core technologies: React, Go, MongoDB, and AI.
 - Modern , Scalable , Dynamic and Cost effective.
- Problem Statement:
 - "Build a real-time e-commerce platform that provides personalized product recommendations and intelligent customer support."

System Architecture Design

Front End (React)

- Key Features:
 - Component-based architecture
 - Virtual DOM for efficient updates
 - Rich ecosystem of libraries and tools
- Responsibilities:
 - User interface rendering
 - Handling user input and events
 - State management
 - Routing

Backend (Go lang)

- Key Features:
 - High performance and concurrency
 - Strong typing and static compilation
 - Easy to learn and use
- Responsibilities:
 - API development
 - Data processing and validation
 - Integration with MongoDB
 - AI model integration

Database (Mongo DB)

- Key Features:
 - Flexible schema
 - High scalability and performance
 - Rich query language
- Responsibilities:
 - Data storage and retrieval
 - User data, product information, and AI model data

AI Component

- AI Integration:
 - Explain how the AI component interacts with the backend and frontend.
 - Discuss potential use cases, such as:
 - i. Recommendation Systems: Personalized product suggestions
 - ii. Natural Language Processing: Chatbots and sentiment analysis
 - iii. Machine Learning: Predictive analytics and anomaly detection

AI Component

- AI Framework/Library:
 - Mention the specific AI framework or library used (e.g., TensorFlow, PyTorch, or a cloud-based AI service).

Deployment and Scalability

- Deployment Strategies:
 - Discuss deployment options like containerization (Docker) and cloud platforms (AWS/GCP/Azure).
- Scalability Considerations:
 - Explain how the architecture can scale to handle increased traffic and data.
 - Mention load balancing, caching, and horizontal scaling techniques.

Security and Best Practices

- Security Measures:
 - Data encryption
 - Secure authentication and authorization
 - Input validation and sanitization
 - Regular security audits
- Best Practices:
 - Code quality and testing
 - Continuous integration and continuous delivery (CI/CD)
 - Monitoring and logging

Conclusion

- Recap:
 - Summarize the key points of the presentation.
- Future Directions:
 - Discuss potential future enhancements and integrations.
- Q&A:
 - Open the floor for questions and discussions.

Q & A