FAIZ COMPUTER INSTITUTE

Backend Development Syllabus

1. Introduction to Backend Development

- Overview of backend development and its role in web applications
- Server, databases, and client-server architecture
- Common backend technologies and frameworks (Node.js, Django, Flask, Java, Ruby on Rails, etc.)
- Overview of RESTful APIs, HTTP methods, and request-response cycle
- Setting up the development environment (IDEs, version control with Git)

2. Databases and Data Modeling

- Introduction to databases: Relational vs NoSQL databases
- SQL vs NoSQL (Examples: MySQL, PostgreSQL, MongoDB, Redis)
- Data modeling and schema design
- CRUD operations (Create, Read, Update, Delete) with SQL
- Writing SQL queries and joins (INNER JOIN, LEFT JOIN, etc.)
- Indexing and performance optimization in databases

3. Web Servers and Networking

- Introduction to web servers (Apache, Nginx)
- Understanding the client-server model and HTTP protocol
- Working with REST APIs and understanding REST principles
- Working with JSON and XML data formats
- Setting up a simple backend server (Node.js with Express, Django with Python, etc.)
- CORS (Cross-Origin Resource Sharing) and security concerns

4. Authentication and Authorization

- Introduction to authentication and authorization
- Authentication protocols (OAuth, JWT, OpenID)
- Managing sessions and cookies
- Implementing user authentication using JWT (JSON Web Tokens)
- Role-based access control (RBAC) and permissions
- OAuth 2.0 and integrating third-party authentication (Google, Facebook, etc.)

5. API Development and RESTful Design

- Introduction to RESTful API design principles
- HTTP methods (GET, POST, PUT, DELETE)
- Building a simple REST API (with Node.js/Express, Django, Flask, etc.)
- API response status codes and error handling
- Pagination, filtering, and sorting in APIs
- Versioning and security considerations in APIs

- Rate limiting and throttling APIs

6. Backend Frameworks

- Introduction to backend frameworks and their role (Node.js, Django, Flask, Ruby on Rails, etc.)
- Setting up a backend framework (Node.js with Express, Django setup)
- Routing and middleware concepts
- Request handling and response rendering
- Templating engines (e.g., EJS, Jinja2)
- Working with APIs in backend frameworks

7. Working with NoSQL Databases

- Introduction to NoSQL databases (MongoDB, Cassandra, Firebase)
- Data modeling in NoSQL (document, key-value, graph databases)
- CRUD operations in NoSQL (insert, query, update, delete)
- Integrating NoSQL databases with backend frameworks (Mongoose with Node.js, etc.)

8. Testing and Debugging

- Writing unit tests for backend services (Jest, Mocha, PyTest, etc.)
- API testing using Postman or Swagger
- Integration testing for database interactions
- Debugging techniques using logging, error handling, and stack traces
- TDD (Test-Driven Development) in backend services

9. Deployment and Hosting

- Introduction to cloud hosting and deployment (AWS, Heroku, DigitalOcean)
- Setting up a web server and deploying backend applications
- Continuous Integration and Continuous Deployment (CI/CD)
- Working with Docker for containerization
- Load balancing and handling traffic spikes
- Database deployment and backup strategies
- Monitoring and scaling backend applications

10. Security Best Practices

- Introduction to backend security principles
- Data encryption (AES, RSA) and hashing (bcrypt, SHA)
- Protecting against SQL injection and Cross-Site Scripting (XSS)
- Secure API design and prevention of API abuse
- Implementing HTTPS and securing communication
- User password security and multi-factor authentication (MFA)

11. Asynchronous Programming and Queues

- Introduction to asynchronous programming

- Callbacks, promises, and async/await in JavaScript (Node.js)
- Background job processing using message queues (RabbitMQ, Kafka, Redis)
- Setting up background workers and processing long-running tasks
- Using Redis for caching and queue management

12. Final Projects

- **Project Ideas**:
  - Build a simple blogging platform with user authentication and a RESTful API
  - Develop a task management system with API and backend services
  - Create an e-commerce backend system (product management, cart, orders)
  - Build a real-time chat application with WebSockets and a backend server
  - Design a simple social media API with user profiles, posts, and comments