Mobil app development complete course

With Flutter and dart NodJs backend

For junior and middle level programmer

This course includes

- ✓ **Online teaching through video**
- ✓ **Teaching material sharing**
- ✓ **Weekly question and answer**
- ✓ **Private consulting if needed**
- ✓ **Project support and consulting**
- ✓ **Follow up**
- ✓ **Practical Assignment and project**

By Abyssinia software solution

Contents

This course includes 1
Course Title for backend 4
Course Description: 4
Nodjs Structure**:** 4
Assessment: sample project on prerequisite 4
Introduction to Backend Development 4
Assessment: assignment and discussion 4
Setting Up Development Environment 4
Assessment: setup development environment with student side 4
Node.js Fundamentals 5
Express.js Framework 5
Assessment sample project 5
Working with MySQL Database 5
Authentication and Authorization 5
Assessment sample project 5
Error Handling and Logging 5
Security Best Practices 5
RESTful API Development 6
Assessment sample project 6
Testing and Debugging 6
Deployment and Scalability 6
Performance Optimization 6
Final real world Project Work 6
Software development life cycle 6
Course Duration: 7
Learning Outcomes: 7
Prerequisites: 7
Course Structure for Frontend **:** 7
Week 1: Introduction to Flutter and Dart 8
Sample project 8
Week 2: Flutter Basics 8
Flutter frontend design project 8
Week 3: Intermediate Flutter Development 8

Week 4: Advanced UI Design and Animations 8
Sample advanced project with discussion 9
Week 6: Integrating Third-party Packages 9
Week 7: Testing and Debugging 9
Week 8: Deployment and Publishing 9
Week 9-10: Final Project 9
Learning Outcomes: 10
Assessment and engagement: 10

Course Title for backend

Backend Development with Node.js and MySQL

Course Description:

This course provides an in-depth exploration of backend web development using Node.js and MySQL. Students will learn to build robust server-side applications, interact with databases, and deploy scalable solutions. The course caters to both junior developers looking to gain foundational skills and senior developers aiming to enhance their expertise in backend development.

Nodjs Structure:

Prerequisites:

- Basic knowledge of HTML, CSS, and JavaScript.
- Familiarity with web development concepts and terminology.
- Prior experience with programming languages is beneficial but not mandatory.

Assessment: sample project on prerequisite

Introduction to Backend Development

- Overview of backend development concepts
- Role of backend in web applications
- Introduction to Node.js and MySQL

Assessment: assignment and discussion

Setting Up Development Environment

- Installing Node.js and MySQL
- Setting up a development server
- IDEs and development tools for Node.js

Assessment: setup development environment with student side

Node.js Fundamentals

- Introduction to Node.js and JavaScript runtime environment
- Asynchronous programming with callbacks, promises, and async/await
- Node Package Manager (npm) and package management

Express.js Framework

- Introduction to Express.js and its role in backend development
- Routing and handling HTTP requests
- Middleware and request processing pipelines

Assessment sample project

Working with MySQL Database

- Introduction to MySQL and relational databases
- Connecting Node.js with MySQL using libraries (e.g., mysql, sequelize)
- Performing CRUD operations: Create, Read, Update, Delete

Authentication and Authorization

- User authentication strategies (e.g., sessions, JWT)
- Role-based access control (RBAC)
- Implementing authentication middleware

Assessment sample project

Error Handling and Logging

- Handling errors in Express.js applications
- Logging techniques for debugging and monitoring
- Error reporting and exception handling best practices

Security Best Practices

- Understanding common security vulnerabilities (e.g., SQL injection, XSS)
- Implementing security measures (e.g., input validation, parameterized queries)

- Securing APIs with HTTPS and SSL/TLS

RESTful API Development

- Principles of RESTful architecture
- Designing RESTful APIs
- Implementing CRUD operations with RESTful routes

Assessment sample project

Testing and Debugging

- Writing unit tests with frameworks like Mocha and Chai
- Integration testing with Super test
- Debugging Node.js applications using built-in tools and third-party libraries

Deployment and Scalability

- Deployment strategies for Node.js applications
- Containerization with Docker
- Load balancing and scaling with tools like Nginx and PM2

Performance Optimization

- Identifying performance bottlenecks in Node.js applications
- Techniques for improving application performance
- Caching strategies with Redis

Final real world Project Work

Software development life cycle

- Project **Planning**: Define requirements, user stories, and timeline.
- Technology **Stack**: Select appropriate frameworks and technologies.
- Team **Formation**: Form teams for collaborative development.
- Development **Environment Setup**: Assist with setting up development environments.

- Iteration **and Feedback**: Foster iterative development with regular feedback.
- Documentation: Emphasize code and technical documentation.
- Testing **and Quality Assurance**: Teach testing techniques and code quality practices.
- Deployment **and DevOps**: Cover deployment and CI/CD concepts.

- Project **Presentation**: Organize a showcase or demo day for project presentations.

Course Duration:

The course duration can vary depending on the format (e.g., weeks, months, or hours of content), but it's recommended to cover the material over a span of 12 to 16 weeks for thorough understanding and practical application.

Learning Outcomes:

By the end of this course, students will:

- Have a solid understanding of backend development principles and technologies.
- Be proficient in building RESTful APIs using Node.js and Express.js.
- Gain hands-on experience in interacting with MySQL databases.
- Understand authentication, authorization, and security best practices.
- **Be able to design, develop, test, deploy, and maintain scalable backend applications.**

Prerequisites:

- Basic knowledge of HTML, CSS, and JavaScript.
- Familiarity with web development concepts and terminology.
- Prior experience with programming languages is beneficial but not mandatory.

Course Structure for Frontend :

Week 1: Introduction to Flutter and Dart

- **Day 1-2:**
  - Overview of Flutter and its capabilities
  - Installation and setup of Flutter and Dart
  - Introduction to Dart programming language
- **Day 3-4:**
  - Dart fundamentals: variables, data types, and operators
  - Control structures and functions in Dart
- **Day 5:**
  - Object-oriented programming in Dart
  - Classes, inheritance, and interfaces

Sample project

Week 2: Flutter Basics

- **Day 6-8:**
  - Introduction to Flutter widgets
  - Building layouts with Flutter
  - Understanding stateful and stateless widgets
- **Day 9-10:**
  - Handling user input and gestures
  - Navigation and routing

Flutter frontend design project

Week 3: Intermediate Flutter Development

- **Day 11-13:**
  - Working with lists and grids
  - Managing state with provider and setState
- **Day 14-15:**
  - Networking: Fetching data from APIs
  - Parsing JSON data

Week 4: Advanced UI Design and Animations

- **Day 16-18:**
  - Custom widgets and theming
  - Advanced layout techniques
- **Day 19-20:**
  - Adding animations to Flutter apps
  - Using the animation and motion widgets

Sample advanced project with discussion

Week 5: Data Persistence

- **Day 21-23:**
  - Local storage with SQLite
  - Using shared preferences for simple key-value storage
- **Day 24-25:**
  - Implementing Firebase Firestore for real-time data
  - Authentication with Firebase

Week 6: Integrating Third-party Packages

- **Day 26-28:**
  - Exploring and using popular Flutter packages
  - Integrating Google Maps and location services
- **Day 29-30:**
  - Using camera and image picker packages
  - Push notifications with Firebase Cloud Messaging

Week 7: Testing and Debugging

- **Day 31-33:**
  - Writing unit tests in Flutter
  - Widget and integration testing
- **Day 34-35:**
  - Debugging techniques and tools
  - Performance optimization

Week 8: Deployment and Publishing

- **Day 36-38:**
  - Preparing Flutter apps for deployment
  - Building and releasing apps for iOS and Android
- **Day 39-40:**
  - Continuous Integration/Continuous Deployment (CI/CD) for Flutter apps
  - Publishing apps on Google Play Store and Apple App Store

Week 9-10: Final Project

- **Day 41-55:**
  - Capstone project: Planning, design, and implementation
  - Individual or group project development phase
- **Day 56-60:**
  - Project testing and debugging
  - Project presentation and peer review

Learning Outcomes:

By the end of this course, students will:

- Have a solid understanding of Flutter and Dart programming language.
- Be able to build and deploy high-quality mobile applications.
- Understand state management, API integration, and data persistence.
- Be proficient in advanced UI design, animations, and custom widget creation.
- Be able to test, debug, and optimize Flutter applications.

Assessment and engagement:

Video teaching

Assignments and quiz

Real world project

Live QA

VIP teaching