

Software Developer Bootcamp (JavaScript & Python)

Completion Period: 9

This course is tailored for anyone who wants to build a career as a software developer. During the course, the students will be trained on the following languages/technologies;-

- HTML.
- CSS.
- Bootstrap.
- JavaScript.
- Python.
- Django.
- Django REST Framework.
- Databases, both SQL databases(PostgreSQL) and NoSQL(MongoDB).
- Version Control, using Git, GitHub and GitLab(optionally).
- ReactJs Basics.

On top of being trained on technical skills, the students will also be trained on soft skills such as;-

- Collaboration.
- Team work.
- Communication.
- Creating linkedin profiles.
- Creating resumes.
- Networking.

Chapter 1: Web Development, Part 1

Module 1: HTML

- Overview of web development
- Introduction to HTML
 - Basic structure
 - Tags and elements
 - Semantic HTML
- Hands-on exercises

Module 2: CSS

- Understanding CSS fundamentals
 - Selectors, properties, and values
 - Box model
 - Layout techniques.
- Introduction to CSS frameworks like Bootstrap
- Hands-on exercises

Module 3: JavaScript

- Introduction to JavaScript
 - Variables, data types, and operators
 - Control structures (if statements, loops)
 - Functions and scope
- DOM manipulation
- Events and event handling
- Hands-on exercises and projects

Project 1: HTML, CSS & JavaScript Project

This is a project that will aim to assess students understanding of HTML, CSS & JavaScript

Module 4: Responsive Web Design with Bootstrap

- Introduction to Bootstrap framework
- Grid system and responsive design principles
- Components and utilities
- Hands-on exercises and projects

Project 2: Bootstrap Project

This is a project that will aim to assess students' understanding of bootstrap.

Chapter 2: Python Programming

Module 1: Introduction to Python

- What is Python?.
- Setting up a Python environment (IDEs, text editors).
- Basic syntax and data types (strings, numbers, booleans).
- Variables and data structures (lists, tuples, dictionaries).

Module 2: Control Flow

- Conditional statements (if, else, elif) Loops (for loops, while loops).
- Control flow statements (break, continue).

Module 3: Functions

- Defining and calling functions.
- Function arguments and return values.
- Built-in and user-defined modules.

Module 4: Working with Files

- Reading from and writing to files.
- Working with different file formats (text files, CSV, JSON).

Module 5: Handling Errors

- Understanding exceptions.
- Handling exceptions using try-except blocks.
- Raising custom exceptions.

Module 6: Object Oriented Programming

- Classes and objects.
- Attributes and methods.
- Encapsulation, inheritance, and polymorphism.

Module 7: Data Structures

- Sets and frozensets More about dictionaries (default dictionaries, OrderedDict)
- List comprehensions, generator expressions.
- Queues, Stacks, LinkedLists etc.

Module 8: Advanced Functions

- Lambda functions
- Higher-order functions
- Decorators

Module 9: Working with Databases

- Introduction to databases (SQL vs NoSQL)
- Connecting to and querying databases from Python (SQLite, MySQL, MongoDB)

Module 10: Testing & Debugging

- Writing unit tests with unit test or pytest
- Debugging techniques and tools

Project 3: Python Project

This is a project that will assess students understanding of core Python concepts such as;- Control flow, Loops, Data Structures, Functions & OOP.

Chapter 3: Databases

Module 1: Introduction to Databases

- 1.1. What is a Database?
- 1.2. Types of Databases (Relational, NoSQL, etc.)
- 1.3. Importance of Databases in Software Development

Module 2: Relational Database Management Systems (RDBMS)

- 2.1. Understanding Relational Data Model
- 2.2. Introduction to SQL (Structured Query Language)
- 2.3. Creating Databases and Tables
- 2.4. Basic CRUD Operations (Create, Read, Update, Delete)
- 2.5. Data Integrity Constraints (Primary Keys, Foreign Keys, etc.)

Module 3: Querying and Manipulating Data

- 3.1. Retrieving Data with SELECT Statements
- 3.2. Filtering Data with WHERE Clause

- 3.3. Sorting Results with ORDER BY Clause
- 3.4. Modifying Data with INSERT, UPDATE, and DELETE Statements
- 3.5. Joining Tables to Retrieve Related Data

Module 4: Advanced SQL Concepts

- 4.1. Aggregation Functions (SUM, AVG, COUNT, etc.)
- 4.2. Grouping Data with GROUP BY Clause
- 4.3. Subqueries and Nested Queries
- 4.4. Common Table Expressions (CTEs)
- 4.5. Understanding Transactions and ACID Properties

Module 5: Database Design and Normalization

- 5.1. Principles of Database Design
- 5.2. Entity-Relationship (ER) Modeling
- 5.3. Normalization and Denormalization
- 5.4. Designing Efficient Database Schemas

Module 6: Introduction to NoSQL Databases

- 6.1. What is NoSQL?
- 6.2. Types of NoSQL Databases (Document, Key-Value, Columnar, Graph, etc.)
- 6.3. Comparing NoSQL Databases with Relational Databases
- 6.4. Use Cases and Examples

Module 7: Introduction to Database Administration

- 7.1. Database Administration Tasks
- 7.2. User and Privilege Management
- 7.3. Backup and Recovery Strategies
- 7.4. Monitoring and Performance Tuning

Module 8: Data Security and Privacy

- 8.1. Understanding Data Security Risks
- 8.2. Authentication and Authorization Mechanisms
- 8.3. Encryption and Data Masking Techniques
- 8.4. Compliance with Data Privacy Regulations

Module 9: Database Deployment and Scaling

- 9.1. Deploying Databases on Different Platforms (On-premises, Cloud)
- 9.2. Scalability Considerations
- 9.3. High Availability and Disaster Recovery Strategies

9.4. Containerization and Orchestration (Docker, Kubernetes)

Module 10: Emerging Trends in Databases

- 10.1. Big Data and Distributed Databases
- 10.2. Blockchain and Distributed Ledger Technology
- 10.3. Machine Learning and Data Analytics Integration
- 10.4. Serverless Databases and Microservices Architecture

Module 11: Conclusion and Next Steps

- 11.1. Recap of Key Concepts
- 11.2. Continuing Learning Resources
- 11.3. Practical Applications of Database Skills
- 11.4. Career Paths in Database Management and Administration

Chapter 4: Web Development, Part 2

Module 1: Introduction to Django

- What is Django?
- MVC vs. MTV architecture
- Why use Django for web development?

Module 2: Setting Up Django Development Environment

- Installing Django
- Creating a new Django project
- Understanding project structure

Module 3: Django Models & Databases

- Defining models in Django
- Database migrations with Django ORM
- Querying the database with Django ORM

Module 4: Django Views and Templates

- Introduction to views and templates
- Creating views in Django
- Rendering templates with context data
- Working with template inheritance includes

Module 5: Django Forms

- Creating HTML forms in Django
- Handling form submission and validation
- Using built-in form fields and widgets.

Module 6: Django Admin Interface

- Overview of Django admin interface

- Customizing admin interface
- Adding models to the admin interface

Module 7: Authentication and Authorization

- User authentication in Django
- Custom user models.
- Role-based access control with permissions and decorators.

Module 8: Working with SQL Databases

- Introduction to relational databases
- Basics of SQL (Structured Query Language)
- Creating and querying databases using SQLite
- Integration of Django with databases
- Hands-on exercises and projects

Project 4: Django Project

This is a project that will aim to assess students understanding of Django Web Framework

Chapter 5: Django REST Framework

Module 1: Introduction to Django REST Framework

- 1.1. Overview of RESTful APIs
- 1.2. Introduction to Django and Django REST Framework
- 1.3. Setting up Development Environment
- 1.4. Creating a Django Project
- 1.5. Creating Django Apps for REST API

Module 2: Building RESTful APIs with Django

- 2.1. Serializers and Model Serializers
- 2.2. Views and ViewSets
- 2.3. Routing and URL Configuration
- 2.4. Authentication and Permissions
- 2.5. Pagination and Filtering
- 2.6. Versioning APIs
- 2.7. Handling Errors and Exceptions

Module 3: Working with Data in Django REST Framework

- 3.1. Database Models and Migrations
- 3.2. CRUD Operations with Django ORM
- 3.3. Advanced Querying and Filtering
- 3.4. Serializing Related Data
- 3.5. Nested Serializers

Module 4: Testing and Debugging APIs

- 4.1. Unit Testing with Django REST Framework
- 4.2. Integration Testing APIs
- 4.3. Debugging Techniques
- 4.4. Using Django Debug Toolbar

Module 5: Advanced Topics in Django REST Framework

- 5.1. Customizing Serializers and Views
- 5.2. Advanced Authentication Methods (OAuth, JWT)
- 5.3. Custom Permissions and Authentication
- 5.4. Rate Limiting and Throttling
- 5.5. Working with File Uploads
- 5.6. Caching Responses

Module 6: Building a Project with Django REST Framework

- 6.1. Project Planning and Structure
- 6.2. Designing API Endpoints
- 6.3. Implementing Authentication and Permissions
- 6.4. Testing and Debugging
- 6.5. Deployment Considerations

Module 7: Best Practices and Optimization

- 7.1. Code Organization and Maintainability
- 7.2. Performance Optimization Techniques
- 7.3. Security Best Practices
- 7.4. Documentation Strategies
- 7.5. API Versioning Strategies

Module 8: Real-world Applications and Case Studies

- 8.1. Building a Blogging Platform API
- 8.2. E-commerce API Development
- 8.3. Social Media API with Follow System
- 8.4. Integrating Third-party APIs
- 8.5. Scalability and Load Testing

Module 9: Deployment and Scaling

- 9.1. Preparing Application for Deployment
- 9.2. Choosing Deployment Platforms (Heroku, AWS, etc.)
- 9.3. Containerization with Docker
- 9.4. Continuous Integration and Continuous Deployment (CI/CD)
- 9.5. Scaling Strategies and Considerations

Module 10: Conclusion and Next Steps

- 11.1. Recap of Key Concepts
- 11.2. Continuing Learning Resources
- 11.3. Final Project Showcase and Feedback
- 11.4. Career Opportunities in Django REST Development

Chapter 6: Advanced Concepts

Module 1: Future Trends and Beyond

- 10.1. GraphQL vs. REST
- 10.2. Serverless Architecture with Django
- 10.3. AI and Machine Learning Integration
- 10.4. Microservices with Django
- 10.5. Websockets and Real-time Communication

Module 2: Advanced Topics

- Docker.
- Kubernetes Basics.
- CI/CD using Gitlab CI.
- Version control with Git
- Deployment strategies
- Code review and best practices.
- Performance optimization techniques
- Cloud computing and serverless architecture

Chapter 7: Soft Skills

Module 9: Career Development and Beyond

- Resume building and interview preparation
- Freelancing vs. Full-time employment
- Continuous learning strategies and resources
- Networking and community involvement
- Industry trends and future outlook

Chapter 8: Capstone Project

The students will come up with projects which they will be expected to implement using the above obtained skills under supervision of the instructors.

This course outline is structured to gradually introduce learners to various web development technologies and programming concepts, starting from the basics and progressing towards more advanced topics. Hands-on exercises, projects, and assignments are included throughout the course to reinforce learning and practical application. Additionally, the course covers essential career development aspects to help learners transition into professional software development roles.