# Revised Course Outline

# Full Stack Web Development

# Batch 9

Course Instructors
Saidur Rahman Setu
Asief Mahir
Tanveer Hossain Munim

Support Instructor
Anindita Bose

# Frontend Development Stack

## Languages: HTML,CSS & Javascript

## Frameworks: Bootstrap & ReactJS

# HTML, CSS & JAVASCRIPT

Instructor: Saidur Rahman Setu

## Week 01:

- Class 01: Introduction to HTML and Basic Tags
- Class 02: HTML Semantic Tags and Grouping
- Class 03: Practical Implementations of HTML Concepts

## Week 02:

- Class 04: Introduction to CSS and Important Selectors
- Class 05: CSS Box Model and Fonts
- Class 06: CSS Units and Combinators

## Week 03:

- Class 07: CSS Backgrounds and Links
- Class 08: CSS Specificity and Floating
- Class 09: CSS Forms, Lists and Table

## Week 04:

- Class 10: CSS Position and Gradient
- Class 11: CSS Flexbox
- Class 12: CSS Full Live Project Part-01

## Week 05:

- Class 13: CSS Full Live Project Part-02
- Class 14: CSS Pseudo Class and Elements
- Class 15: CSS Media Queries

## Week 06:

- Class 16: Introduction to CSS Frameworks (Bootstrap)
- Class 17: Exploring Bootstrap Grid System
- Class 18: Practical Project with Bootstrap Grid System

## Week 07:

- Class 19: Introduction to SCSS
- Class 20: Practical Implementations of SCSS and BEM
- Class 21: Introductions to Git and GitHub

## Week 08:

- Class 22: Introduction to Programming Language with JavaScript
- Class 23: Variables and Data Types with JavaScript
- Class 24: JavaScript Operators and Arrays

## Week 09:

- Class 25: JavaScript Conditions and General Loops
- Class 26: Functions in JavaScript
- Class 27: Deep Look into JavaScript Functions

## Week 10:

- Class 28: Object Literals in JavaScript
- Class 29: Factory Functions and Constructor Functions with more Object concepts
- Class 30: Important built-in Methods Part-01

## Week 11:

- Class 31: Important built-in Methods Part-02
- Class 32: Array Methods in JavaScript (Map, Every, Some)
- Class 33: Array Methods in JavaScript (Reduce, Filter, Sort)

## Week 12:

- Class 34: Understanding Execution Context
- Class 35: Introduction to DOM and DOM Selections
- Class 36: DOM Traversal

## Week 13:

- Class 37: DOM Manipulation
- Class 38: DOM Creations
- Class 39: JavaScript Events

# Project List:

### HTML + CSS:

- Personal Portfolio
- Food Store Landing Page
- E-Commerce Landing Page
- App Landing Page

### JavaScript:

- Drum Kit
- Book List
- Weather App
- Calculators

# ReactJS

Instructor: Asief Mahir

- **Week - 1**
    1. Class - 1 ⟶ Introduction to React Fundamentals, Components and Props, Reusable Components, Component Composition in React
    2. Class - 2 ⟶ Understanding is State in React and How to manage it using the “useState” React Hook. Understanding Event Handling, Virtual DOM, and Rendering Mechanism in React
    3. Class - 3 ⟶ Creating a Note-Taking/Todo App. Working with Lists in React. Understanding Conditional Rendering in React
    4. **Assignments**:
        1. Creating a Simple BioData type layout using React Components and Props
        2. Create a Simple Counter-application
        3. Enhancing some features of our Note-Taking/Todo App.
- Week - 2
    1. Class - 4 ⟶ Understanding more about State and “Derived State” in React by Creating a Simple Attendance App (CRUD)
    2. Class - 5 ⟶ Understanding “State Lifting Mechanism in React”
    3. Class - 6 ⟶ Introduction to “Context API” in React - Converting our Attendance App with the help of Context API
    4. **Assignments**:
        1. Add Extra Functionalities to the Todo and Attendance Apps
        2. Re-create the Attendance/Todo App with the State Lifting Mechanism
- Week - 3
    1. Class - 7 ⟶ Introduction to “useReducer” Hook in React ⟶ A more organized way to manage Complex States in React
    2. Class - 8 ⟶ Recreating our Attendance App with Context API and useReducer Hook

   3. Class - 9 → Introduction to the “useEffect” hook in React → Understanding more about the Rendering mechanism, Virtual DOM, and how to handle side-effects in React
   4. **Assignments**:
      1. Recreating the Todo/Note-Taking Application with the Context API
      2. Recreating our Todo/Note-Taking App with Context API and useReducer Hook
- Week - 4
   1. Class - 10 → Creating a full fake API with the “json**-**server” ****package and Connecting APIs with our Note-Taking/Todo App and Recreating the CRUD Operations with API
   2. Class - 11 → Introduction to Frontend Routing in React through the “React Router” Library
   3. Class - 12 → Creating a Project Management App like “Trello” (The boards, lists, and tasks-related, and Drag & Drop features from the Frontend ONLY!) (Part -1)
   4. **Assignments**:
      1. Recreating our Note-Taking/Attendance App by integrating APIs
      2. Add Routing in our Todo/Attendance App
- Week - 5
   1. Class - 13 → Creating a Project Management App like “Trello” (The boards, lists, and tasks-related, and Drag & Drop features from the Frontend ONLY!) (Part - 2)
   2. Class - 14 → Creating a Project Management App like “Trello” (The boards, lists, and tasks-related, and Drag & Drop features from the Frontend ONLY!) (Part - 3)
   3. Class - 15 → Reusing Logics across Components - Understanding “Custom Hooks” in React, Understanding “useRef” Hook in React
   4. **Assignments**:
      1. Enhancing Our Trello Clone Project with additional features
- Week - 6
   1. Class - 16 → Performance bottlenecks and improving techniques in React - “memo Function” “useMemo Hook”, “useCallback Hook”

   2. Class - 17 → Introduction to “Redux” - The organized way to manage ”Global States” in React Applications
   3. Class - 18 → Hands-on Practice with Redux - Creating a Shopping Cart Application
   4. **Assignments**:
      1. Recreating our Trello App with Redux
- Week - 7
   1. Class - 19 → Introduction to “Redux Toolkit Query” - An easier approach for using Redux Logic → Recreating our Shopping Cart App with Redux Toolkit Query
   2. Class - 20 → Introduction to Asynchronous Actions in Redux - Redux Middleware and Thunks
   3. class - 21 → Handling Asynchronous Actions in Redux Toolkit
   4. **Assignments**:
      1. Recreating our Trello App with Redux-Toolkit
- Week - 8
   1. Class - 22 → Tanstack React Query → Understanding “Server States” and how to manage those in our React Apps
   2. Class - 23 → Redux Toolkit Query → An easier approach to manage Server States in Redux Applications
   3. Class - 24 → Introduction to Class Components in React - Working on Legacy projects
   4. **Assignments**:
      1. Add Tanstack React Query to our Note-Taking App
- Week - 9
   1. Class - 25 → HOC (Higher Order Components) - Reusing Logics across Components in Class Components
   2. Class - 26 → Render Props Pattern - Another Beautiful technique for Reusing Logics across Class Components
   3. Class - 27 → Introduction to Next JS - Basics of Next JS (Client-Side Rendering vs Server Rendering, File Based Routing Mechanism, Dynamic Routing)

4. **Assignments**:
    1. Recreate our Attendance App with Class Components

- Week - 10
    1. Class - 28 → Different Rendering Options in Next Js - SSG (Static Site Generation), ISR (Incremental Static Regeneration), SSR (Server Side Rendering)
    2. Class - 29 → Introduction to App Router in Next Js 13
    3. Class - 30 → Final Project (Blog/Single Vendor E-commerce) Part 1
- Week - 11
    1. Class - 31 → Final Project (Blog/Single Vendor E-commerce) Part 2
    2. Class - 32 → Final Project (Blog/Single Vendor E-commerce) Part 3
    3. Class - 33 → Final Project (Blog/Single Vendor E-commerce) Part 4
    4. **Assignments**:
        1. Add Additional Features to our Existing final project

## Frontend Projects (React)

1. Todo/Note-Taking App (Proper CRUD with API)
2. A Simple Attendance App
3. A Project management App like Trello (Main features and from FRONTEND ONLY)
4. A full-featured Shopping Cart App
5. A single-vendor E-commerce/Blog Application with Authentication and Authorization (Both User Facing Application and Admin Dashboard (minimalist) Application)

## Backend Development Stack

**Languages: Python, SQL**

**Framework: Django**

**Database: MySQL**

# Python, Django & SQL

Instructor: Tanveer Hossain Munim

## Python

### Week 1

- **Class 1: Introduction to Python, Installation, and IDE Setup**
    - ❖ Overview of Python and its applications.
    - ❖ Installing Python on different platforms (Windows, macOS, Linux).
    - ❖ Setting up a code editor or IDE (e.g., VSCode, PyCharm).
    - ❖ Writing and executing your first Python program.

- **Class 2: Basic Input/Output, Python Data Types, and Arithmetic Operations**
    - ❖ Understanding Python's input and output functions (input() and print()).
    - ❖ Introduction to Python data types (integers, floats, strings).
    - ❖ Basic arithmetic operations (+, -, *, /, %, //, **).
    - ❖ Practice examples of performing arithmetic operations.
    - ❖ Assignment - Week 1: Basic Arithmetic Problems

- **Class 3: Basic Logical Operations, Boolean Operators, and Conditionals**
    - ❖ Introduction to basic logical operations (and, or, not).
    - ❖ Using boolean logical operators in conditionals.
    - ❖ Conditional statements in Python (if, elif, else).
    - ❖ Examples and practice with conditional statements.

## Week 2: Conditional Statements and Loops

- **Class 1: Basics of Iteration and Looping, For and While Loops**
  - ❖ Introduction to iteration and loops in Python.
  - ❖ Using for loops to iterate over sequences (lists, strings, etc.).
  - ❖ Using while loops for indefinite iteration.
  - ❖ Break and continue statements.
  - ❖ Practical examples with loops.
  - ❖ Assignment - Week 2: Problems with Conditionals and Loops

- **Class 2: Introduction to Strings**
  - ❖ What are strings in Python?
  - ❖ String declaration and assignment.
  - ❖ String operations (concatenation, repetition).
  - ❖ String indexing and slicing.
  - ❖ String formatting.

- **Class 3: String Methods**
  - ❖ Common string methods (e.g., split(), join(), strip()).
  - ❖ Searching and replacing substrings.
  - ❖ Changing case (upper, lower, title).
  - ❖ Checking for substring presence.
  - ❖ String interpolation with f-strings (Python 3.6+).
  - ❖ Assignment - Week 3: Strings

## Week 3:

- **Class 1: Lists in Python**
  - Introduction to lists as a data type.
  - Creating lists, list elements, and accessing elements.
  - List methods (append, insert, remove, index, etc.).
  - List operations (slicing, concatenation).
  - List comprehension.

- **Class 2: Tuples and Dictionaries**
  - Tuples as immutable sequences.
  - Dictionary basics (key-value pairs, methods).
  - Accessing and modifying dictionary elements.
  - Dictionary comprehensions.
  - Assignment - Week 4: Lists, Tuples, and Dictionaries

- **Class 3: Functions and Function Arguments**
  - Defining functions in Python.
  - Function calling and arguments.
  - Default arguments and keyword arguments.
  - Returning values from functions.
  - Variable scope and lifetime.

## Week 4

- **Class 1: Working with Files**
  - Reading and writing text files in Python.
  - Using the open() function and file modes (read, write, append).
  - Context managers (with statement) for file handling.
  - Error handling with try, except, finally.

- **Class 2: Miscellaneous Python Topics**
  - Python modules and libraries (importing, using external modules).

- Introduction to regular expressions.
- Exception handling with try, except, finally.
- Miscellaneous Python topics and best practices.
- Assignment - Week 7: Files and Miscellaneous Topics

- **Class 3: Introduction to OOP**
  - Fundamentals of Object-Oriented Programming (OOP).
  - Classes, objects, attributes, and methods.
  - Defining and using classes in Python.
  - Constructor methods (__init__) and instance variables

## Week 5

- **Class 1: Advanced OOP Concepts**
  - Inheritance and base classes.
  - Method overriding and super().
  - Encapsulation and access control.
  - Polymorphism and method overloading.
  - Assignment - Week 6: OOP

- **Class 2: Getting Started with Django**
  - What is Django?
  - Setting up a development environment.
  - Creating a new Django project.
  - Understanding the project structure.

- **Class 3: Django Models and Admin**
  - Introduction to Django models.
  - Defining models and fields.
  - Using the Django admin interface.

- Creating and managing database tables.
- Assignment - Week 1: Django Models and Admin

# Django

## Week 1: Introduction to Django

- **Class 1: Getting Started with Django**
    - What is Django?
    - Setting up a development environment.
    - Creating a new Django project.
    - Understanding the project structure.

- **Class 2: Django Models and Admin**
    - Introduction to Django models.
    - Defining models and fields.
    - Using the Django admin interface.
    - Creating and managing database tables.
    - Assignment - Week 1: Django Models and Admin

- **Class 3: Views and URL Routing**
    - Understanding Django views.
    - URL routing and URL patterns.
    - Creating views for web pages.
    - Passing data to views.

## Week 2

- **Class 1: Views and URL Routing**
    - Understanding Django views.
    - URL routing and URL patterns.
    - Creating views for web pages.

  - Passing data to views.

- **Class 2: Django Templates**
  - Introduction to Django templates.
  - Template language and syntax.
  - Rendering templates in views.
  - Template inheritance and reuse.
  - Assignment - Week 2: Views and Templates

- **Class 3: Working with Forms**
  - Introduction to Django forms.
  - Creating forms in Django.
  - Processing form data in views.
  - Form validation and error handling.

## Week 3

- **Class 1: Working with Forms**
  - Introduction to Django forms.
  - Creating forms in Django.
  - Processing form data in views.
  - Form validation and error handling.

- **Class 2: User Authentication**
  - User authentication and registration.
  - Login and logout views.
  - User authentication in Django templates.
  - Adding user-related features to your site.
  - Assignment - Week 3: Forms and User Authentication

- **Class 3: Database Relationships**
    - ❖ Defining relationships between models (e.g., ForeignKey, ManyToManyField).
    - ❖ Querying related data in views.
    - ❖ One-to-many and many-to-many relationships.

## Week 4

- **Class 1: Advanced Database Operations**
    - ❖ Aggregations and annotations.
    - ❖ Using Django's ORM for complex queries.
    - ❖ Database migrations and schema changes.
    - ❖ Database optimization and best practices.
    - ❖ Assignment - Week 4: Database Relationships

- **Class 2: Class-Based Views**
    - ❖ Introduction to class-based views (CBVs).
    - ❖ Implementing CBVs for various use cases.
    - ❖ Class-based view mixins and generic views.

- **Class 3: URL Patterns and Routing**
    - ❖ Advanced URL routing with regular expressions.
    - ❖ Handling dynamic URLs and URL parameters.

## Projects :

1. Basic data viewing site (like IMDB)
2. Creating a form creation site
3. Fully Functional Management system (or Ecommerce)

# Django Rest Framework

## Week 1

- **Class 1: Introduction to REST and DRF**
  - What is REST (Representational State Transfer)?
  - Understanding the principles of RESTful architecture.
  - Introduction to Django REST Framework (DRF).
  - Setting up a Django project for REST API development.

- **Class 2: Serializers and Views**
  - Working with DRF serializers to transform complex data types.
  - Creating model serializers for database interactions.
  - Implementing DRF views for handling API requests.
  - Configuring and routing views in DRF.
  - Assignment - Week 1: Serializers and Views

- **Class 3: Advanced DRF Concepts**
  - Pagination and filtering for large datasets.
  - Customizing response formats (JSON, XML, etc.).
  - Versioning and handling API changes.
  - Implementing API documentation with tools like DRF's built-in Browsable API.

## Week 2

- **Class 1: Advanced DRF Concepts**
  - Pagination and filtering for large datasets.
  - Customizing response formats (JSON, XML, etc.).
  - Versioning and handling API changes.
  - Implementing API documentation with tools like DRF's built-in Browsable API.

- **Class 2: Custom Authentication, Permissions, and Filtering**
  - Implementing custom authentication classes (e.g., API key-based, OAuth).
  - Customizing permission classes for fine-grained access control.
  - Creating custom filtering mechanisms for API endpoints.
  - Practical examples of custom authentication, permissions, and filtering.

- Assignment - Week 2: Custom Auth, Permissions, and Filtering

## Projects :

1. Create a basic CRUD API system for blogs.
2. Create an API system for an LMS

# SQL

## Week 1: Introduction to SQL

- **Class 1: Introduction to Databases and SQL**
  - ❖ What is a relational database?
  - ❖ Overview of SQL (Structured Query Language).
  - ❖ Setting up a local database (e.g., SQLite).
  - ❖ Basic SQL statements (SELECT, INSERT, UPDATE, DELETE).

- **Class 2: Querying Data with SQL**
  - ❖ Retrieving data from a database using SELECT statements.
  - ❖ Filtering and sorting data with WHERE and ORDER BY clauses.
  - ❖ Aggregate functions (e.g., COUNT, SUM, AVG, MAX, MIN).
  - ❖ Grouping data with GROUP BY and HAVING clauses.
  - ❖ Assignment - Week 1: SQL Fundamentals

- **Class 3: Database Design Fundamentals**
  - ❖ Understanding database design concepts.
  - ❖ Data modeling and schema design.
  - ❖ Creating tables and defining relationships (primary keys, foreign keys).

## Week 2

- **Class 1: Advanced SQL Concepts**
  - ❖ Subqueries and correlated subqueries.
  - ❖ Joins and multi-table queries.
  - ❖ Creating and modifying database schemas.
  - ❖ Views and stored procedures.
  - ❖ Assignment - Week 2: Database Design and Advanced SQL

- **Class 2: Data Manipulation and Transactions**
  - ❖ Modifying data with SQL (INSERT, UPDATE, DELETE).
  - ❖ Transactions and ACID properties (Atomicity, Consistency, Isolation, Durability).
  - ❖ Handling errors and exceptions in SQL.

- **Class 3: Indexing and Optimization**
  - ❖ Importance of indexing for query performance.
  - ❖ Creating and managing indexes.

- Optimizing SQL queries.
- Practical tips for improving database performance.
- Assignment - Week 3: Data Manipulation and Optimization

# Job Preparation

## Week 1:

- Class 1 **-** Job Preparation Introduction Lecture
- Class 2 - CV Creation
- Class 3 - How to create a Cover Letter

## Week 2:

- Class 1 - How to search job through Facebook Lecture
- Class 2 - How to search job through Linkedin
- Class 3 - Before attending the interview

## Week 3:

- Class 1 **-** How to do well in Interview and Soft Skill Based Interview Questions
- Class 2 - Types of Technical Based Interview