Summer Training Schedule 45 Days

45-Day Summer Training Syllabus for BTech Students

Days 1–15: Web Technologies (HTML, CSS, Bootstrap, JavaScript)

Day 1: Introduction to Web Development

- Web basics: client-server model, browsers, HTTP
- HTML: structure, tags (head, body, div, p, headings)
- Task: Create a simple static webpage (personal bio)
- **Duration**: 4 hours
- **Day 2: HTML Advanced**
- HTML forms, tables, semantic tags (header, footer, article)
- Task: Build a registration form with validation
- **Duration**: 4 hours

Day 3: CSS Basics

- CSS syntax, selectors, properties (color, font, margin)
- Box model, positioning (relative, absolute)
- Task: Style the Day 2 form
- **Duration**: 4 hours

Day 4: CSS Layouts

- Flexbox, CSS Grid
- Responsive design basics
- Task: Create a responsive layout for a blog page
- **Duration**: 4 hours

Day 5: Bootstrap Introduction

- Bootstrap setup, grid system, containers
- Typography, buttons, navbar
- Task: Convert Day 4 blog layout to Bootstrap

- **Duration**: 4 hours

Day 6: Bootstrap Components

- Cards, modals, carousel
- Task: Add a carousel and cards to the blog page
- **Duration**: 4 hours

Day 7: JavaScript Basics

- JS syntax, variables, data types, operators
- DOM manipulation (getElementById, querySelector)
- Task: Add interactive buttons to the blog page
- **Duration**: 4 hours

Day 8: JavaScript Events

- Event listeners (click, input, mouseover)
- Task: Create a form validation script
- **Duration**: 4 hours

Day 9: JavaScript Functions and Arrays

- Functions, arrow functions, arrays (map, filter)
- Task: Build a to-do list with add/remove functionality
- **Duration**: 4 hours

Day 10: JavaScript DOM and Debugging

- Advanced DOM manipulation, debugging with console
- Task: Enhance to-do list with edit and delete buttons
- **Duration**: 4 hours

Day 11: JavaScript ES6 Features

- Let/const, template literals, destructuring
- Task: Refactor to-do list using ES6 features
- **Duration**: 4 hours

Day 12: Introduction to APIs

- Fetch API, JSON handling

- Task: Fetch and display data from a public API (e.g., weather API)
- **Duration**: 4 hours

Days 13–14: Mini-Project: IIIM Ayodhya Website

- Plan and design website for IIIM Ayodhya (home, about, courses, contact)
- Use HTML, CSS, Bootstrap, JS
- Task: Build static pages with navigation and contact form
- **Duration**: 8 hours/day (16 hours total)

Day 15: Mini-Project Completion and Review

- Add interactivity (e.g., form validation, dynamic content)
- Test responsiveness, present, and debug
- Task: Finalize and deploy IIIM Ayodhya website
- **Duration**: 4 hours

Days 16–22: Core Python (7 Days)

Day 16: Introduction and Basics

- Python setup, IDEs (VSCode, Jupyter)
- Variables, data types, operators
- Task: Write 5 programs (e.g., calculator, string manipulation)
- **Duration**: 4 hours

Day 17: Control Structures

- If-elif-else, loops (for, while)
- List, tuple basics
- Task: Programs for pattern printing, sum of numbers
- **Duration**: 4 hours

Day 18: Functions and Modules

- Functions, arguments (*args, **kwargs)
- Modules (math, random)
- Task: Write functions for factorial, prime check
- **Duration**: 4 hours

Day 19: Data Structures

- Lists, tuples, sets, dictionaries
- List comprehensions
- Task: Programs for list sorting, dictionary operations
- **Duration**: 4 hours

Day 20: File and Exception Handling

- File read/write, CSV handling
- Try-except, custom exceptions
- Task: Create a program to read/write student data to CSV
- **Duration**: 4 hours

Day 21: Object-Oriented Programming

- Classes, objects, inheritance
- Task: Build a class-based program (e.g., bank account)
- **Duration**: 4 hours

Day 22: Advanced Python and Mini-Project

- Regular expressions, requests library (API calls)
- Task: Build a console-based quiz app or weather app
- **Duration**: 4 hours

Days 23–29: Django Framework (7 Days)

Day 23: Django Introduction and Setup

- Django installation, project/app structure
- Create a simple project
- Task: Set up a Django project and run a basic app
- **Duration**: 4 hours

Day 24: Django Models

- Models, migrations, SQLite database
- Task: Create a model for a blog (Post)
- **Duration**: 4 hours

Day 25: Django Views and URLs

- Function-based views, URL routing
- Task: Display a list of blog posts
- **Duration**: 4 hours

Day 26: Django Templates

- Template syntax, rendering dynamic data
- Task: Create templates for blog post list and detail
- **Duration**: 4 hours

Day 27: CRUD Operations

- Create, read, update, delete operations
- Task: Implement CRUD for blog posts
- **Duration**: 4 hours

Day 28: Django Admin and Static Files

- Django admin setup, static files (CSS, JS)
- Task: Customize admin panel
- **Duration**: 4 hours

Day 29: Mini-Project Review

- Enhance blog app (add search or categories)
- Task: Test and debug the blog app
- **Duration**: 4 hours

Days 30–45: Full-Fledged Web Application Project

Days 30–32: Project Planning and Setup

- Define project (e.g., e-commerce, library system)
- Set up Django project, design database models
- Task: Create models and wireframe frontend
- **Duration**: 12 hours (4h/day)

Days 33–36: Backend Development

- Implement CRUD operations, views, URLs

- Task: Build backend logic for core features
- **Duration**: 16 hours (4h/day)

Days 37–40: Frontend Development

- Use Bootstrap, JS for responsive UI
- Integrate with Django templates
- Task: Design and connect frontend pages
- **Duration**: 16 hours (4h/day)

Days 41–43: Integration and Testing

- Connect frontend and backend
- Test functionality, fix bugs
- Task: Ensure all features work seamlessly
- **Duration**: 12 hours (4h/day)

Days 44–45: Deployment and Presentation

- Deploy on a local server (or cloud if feasible)
- Prepare documentation, present project
- Task: Finalize and showcase the web application
- **Duration**: 8 hours (4h/day)