

LIVE COURSE

Python AI Full Stack Development

Learn Python, web development, databases and AI from scratch in live mentor-led sessions. Build 9 real projects and become job-ready as a full stack developer with AI skills.

Industry-Led Course

Batch details updating soon

Technical



DURATION
6 Months



LEVEL
Beginner to Advanced



MODE
Live Online



LANGUAGE
English



BATCH STARTS
To be announced



ENROLMENT CLOSES
Open

COURSE FEE

₹7,999

₹30,000 - save 73% OFF



LIVE COURSE BROCHURE

Inside this brochure

Everything on the Python AI Full Stack Development page, in one document. Tap a section to jump to it.

01		Course Overview	03
02		How Our Courses Work	04
03		Eligibility & Prerequisites	05
04		Skills You Will Gain	05
05		How we designed our Product?	06
06		What You Will Learn	08
07		Meet Our SME Board Members	47
08		Key Tools and Techniques	49
09		Projects You Will Work On	50
10		Python AI Full Stack Development	52
11		World's First Ed-Tech with 8C Appraisal	53
12		Frequently Asked Questions	54
13		Ready to take the next step?	55

01 · AT A GLANCE



Course Overview

Learn Python, web development, databases and AI from scratch in live mentor-led sessions. Build 9 real projects and become job-ready as a full stack developer with AI skills.

This live, mentor-led course takes you from Python basics to building full stack web applications powered by AI. You will work with Django, REST APIs, databases, React and machine learning models, completing hands-on projects and graduating with a deployable, job-ready portfolio.



DURATION
6 Months



LEVEL
Beginner to Advanced



MODE
Live Online



LANGUAGE
English



BATCH STARTS
To be announced



ENROLMENT CLOSES
Open



LEARNERS
Trusted by ambitious learners nationwide

COURSE FEE

₹7,999

₹30,000 - save 73% OFF

Highlights

Industry-Led Upskilling Courses

Learn from industry professionals with years of real-world experience.

Personalized Mentorship

Get one-on-one guidance tailored to your career goals and aspirations.

Premium Resources

Access course materials, job portal, free events, challenges and learner 8C Appraisal reports.

Vibrant Community

Connect with like-minded peers and build lasting professional relationships.



02 · SIMPLE 3-STEP JOURNEY

How Our Courses Work

A structured, step-by-step path designed to take you from foundational learning to career success with industry guidance.

01

Step 1: Enroll & Get Started

Choose your course and complete the enrollment process. Access your learning materials instantly and join our community of learners.

✓ Instant access to course materials

✓ Personalized learning dashboard

✓ Welcome orientation session

02

Step 2: Learn & Practice

Engage with interactive lessons, live sessions, and hands-on projects. Learn from industry experts and apply your knowledge in real-world scenarios.

✓ Live classes with expert instructors

✓ Hands-on practical projects

✓ One-on-one mentorship support

03

Step 3: Complete & Get Certified

Finish your coursework, pass assessments, and receive your professional certificate. Launch your new career with confidence and credentials.

✓ Industry-recognized certificate

✓ Portfolio of completed projects

✓ Career placement assistance

03 · BEFORE YOU ENROLL



Eligibility & Prerequisites

Check who this course is designed for and what you should know before your first class.

WHO CAN JOIN**Eligibility**

- ✓ Students from any degree or diploma stream
- ✓ Freshers looking for their first software job
- ✓ Working professionals planning a career switch to tech
- ✓ Anyone passionate about coding and AI

BEFORE YOU START**Prerequisites**

- ✓ A laptop or desktop with internet access
- ✓ Basic computer and typing skills
- ✓ Comfort with high school level maths
- ✓ Willingness to practice coding daily
- ✓ No prior programming experience required

04 · CAREER-READY SKILLS



Skills You Will Gain

Hands-on skills you will practise in live classes, assignments and projects throughout the course.

Python

Django

Flask

REST APIs

React

JavaScript

HTML and CSS

SQL

PostgreSQL

Git and GitHub

Machine Learning

Scikit-learn

OpenAI API

Docker

Cloud Deployment

05 · PRODUCT ECOSYSTEM ARCHITECTURE



How we designed our Product?

An all-in-one ed-tech ecosystem carefully engineered to bridge learning, practical skill assessment, career guidance, and employment.

WORLD'S FIRST FEATURE

8C Appraisal Report

Proprietary evaluation matrix measuring holistic performance across 8 career pillars for authentic proof-of-work.

Communication

Configuration

Chronicle

Completeness

Cognizance

Conception

Comprehension

Case Study

Industry Validated · 100% Data Score

SUBJECT MATTER EXPERT AI

Insyrio AI

Get instant 24/7 doubt solving and concept clarifications directly from our specialized AI trained on your enrolled courses.

- ✓ 24/7 Course-Specific SME Assistant
- ✓ Instant Step-by-Step Doubt Resolution

Instant Learning Support

GAMIFIED REWARDS

iSure Coins

Use iSure Coins for AI career tools, masterclasses and CertiQ - and part-pay your course fee with them.

- ✓ Earn Perks & Rewards · Pay with Coins

Redeem Rewards Store

AI ENGINE

AI Career Diagnosis

Algorithmic analysis mapping strengths, gaps, and tailored pathways for industry roles.

- ✓ Skill Gap Mapping

Personalized Roadmap

CURRICULUM

Industry-Led Courses

Handcrafted learning modules built directly alongside senior engineering, design, and management leads.

Practical Frameworks

LIVE

Live Classes

Interactive classrooms with live chat, code walk-throughs, and real-time engagement.

Real-time Q&A Support

AUTO-SCAN

AI Resume ATS

Instant ATS optimization scanner ensuring your resume passes screening software effortlessly.

High Parser Score

PRACTICAL

Assignments

Industry-grade project briefs with detailed feedback and grading from experienced mentors.

Code & Work Reviews

WEEKLY

Challenges

Weekly hackathons, coding sprints, and design contests to test your problem-solving skills.

Leaderboard Ranking

NETWORKING

Events

Exclusive expert keynotes, career fairs, and interactive networking workshops.

Free Community Pass

1-ON-1

Live Mentorship

Direct one-on-one video guidance sessions with experienced industry practitioners.

Personalized Growth

TRACKING

Attendance

Class-by-class attendance tracking to ensure disciplined, consistent progression.

Attendance Reports

HIRING

Jobs

Direct hiring pipeline connecting top-performing learners directly with recruiter managers.

Exclusive Placements

FREE ACCESS

Free Books

Curated digital library filled with essential tech, management, and soft-skill books.

Instant Digital Library

ARTICLES

Blog

In-depth technical tutorials, industry trend analysis, and career growth strategies.

Updated Weekly

FREE ACCESS

Free Study Material

Downloadable cheat sheets, interview preparation guides, templates, and revision notes.

One-Click Downloads

06 · COURSE CURRICULUM



What You Will Learn

A carefully sequenced curriculum - every section breaks into modules and hands-on topics taught live by industry experts.

8

SECTIONS

48

MODULES

288

TOPICS

9

PROJECTS

01

SECTION 1

Python Programming Foundations and Problem Solving Basics

Build a strong base in Python syntax, data structures, functions, and logical problem solving that prepares learners for every later stage of full stack AI development.

6 modules · 36 topics

M1.1 **Getting Started with Python and Development Setup** · 6 topics

Learners set up a professional Python environment and write their first programs with confidence.

1 **Introduction to Python and Its Role in AI Development**

Understand why Python dominates AI and full stack work and see the course roadmap. Explore real products built with Python.

PYTHON OVERVIEW

AI LANDSCAPE

CAREER ROADMAP

2 **Installing Python and Configuring VS Code**

Install Python, set up VS Code, and run scripts from the editor and terminal. Verify the setup with a first program.

INSTALLATION

VS CODE

TERMINAL

3 **Virtual Environments and Package Management with pip**

Create isolated environments and install packages with pip. Learn how to manage dependencies with requirements files.

VENV

PIP

DEPENDENCIES

4 **Writing and Running Your First Python Programs**

Write scripts using print, comments, and simple input. Learn how Python executes code line by line.

SCRIPTS

INPUT AND OUTPUT

COMMENTS

5 Using Jupyter Notebooks and Google Colab

Work with notebooks for experiments and quick prototyping. Learn cells, markdown, and sharing notebooks.

JUPYTER

COLAB

PROTOTYPING

6 Python Style, PEP 8, and Debugging Basics

Write clean readable code following PEP 8 conventions. Read error messages and fix common beginner mistakes.

PEP 8

DEBUGGING

READABILITY

M1.2 Core Syntax: Variables, Data Types, and Operators · 6 topics

Learners master the building blocks used in every Python program.

1 Variables, Naming Rules, and Dynamic Typing

Create variables and understand how Python handles types dynamically. Apply naming conventions that keep code clear.

VARIABLES

DYNAMIC TYPING

NAMING

2 Numbers and Arithmetic Operations

Work with integers, floats, and math operators. Handle precision issues and use the math module.

INTEGERS

FLOATS

MATH MODULE

3 Strings and Text Manipulation

Use indexing, slicing, and string methods to process text. Practice cleaning and transforming real text data.

STRINGS

SLICING

METHODS

4 Formatted Output with f-Strings

Format output cleanly using f-strings and format specifiers. Build readable reports from variable values.

F-STRINGS

FORMATTING

OUTPUT

5 Booleans, Comparison, and Logical Operators

Evaluate conditions using comparison and logical operators. Understand truthy and falsy values in Python.

BOOLEANS

COMPARISON

LOGIC

6 Type Conversion and User Input Handling

Convert between types and process user input safely. Build small interactive programs such as calculators.

TYPE CASTING

USER INPUT

VALIDATION

M1.3 Control Flow: Decisions and Loops · 6 topics

Learners control program behavior using conditions and repetition.

1 Conditional Statements with if, elif, and else

Write branching logic to make programs respond to different inputs. Practice with grading and eligibility examples.

CONDITIONALS

BRANCHING

DECISION LOGIC

2 Nested Conditions and Complex Decision Making

Combine and nest conditions to model real-world rules. Simplify complex logic for readability.

NESTED IF

BOOLEAN LOGIC

REFACTORING

3 for Loops and the range Function

Iterate over sequences and number ranges using for loops. Solve counting and summation problems.

FOR LOOP

RANGE

ITERATION

4 while Loops and Loop Control Statements

Use while loops with break, continue, and pass. Avoid infinite loops and design clear exit conditions.

WHILE LOOP

BREAK AND CONTINUE

LOOP DESIGN

5 Nested Loops and Pattern Printing

Use loops inside loops to build patterns and tables. Strengthen logical thinking through visual output.

NESTED LOOPS

PATTERNS

LOGIC BUILDING

6 Mini Project: Number Guessing and Quiz Game

Combine conditions, loops, and input into a small interactive game. Practice planning and testing a complete program.

MINI PROJECT

INTEGRATION

TESTING

M1.4 Core Data Structures in Python · 6 topics

Learners store, organize, and process data using built-in collections.

1 Lists and List Operations

Create and modify lists using indexing, slicing, and methods. Apply lists to everyday data tasks.

LISTS

INDEXING

METHODS

2 List Comprehensions and Sorting Techniques

Write concise list comprehensions and sort data with custom keys. Compare them with traditional loops.

COMPREHENSIONS

SORTING

KEY FUNCTIONS

3

Tuples and Immutability

Understand tuples, packing, and unpacking. Learn when immutable data is the better choice.

TUPLES

UNPACKING

IMMUTABILITY

4

Dictionaries and Key-Value Data Modeling

Store and retrieve data with dictionaries and their methods. Model records such as students and products.

DICTIONARIES

KEY-VALUE

DATA MODELING

5

Sets and Membership Operations

Use sets to remove duplicates and perform union and intersection. Understand performance benefits of fast lookups.

SETS

SET OPERATIONS

LOOKUP

6

Choosing and Combining Data Structures

Work with nested structures such as lists of dictionaries. Decide which structure suits each problem.

NESTED DATA

SELECTION

DATA HANDLING

M1.5

Functions, Modules, and Error Handling · 6 topics

Learners write reusable, organized, and reliable code.

1

Defining Functions with Parameters and Return Values

Create functions that accept inputs and return results. Understand why functions make code reusable.

FUNCTIONS

PARAMETERS

RETURN

2

Default, Keyword, and Variable-Length Arguments

Use default values, keyword arguments, args, and kwargs. Design flexible function signatures.

DEFAULT ARGS

ARGS AND KWARGS

FLEXIBILITY

3

Scope, Lambda Functions, and Built-in Helpers

Understand local and global scope and write lambda expressions. Use map, filter, and zip effectively.

SCOPE

LAMBDA

MAP AND FILTER

4

Modules, Packages, and the Standard Library

Import built-in modules and create your own. Explore useful libraries like random, datetime, and collections.

MODULES

IMPORTS

STANDARD LIBRARY

5

Exception Handling with try, except, and finally

Catch and handle runtime errors gracefully. Raise custom exceptions to enforce correct usage.

EXCEPTIONS

TRY EXCEPT

ERROR HANDLING

6 File Handling with Text, CSV, and JSON Files

Read and write files using context managers. Process CSV and JSON data for practical tasks.

FILE I/O

CSV

JSON

M1.6 Problem Solving and Logic Building Practice · 6 topics

Learners apply Python fundamentals to structured problem solving and coding challenges.

1 Problem Solving Framework and Algorithmic Thinking

Break problems into steps using pseudocode and flowcharts. Learn to plan before writing code.

ALGORITHMS

PSEUDOCODE

DECOMPOSITION

2 Solving Problems with Strings and Lists

Practice common interview-style problems such as palindromes, reversals, and frequency counts. Compare multiple solutions.

STRING PROBLEMS

LIST PROBLEMS

PRACTICE

3 Recursion Fundamentals and Practical Examples

Understand base cases and recursive calls through factorial and Fibonacci. Compare recursion with iteration.

RECURSION

BASE CASE

ITERATION

4 Searching and Sorting Algorithms Basics

Implement linear and binary search along with bubble and selection sort. Discuss efficiency at an introductory level.

SEARCHING

SORTING

BIG O BASICS

5 Testing and Debugging with assert and unittest

Write simple tests to verify function behavior. Use debugging tools to trace and fix logic errors.

TESTING

UNITTEST

DEBUGGING

6 Capstone Mini Project: Expense Tracker Application

Build a command line expense tracker using functions, data structures, and file storage. Present and review the solution as a team.

CAPSTONE

PROJECT BUILDING

CODE REVIEW

02

SECTION 2

Data Structures, OOP and Version Control with Git

Learners master core Python data structures, object-oriented design and Git-based collaboration to write organized, reusable and team-ready code.

6 modules · 36 topics

M2.1 Built-in Data Structures in Depth · 6 topics

Learners work confidently with Python lists, tuples, sets and dictionaries to store and manipulate data efficiently.

1 Lists and List Operations

Create, index, slice and modify lists using common methods. Practice real-world tasks such as cleaning and sorting records.

LISTS

SLICING

SORTING

2 Tuples and Immutability

Understand when to use tuples instead of lists and how unpacking works. Explore immutability and its benefits in safe data handling.

TUPLES

UNPACKING

IMMUTABILITY

3 Sets and Set Operations

Use sets to remove duplicates and perform union, intersection and difference. Apply them to fast membership testing.

SETS

DEDUPLICATION

MEMBERSHIP

4 Dictionaries and Key-Value Modeling

Build and update dictionaries to represent structured data. Learn safe access patterns using get and default values.

DICTIONARIES

KEY-VALUE

DATA MODELING

5 Nested Data Structures and JSON-like Data

Work with lists of dictionaries and dictionaries of lists. Read and reshape nested data similar to API responses.

NESTED DATA

JSON

PARSING

6 Comprehensions and Choosing the Right Structure

Write list, set and dictionary comprehensions for concise code. Compare structures by speed and use case to make smart choices.

COMPREHENSIONS

EFFICIENCY

DESIGN CHOICES

M2.2 Algorithmic Thinking with Core Structures · 6 topics

Learners implement classic structures and algorithms to build strong problem-solving and complexity awareness.

1 Big O Notation and Complexity Basics

Learn to estimate how code scales with input size. Compare common time complexities using simple examples.

BIG O **COMPLEXITY** **ANALYSIS**

2 Stacks and Queues

Implement stacks and queues using lists and deque. Solve problems such as bracket matching and task scheduling.

STACK **QUEUE** **DEQUE**

3 Linked Lists

Build singly linked lists from scratch with insert, delete and traverse operations. Understand how they differ from Python lists.

LINKED LIST **POINTERS** **TRAVERSAL**

4 Searching Algorithms

Implement linear and binary search and compare their performance. Learn when sorted data makes searching faster.

BINARY SEARCH **LINEAR SEARCH** **OPTIMIZATION**

5 Sorting Algorithms

Code bubble, selection and merge sort and observe their trade-offs. Learn how Python's built-in sort compares.

SORTING **MERGE SORT** **TRADE-OFFS**

6 Hash Maps, Trees and Recursion Practice

Explore how hash tables power dictionaries and see basic binary trees in action. Use recursion to traverse and solve problems.

HASHING **TREES** **RECURSION**

M2.3 Object-Oriented Programming Fundamentals · 6 topics

Learners model real-world entities with classes and objects to write structured and maintainable programs.

1 Classes and Objects

Define classes and create objects with attributes and methods. Understand the idea of modeling real-world entities in code.

CLASSES **OBJECTS** **METHODS**

2 Constructors and Instance Attributes

Use the init method to set up object state. Distinguish instance attributes from class attributes.

CONSTRUCTOR

ATTRIBUTES

STATE

3 Encapsulation and Properties

Protect internal data using naming conventions and property decorators. Add validation to attribute access.

ENCAPSULATION

PROPERTIES

VALIDATION

4 Inheritance and Method Overriding

Create subclasses that reuse and extend parent behavior. Use super to build clean class hierarchies.

INHERITANCE

OVERRIDING

SUPER

5 Polymorphism and Abstract Classes

Write code that works across different object types through shared interfaces. Define abstract base classes to enforce structure.

POLYMORPHISM

ABSTRACTION

ABC

6 Magic Methods and Operator Overloading

Customize object behavior with methods like str, repr, eq and add. Make classes feel natural and Pythonic.

DUNDER METHODS

OPERATORS

PYTHONIC CODE

M2.4 Advanced OOP and Clean Code Practices · 6 topics

Learners apply design principles and professional coding habits to build robust and reusable class-based programs.

1 Composition versus Inheritance

Compare building objects from smaller parts with extending parent classes. Decide which approach fits each design problem.

COMPOSITION

DESIGN

REUSABILITY

2 Class Methods and Static Methods

Use classmethod and staticmethod decorators for alternative constructors and utilities. Learn when each is appropriate.

CLASSMETHOD

STATICMETHOD

DECORATORS

3 Dataclasses and Modern Python Models

Reduce boilerplate with dataclasses and type hints. Build clean data-holding classes quickly.

DATACLASSES

TYPE HINTS

MODELS

4

SOLID Principles in Python

Learn the five SOLID principles through short Python examples. Refactor a messy class into a cleaner design.

SOLID

REFACTORING

DESIGN PRINCIPLES

5

Exception Handling and Custom Exceptions

Handle errors with try, except and finally blocks. Define custom exceptions to make programs more reliable.

EXCEPTIONS

ERROR HANDLING

RELIABILITY

6

Modules, Packages and Code Organization

Split code into modules and packages with clear imports. Follow PEP 8 and docstring conventions for readable projects.

PACKAGES

PEP 8

DOCSTRINGS

M2.5

Git Fundamentals for Everyday Development · 6 topics

Learners track, save and review code changes using Git commands in a local repository.

1

Introduction to Version Control and Git Setup

Understand why version control matters and install and configure Git. Create a first repository and explore its structure.

GIT

SETUP

REPOSITORIES

2

Staging, Committing and Viewing History

Use add, commit, status and log to record project changes. Write clear commit messages that explain the work.

COMMIT

STAGING

HISTORY

3

Branching and Switching Workflows

Create and switch branches to develop features safely. Understand how branches isolate work from the main line.

BRANCHES

CHECKOUT

ISOLATION

4

Merging and Resolving Conflicts

Merge branches and fix conflicts step by step. Build confidence handling overlapping changes.

MERGE

CONFLICTS

PROBLEM SOLVING

5

Undoing Changes Safely

Use restore, reset, revert and stash to correct mistakes. Learn which command is safest in each situation.

RESET

REVERT

STASH

6

Gitignore, Tags and Project Hygiene

Exclude unwanted files with gitignore and mark releases with tags. Keep repositories clean and professional.

GITIGNORE

TAGS

BEST PRACTICES

M2.6 Collaboration with GitHub and Team Workflows · 6 topics

Learners collaborate on shared codebases using GitHub and apply all section skills in a team-style capstone.

1 GitHub Basics and Remote Repositories

Create a GitHub account and connect local repositories to remotes. Practice push, pull and clone operations.

GITHUB

REMOTES

PUSH AND PULL

2 SSH Keys and Secure Authentication

Set up SSH keys or tokens for safe access to repositories. Understand basic security habits for developers.

SSH

AUTHENTICATION

SECURITY

3 Forks, Pull Requests and Code Reviews

Contribute to a project through forks and pull requests. Give and receive constructive review feedback.

PULL REQUESTS

CODE REVIEW

COLLABORATION

4 Team Branching Strategies and Rebase

Compare feature branching and trunk-based workflows. Use rebase to keep history clean and linear.

WORKFLOWS

REBASE

TEAM PRACTICE

5 Issues, Project Boards and README Writing

Track tasks with GitHub issues and boards. Write a strong README and portfolio-ready project documentation.

ISSUES

README

DOCUMENTATION

6 Capstone: Team OOP Project on GitHub

Build a small object-oriented application in teams using branches and pull requests. Present the final repository and code design.

CAPSTONE

TEAMWORK

PORTFOLIO

03

SECTION 3

Frontend Development with HTML, CSS and JavaScript

Learners build responsive, interactive and accessible web interfaces from scratch, progressing from semantic markup to modern JavaScript and API-driven pages.

6 modules · 36 topics

M3.1 HTML Fundamentals and Semantic Page Structure · 6 topics

Learners master the building blocks of web pages using clean, semantic and accessible HTML.

1 How the Web Works and Setting Up the Dev Environment

Understand browsers, servers, HTTP and the request-response cycle. Set up VS Code, Live Server and browser developer tools.

[HTTP BASICS](#)[VS CODE](#)[DEVTOOLS](#)

2 HTML Document Structure and Core Elements

Write a complete HTML5 document using headings, paragraphs, links, images and lists. Learn how the browser parses and renders markup.

[HTML5](#)[ELEMENTS](#)[DOCUMENT STRUCTURE](#)

3 Semantic HTML and Page Layout Tags

Use header, nav, main, section, article and footer to give pages meaning. Understand why semantics matter for SEO and accessibility.

[SEMANTIC TAGS](#)[SEO](#)[ACCESSIBILITY](#)

4 Tables, Media and Embedded Content

Present data with tables and embed audio, video and iframes. Optimize images and media for the web.

[TABLES](#)[MULTIMEDIA](#)[EMBEDDING](#)

5 Forms and Input Controls

Build forms with text fields, selects, checkboxes, radio buttons and buttons. Apply built-in HTML validation attributes.

[FORMS](#)[INPUT TYPES](#)[VALIDATION](#)

6 Accessibility and HTML Best Practices

Apply ARIA roles, alt text and keyboard navigation principles. Validate markup and follow clean coding conventions.

[ARIA](#)[WCAG](#)[CODE QUALITY](#)

M3.2 CSS Styling, Selectors and the Box Model · 6 topics

Learners style pages confidently by understanding how CSS selects, cascades and renders elements.

1 CSS Syntax and Ways to Add Styles

Apply inline, internal and external CSS and understand rule structure. Learn how to organize stylesheets in a project.

[CSS SYNTAX](#)
[STYLESHEETS](#)
[PROJECT STRUCTURE](#)

2 Selectors, Specificity and the Cascade

Target elements with class, ID, attribute and combinator selectors. Resolve style conflicts using specificity and inheritance.

[SELECTORS](#)
[SPECIFICITY](#)
[CASCADE](#)

3 The Box Model and Element Sizing

Work with margin, padding, border and box-sizing. Debug spacing issues using browser developer tools.

[BOX MODEL](#)
[SPACING](#)
[DEBUGGING](#)

4 Typography, Colors and Backgrounds

Style text with web fonts and apply color systems, gradients and background images. Build readable and consistent visual design.

[TYPOGRAPHY](#)
[COLOR THEORY](#)
[GRADIENTS](#)

5 Pseudo Classes, Transitions and Animations

Add hover and focus states and smooth transitions. Create keyframe animations that enhance user experience.

[PSEUDO CLASSES](#)
[TRANSITIONS](#)
[KEYFRAMES](#)

6 CSS Variables and Scalable Styling Practices

Use custom properties and naming methods like BEM to keep styles maintainable. Build a small reusable style guide.

[CSS VARIABLES](#)
[BEM](#)
[MAINTAINABILITY](#)

M3.3 Responsive Layouts with Flexbox, Grid and Bootstrap · 6 topics

Learners design layouts that adapt smoothly to any screen size using modern CSS tools.

1 Responsive Design Principles and Media Queries

Apply the mobile-first approach with viewport settings and breakpoints. Test designs across device sizes.

[MOBILE FIRST](#)
[MEDIA QUERIES](#)
[VIEWPORT](#)

2 Flexbox Layout in Depth

Align and distribute items along main and cross axes. Build navbars, cards and centered layouts with Flexbox.

[FLEXBOX](#)
[ALIGNMENT](#)
[NAVBARs](#)

3 CSS Grid for Two Dimensional Layouts

Define rows, columns and grid areas to create complex page layouts. Combine Grid with Flexbox where each fits best.

CSS GRID

GRID AREAS

PAGE LAYOUT

4 Positioning, Z-Index and Overflow

Use relative, absolute, fixed and sticky positioning. Control stacking order and overflow for overlays and sticky headers.

POSITIONING

Z-INDEX

OVERFLOW

5 Bootstrap Framework Essentials

Use the Bootstrap grid, components and utility classes to build pages quickly. Customize the framework to match a brand.

BOOTSTRAP

COMPONENTS

UTILITIES

6 Tailwind CSS and Utility First Styling

Style interfaces with utility classes directly in markup. Compare utility-first and component-based approaches.

TAILWIND CSS

UTILITY CLASSES

RAPID PROTOTYPING

M3.4 JavaScript Fundamentals for Web Developers · 6 topics

Learners learn core JavaScript concepts and relate them to the Python knowledge they already have.

1 JavaScript Basics, Variables and Data Types

Declare variables with let and const and work with strings, numbers and booleans. Compare JavaScript syntax with Python.

VARIABLES

DATA TYPES

SYNTAX

2 Operators, Conditions and Loops

Write decision logic and repeat tasks using if, switch, for and while. Understand type coercion and strict equality.

CONTROL FLOW

LOOPS

OPERATORS

3 Functions, Scope and Arrow Functions

Create reusable functions and understand local and global scope. Use arrow functions and default parameters.

FUNCTIONS

SCOPE

ARROW FUNCTIONS

4 Arrays and Array Methods

Store collections and process them with map, filter, reduce and forEach. Apply spread and destructuring syntax.

ARRAYS

HIGHER ORDER FUNCTIONS

DESTRUCTURING

5 Objects, JSON and Classes

Model data with objects and classes and convert it to and from JSON. Understand the this keyword and prototypes at a basic level.

[OBJECTS](#)
[JSON](#)
[CLASSES](#)

6 Error Handling and Debugging in the Browser

Use try catch, console tools and breakpoints to find and fix bugs. Read common error messages with confidence.

[ERROR HANDLING](#)
[DEBUGGING](#)
[CONSOLE](#)

M3.5 DOM Manipulation and Interactive User Interfaces · 6 topics

Learners connect JavaScript to the page to create dynamic and interactive experiences.

1 Understanding the DOM and Selecting Elements

Explore the DOM tree and select nodes with querySelector and related methods. Read and change content and attributes.

[DOM](#)
[QUERYSELECTOR](#)
[NODE TRAVERSAL](#)

2 Modifying Elements and Styles Dynamically

Create, insert and remove elements and toggle classes. Update styles in response to user actions.

[DOM UPDATES](#)
[CLASSLIST](#)
[DYNAMIC STYLING](#)

3 Event Handling and Event Delegation

Respond to clicks, key presses and form events. Use event bubbling and delegation for efficient handlers.

[EVENTS](#)
[BUBBLING](#)
[DELEGATION](#)

4 Form Handling and Client Side Validation

Read form values, validate input with JavaScript and show helpful error messages. Prevent default submission behavior.

[FORM HANDLING](#)
[VALIDATION](#)
[UX FEEDBACK](#)

5 Browser Storage with localStorage and sessionStorage

Persist user data and preferences in the browser. Build features like theme switchers and saved to-do lists.

[LOCALSTORAGE](#)
[SESSIONSTORAGE](#)
[PERSISTENCE](#)

6 Mini Project: Interactive To-Do and Quiz App

Combine DOM, events and storage to build a polished interactive app. Practice structuring code into clear functions.

[PROJECT BUILDING](#)
[DOM EVENTS](#)
[CODE STRUCTURE](#)

M3.6 Asynchronous JavaScript, APIs and Modern Tooling · 6 topics

Learners fetch live data, use modern JavaScript features and prepare a frontend project for deployment.

1 Asynchronous JavaScript and the Event Loop

Understand callbacks, timers and how the event loop schedules work. Recognize problems like callback nesting.

ASYNC

EVENT LOOP

CALLBACKS

2 Promises and Async Await

Handle asynchronous tasks with promises and async await syntax. Manage errors and run tasks in parallel.

PROMISES

ASYNC AWAIT

ERROR HANDLING

3 Fetch API and Working with REST APIs

Send GET and POST requests and render JSON responses on the page. Handle loading and error states gracefully.

FETCH API

REST

JSON

4 ES6 Modules and Modern JavaScript Features

Split code into modules using import and export. Apply optional chaining, template literals and other modern syntax.

ES MODULES

MODERN JS

CODE ORGANIZATION

5 Package Management and Build Tools with npm and Vite

Install packages with npm and bundle a project using Vite. Understand development servers and production builds.

NPM

VITE

BUNDLING

6 Capstone: Responsive Weather and News Dashboard

Build a responsive dashboard that consumes public APIs and deploy it using GitHub Pages. Prepare the frontend for later backend integration.

API INTEGRATION

DEPLOYMENT

RESPONSIVE UI

04

SECTION 4

Building Interactive User Interfaces with React

Learners build dynamic single page applications by mastering React components, state, hooks, routing, forms and API integration.

6 modules · 36 topics

M4.1 React Fundamentals and Project Setup · 6 topics

Learners set up a modern React workspace and understand the core ideas behind component based interfaces.

1 Introduction to React and the Virtual DOM

Understand why React exists and how the virtual DOM updates the screen efficiently. Compare it with plain JavaScript DOM manipulation.

[REACT BASICS](#)[VIRTUAL DOM](#)[RENDERING](#)

2 Setting Up a React Project with Vite

Create a new React project using Vite and Node tooling. Explore the folder structure and run the development server.

[VITE](#)[NODE AND NPM](#)[PROJECT SETUP](#)

3 Writing JSX and Embedding Expressions

Learn JSX syntax rules and how to embed JavaScript expressions in markup. Practice conditional output and list output in JSX.

[JSX](#)[EXPRESSIONS](#)[SYNTAX](#)

4 Creating and Composing Functional Components

Build reusable functional components and combine them into larger layouts. Learn naming conventions and file organization.

[COMPONENTS](#)[COMPOSITION](#)[REUSABILITY](#)

5 Passing Data with Props

Send data from parent to child components using props. Apply default values and destructuring for cleaner code.

[PROPS](#)[DESTRUCTURING](#)[DATA FLOW](#)

6 Rendering Lists and Conditional UI

Display collections using map and keys and show or hide elements based on conditions. Build a product list interface.

[LISTS](#)[KEYS](#)[CONDITIONAL RENDERING](#)

M4.2 State Management and Event Handling · 6 topics

Learners make interfaces respond to user actions by managing local component state and events.

1 Understanding State with the useState Hook

Store changing values in components using useState. See how state updates trigger re-rendering.

USESTATE

STATE

RE-RENDERING

2 Handling Click and Input Events

Attach event handlers to buttons and inputs and read event data. Build a counter and a live text preview.

EVENTS

HANDLERS

INTERACTIVITY

3 Updating Objects and Arrays in State Immutably

Update complex state safely using spread syntax and array methods. Avoid common mutation bugs.

IMMUTABILITY

SPREAD SYNTAX

ARRAYS

4 Lifting State Up and Sharing Data

Move state to a common parent so sibling components can share data. Practice with a filter and list example.

LIFTING STATE

DATA SHARING

DESIGN

5 Building a Controlled Component Interface

Bind input values to state to create controlled elements. Build a searchable list with instant feedback.

CONTROLLED INPUTS

SEARCH

STATE BINDING

6 Mini Project: Interactive Task Manager

Combine components, props and state to build a task manager with add, complete and delete features.

PROJECT BUILDING

CRUD BASICS

DEBUGGING

M4.3 Hooks, Effects and Component Lifecycle · 6 topics

Learners use advanced hooks to manage side effects, references and reusable logic.

1 Side Effects with the useEffect Hook

Run code after rendering using useEffect and dependency arrays. Understand when effects run and why.

USEEFFECT

DEPENDENCIES

LIFECYCLE

2 Cleaning Up Effects and Avoiding Common Pitfalls

Clear timers and listeners with cleanup functions. Fix infinite loops and stale values.

CLEANUP

TIMERS

DEBUGGING

3 Using useRef for DOM Access and Persistent Values

Access DOM elements directly and store values that do not trigger re-rendering. Build an auto focus input.

USE REF

DOM ACCESS

FOCUS CONTROL

4 Optimizing with useMemo and useCallback

Avoid unnecessary calculations and re-renders using memoization hooks. Learn when optimization is worthwhile.

USE MEMO

USE CALLBACK

PERFORMANCE

5 Managing Complex State with useReducer

Organize state transitions using reducers and actions. Refactor a form or cart to use useReducer.

USEREDUCER

ACTIONS

STATE LOGIC

6 Building Custom Hooks for Reusable Logic

Extract repeated logic into custom hooks. Create hooks for local storage and window size.

CUSTOM HOOKS

REUSABILITY

CLEAN CODE

M4.4 Routing, Forms and Global State · 6 topics

Learners build multi page experiences with navigation, validated forms and shared application state.

1 Client Side Routing with React Router

Configure routes and navigation links to create a multi page feel. Display different pages without reloading.

REACT ROUTER

ROUTES

NAVIGATION

2 Dynamic Routes and URL Parameters

Create detail pages using route parameters and query strings. Build a product detail view.

URL PARAMS

DYNAMIC ROUTES

QUERY STRINGS

3 Protected Routes and Layouts

Use nested routes and shared layouts and restrict pages to logged in users. Redirect visitors appropriately.

NESTED ROUTES

LAYOUTS

ACCESS CONTROL

4 Form Handling and Validation with React Hook Form

Build forms with validation rules and clear error messages. Reduce boilerplate using a form library.

FORMS

VALIDATION

REACT HOOK FORM

5 Sharing Data with the Context API

Avoid prop drilling by providing shared data through context. Build a theme switcher and user context.

CONTEXT API

PROVIDERS

THEMING

6

Global State with Redux Toolkit

Manage large application state using slices, actions and the store. Connect components to the store.

REDUX TOOLKIT

STORE

SLICES

M4.5

Connecting React to APIs and Data · 6 topics

Learners fetch, display and manage remote data while handling loading and error states gracefully.

1

Fetching Data with Fetch and Async Await

Request data from public APIs and display results in components. Handle responses and parse JSON.

FETCH API

ASYNC AWAIT

JSON

2

Handling Loading and Error States

Show spinners and friendly error messages during requests. Design resilient data driven screens.

LOADING STATES

ERROR HANDLING

USER EXPERIENCE

3

Using Axios and Organizing API Services

Create a reusable API layer with Axios and interceptors. Keep network code out of components.

AXIOS

SERVICE LAYER

INTERCEPTORS

4

Data Fetching and Caching with React Query

Simplify server state using queries and mutations with built in caching. Refetch data automatically.

REACT QUERY

CACHING

MUTATIONS

5

Pagination, Search and Infinite Scrolling

Add pagination and debounced search to large data lists. Implement infinite scroll for smooth browsing.

PAGINATION

DEBOUNCE

INFINITE SCROLL

6

Authentication Flow on the Client Side

Build login and logout flows that store tokens securely and attach them to requests. Persist sessions across reloads.

AUTHENTICATION

TOKENS

SESSIONS

M4.6

Styling, Testing and Deploying React Applications · 6 topics

Learners polish interfaces with modern styling, test components and publish a production ready application.

1

Styling with CSS Modules and Tailwind CSS

Apply scoped styles and utility classes to build consistent layouts. Compare both approaches for real projects.

CSS MODULES

TAILWIND CSS

STYLING

2

Building Responsive Layouts and Reusable UI Kits

Create responsive components such as cards, modals and navbars. Assemble them into a small design system.

RESPONSIVE DESIGN

UI COMPONENTS

DESIGN SYSTEM

3

Animations and Micro Interactions

Add smooth transitions and animations to improve user experience. Use Framer Motion for simple effects.

ANIMATION

FRAMER MOTION

UX

4

Testing Components with Vitest and Testing Library

Write unit tests that check rendering and user interactions. Learn what to test and what to skip.

VITEST

TESTING LIBRARY

UNIT TESTING

5

Performance, Accessibility and Code Splitting

Improve load speed using lazy loading and audit pages for accessibility. Apply fixes using Lighthouse reports.

LAZY LOADING

ACCESSIBILITY

LIGHTHOUSE

6

Capstone: Deploying a React Dashboard Application

Build a data dashboard with routing and API data and deploy it to a hosting platform. Configure environment variables and a production build.

DEPLOYMENT

ENVIRONMENT VARIABLES

CAPSTONE

05

SECTION 5

Backend Development with Django, Flask and REST APIs

Learners build secure, database-driven backend systems and production-ready REST APIs using Flask and Django, ready to connect with React frontends.

6 modules · 36 topics

M5.1

Web Server Fundamentals and Flask Essentials · 6 topics

Learners understand how the web works and build their first backend applications with Flask.

1

How the Web Works: HTTP, Requests and Responses

Understand the client-server model, HTTP methods, status codes and headers. Inspect real requests using browser developer tools.

HTTP

CLIENT-SERVER

DEBUGGING

2

Setting Up Flask and Virtual Environments

Create isolated Python environments and launch a first Flask application. Learn project structure and running the development server.

FLASK

VIRTUALENV

PROJECT SETUP

3

Routing, Views and URL Parameters in Flask

Define routes, handle dynamic URL segments and return different response types. Practice with query strings and path variables.

ROUTING

VIEWS

FLASK

4

Templates and Forms with Jinja2

Render dynamic HTML with Jinja2 templates and process form submissions. Apply template inheritance for reusable layouts.

JINJA2

FORMS

TEMPLATES

5

Flask Blueprints and Application Structure

Organize larger Flask apps using blueprints and configuration files. Apply the application factory pattern.

BLUEPRINTS

ARCHITECTURE

CONFIGURATION

6

Error Handling, Logging and Environment Configuration

Handle exceptions gracefully with custom error pages and log application events. Manage secrets using environment variables.

ERROR HANDLING

LOGGING

ENV VARIABLES

M5.2

Databases and ORM Fundamentals · 6 topics

Learners design relational databases and use ORMs to persist application data.

1

Relational Database Design and SQL Basics

Model data with tables, keys and relationships. Write core SQL queries for creating, reading, updating and deleting records.

SQL

SCHEMA DESIGN

RDBMS

2

Advanced SQL: Joins, Aggregations and Indexes

Combine tables with joins and summarize data with grouping. Learn how indexes improve query performance.

JOINS

AGGREGATION

INDEXING

3

SQLAlchemy Models in Flask

Define models and map Python classes to database tables. Perform CRUD operations through the ORM.

SQLALCHEMY

ORM

CRUD

4

Relationships and Migrations with Flask-Migrate

Implement one-to-many and many-to-many relationships. Evolve database schemas safely using migrations.

RELATIONSHIPS

MIGRATIONS

SCHEMA CHANGES

5

Working with MongoDB and NoSQL Concepts

Compare document databases with relational ones and store data in MongoDB using Python. Decide when NoSQL fits a use case.

MONGODB

NOSQL

PYMONGO

6

Query Optimization and Database Best Practices

Avoid common pitfalls such as N+1 queries and unsafe raw SQL. Apply transactions and connection management techniques.

OPTIMIZATION

TRANSACTIONS

BEST PRACTICES

M5.3

Django Core: Models, Views and Templates · 6 topics

Learners build full-featured web applications using the Django framework and its batteries-included approach.

1

Introduction to Django and Project Anatomy

Create a Django project and understand apps, settings and the manage.py tool. Learn the Model-Template-View architecture.

DJANGO

MTV

PROJECT SETUP

2

Django Models and the ORM

Define models, field types and constraints, then query data using the Django ORM. Use the Django shell for experimentation.

MODELS

QUERYSETS

ORM

3

Migrations and the Django Admin Interface

Generate and apply migrations and customize the admin panel. Manage application data without writing extra code.

MIGRATIONS

ADMIN

DATA MANAGEMENT

4

Views, URL Configuration and Templates

Write function-based views and map them with URL patterns. Render pages using the Django template language.

VIEWS

URLS

TEMPLATES

5

Class-Based Views and Generic Views

Reduce repetition with class-based and generic views for lists, details and forms. Understand mixins and reuse.

CBVS

GENERIC VIEWS

REUSABILITY

6 Django Forms, Validation and Static Files

Build model forms with validation and handle user input safely. Serve CSS, JavaScript and media uploads.

FORMS

VALIDATION

STATIC FILES

M5.4 Building REST APIs with Flask and Django REST Framework · 6 topics

Learners design and implement clean, well-structured REST APIs that power modern frontends.

1 REST Principles and API Design Standards

Learn resources, verbs, status codes and versioning conventions. Design endpoints and payloads before coding.

REST

API DESIGN

JSON

2 Building a REST API with Flask

Create CRUD endpoints in Flask returning JSON responses. Test them with Postman and curl.

FLASK API

POSTMAN

CRUD

3 Serialization and Validation with Marshmallow

Convert models to JSON and validate incoming data using schemas. Return helpful error messages to clients.

SERIALIZATION

VALIDATION

MARSHMALLOW

4 Django REST Framework Serializers and API Views

Set up Django REST Framework and build serializers and API views. Compare function-based and class-based API approaches.

DRF

SERIALIZERS

API VIEWS

5 ViewSets, Routers and Nested Resources

Use ViewSets and routers to generate endpoints quickly. Handle nested and related resources cleanly.

VIEWSETS

ROUTERS

RELATIONSHIPS

6 Pagination, Filtering, Searching and API Documentation

Make APIs scalable with pagination and filters. Generate interactive documentation using Swagger and OpenAPI.

PAGINATION

FILTERING

OPENAPI

M5.5 Authentication, Authorization and API Security · 6 topics

Learners protect applications and APIs using industry-standard authentication and security practices.

1 Session Authentication and User Management in Django

Implement registration, login, logout and password reset flows. Customize the user model for real projects.

AUTHENTICATION

USER MODEL

SESSIONS

2 Password Hashing and Secure Credential Storage

Understand hashing, salting and why plain-text storage is dangerous. Apply secure password handling in Flask and Django.

HASHING

SECURITY

CREDENTIALS

3 Token and JWT Authentication for APIs

Issue and verify JSON Web Tokens for stateless authentication. Handle token expiry and refresh flows.

JWT

TOKENS

STATELESS AUTH

4 Permissions, Roles and Access Control

Restrict resources with permissions and role-based access. Apply object-level checks in DRF.

PERMISSIONS

RBAC

AUTHORIZATION

5 Common Web Vulnerabilities and Defenses

Study SQL injection, XSS and CSRF with practical demonstrations. Apply framework protections and secure coding habits.

OWASP

CSRF

XSS

6 CORS, Rate Limiting and Secure API Configuration

Configure cross-origin access for React clients and throttle abusive traffic. Harden headers and HTTPS settings.

CORS

THROTTLING

HTTPS

M5.6 Testing, Integration and Backend Capstone Project · 6 topics

Learners test, connect and deliver a complete backend service integrated with a React frontend.

1 Unit Testing Backend Code with Pytest

Write isolated tests using pytest and fixtures. Measure coverage and test business logic.

PYTEST

UNIT TESTING

FIXTURES

2 Testing Django Views and REST APIs

Use Django test client and DRF test tools to verify endpoints. Cover authentication and error cases.

API TESTING

DJANGO TESTS

COVERAGE

3 Background Tasks, Email and File Uploads

Send emails, handle file uploads and run asynchronous jobs with Celery. Understand task queues and brokers.

CELERY

EMAIL

FILE HANDLING

4

Connecting React Frontend to the Backend API

Consume the API from React using fetch and Axios and manage auth tokens. Resolve integration and CORS issues.

AXIOS

INTEGRATION

REACT

5

Capstone Planning: Designing the Backend Architecture

Define requirements, data models and endpoints for a full-stack project. Document the design and plan milestones.

ARCHITECTURE

PLANNING

DOCUMENTATION

6

Capstone Build, Review and Presentation

Implement the backend service with authentication, CRUD and tests, then present it for review. Receive feedback on code quality and design.

PROJECT DELIVERY

CODE REVIEW

PRESENTATION

06

SECTION 6

Databases, SQL and Authentication for Web Applications

Learners design relational and NoSQL databases, write efficient SQL, connect data layers to Django and Flask apps, and secure applications with modern authentication and authorization.

6 modules · 36 topics

M6.1

Relational Database Fundamentals and Data Modeling · 6 topics

Learners understand how relational databases work and design clean schemas for real applications.

1

Introduction to Databases and DBMS Concepts

Understand why applications need databases and how a database management system stores and retrieves data.

Compare file storage with relational systems.

DBMS

DATA STORAGE

DATABASE BASICS

2

Tables, Rows, Columns and Data Types

Learn how relational tables are structured and choose suitable data types for each column. Set up PostgreSQL and explore it with a client tool.

POSTGRESQL

DATA TYPES

SCHEMA BASICS

3

Primary Keys, Foreign Keys and Constraints

Use keys and constraints to protect data integrity and link tables together. Practice with unique, not null and check rules.

KEYS

CONSTRAINTS

DATA INTEGRITY

4 Entity Relationship Diagrams and Table Relationships

Model one-to-one, one-to-many and many-to-many relationships using ER diagrams. Translate a diagram into a working schema.

ER DIAGRAMS

RELATIONSHIPS

MODELING

5 Normalization and Schema Design Best Practices

Apply the first three normal forms to remove redundancy and update anomalies. Learn when denormalization is a reasonable trade-off.

NORMALIZATION

SCHEMA DESIGN

BEST PRACTICES

6 Designing a Database for a Full Stack Project

Plan the complete schema for a capstone-style web application from requirements. Review and refine designs in a peer session.

REQUIREMENTS

DESIGN REVIEW

PROJECT PLANNING

M6.2 Core SQL for Querying and Managing Data · 6 topics

Learners write SQL statements to create, read, update and delete data confidently.

1 Creating Databases and Tables with DDL

Use CREATE, ALTER and DROP to build and change database structures. Understand migrations as a safe way to evolve schemas.

DDL

CREATE TABLE

MIGRATIONS

2 Inserting, Updating and Deleting Records with DML

Manipulate data using INSERT, UPDATE and DELETE statements. Learn safe habits such as using WHERE clauses carefully.

INSERT

UPDATE

DELETE

3 Retrieving Data with SELECT, WHERE and ORDER BY

Write queries that filter, sort and limit results. Use operators such as IN, BETWEEN and LIKE.

SELECT

FILTERING

SORTING

4 Aggregate Functions, GROUP BY and HAVING

Summarize data with COUNT, SUM, AVG, MIN and MAX. Group results and filter groups to build simple reports.

AGGREGATION

GROUP BY

REPORTING

5 Working with NULL Values, Dates and String Functions

Handle missing data correctly and use built-in functions for text and date manipulation. Avoid common pitfalls with NULL comparisons.

NULL HANDLING

DATE FUNCTIONS

STRING FUNCTIONS

6 SQL Practice Lab with Real World Datasets

Solve a set of progressively harder query challenges on realistic data. Discuss multiple solutions and readability.

PROBLEM SOLVING

QUERY WRITING

PRACTICE

M6.3 Advanced SQL, Performance and Transactions · 6 topics

Learners combine tables, optimize queries and keep data consistent under concurrent use.

1 Joins Across Multiple Tables

Use INNER, LEFT, RIGHT and FULL joins to combine related data. Choose the right join for each business question.

INNER JOIN

LEFT JOIN

MULTI-TABLE QUERIES

2 Subqueries and Common Table Expressions

Break complex problems into readable steps with subqueries and CTEs. Compare their clarity and performance.

SUBQUERIES

CTES

QUERY READABILITY

3 Window Functions and Analytical Queries

Rank rows, compute running totals and compare values across rows using window functions. Apply them to dashboard-style requirements.

WINDOW FUNCTIONS

RANKING

ANALYTICS

4 Indexes and Query Optimization

Learn how indexes speed up lookups and when they hurt write performance. Read query plans with EXPLAIN to find slow spots.

INDEXES

EXPLAIN

OPTIMIZATION

5 Transactions, ACID and Concurrency Control

Group statements into transactions and understand ACID guarantees. Explore isolation levels and common problems such as dirty reads and deadlocks.

TRANSACTIONS

ACID

CONCURRENCY

6 Views, Stored Procedures and Database Security Basics

Use views and stored routines to simplify access and reuse logic. Apply roles and privileges to limit what each user can do.

VIEWS

STORED PROCEDURES

ROLES

M6.4 ORMs and Database Integration with Django and Flask · 6 topics

Learners connect Python web applications to databases using ORMs and manage data safely.

1 Connecting Python to Databases with psycopg and SQLAlchemy Core

Open connections, run parameterized queries and fetch results from Python code. Understand connection management and cleanup.

PSYCOPG

SQLALCHEMY

CONNECTIONS

2 Django Models, Migrations and the Admin Panel

Define models, generate migrations and manage data through the Django admin. Configure the project to use PostgreSQL.

DJANGO MODELS

MIGRATIONS

ADMIN

3 Django ORM Queries, Relationships and Aggregations

Write filters, related lookups and aggregations using the Django ORM. Reduce query counts with `select_related` and `prefetch_related`.

DJANGO ORM

RELATIONSHIPS

QUERY OPTIMIZATION

4 Flask with SQLAlchemy Models and Alembic Migrations

Build declarative models in Flask and version schema changes with Alembic. Compare the Flask approach to Django conventions.

FLASK

SQLALCHEMY ORM

ALEMBIC

5 Preventing SQL Injection and Handling Data Validation

See how injection attacks work and prevent them with parameterized queries and ORM protections. Validate input before it reaches the database.

SQL INJECTION

VALIDATION

SECURE CODING

6 Seeding, Backups and Managing Data in Production

Load initial data with fixtures and seed scripts and plan backups and restores. Review practical habits for managing production databases.

FIXTURES

BACKUPS

DATA MANAGEMENT

M6.5 NoSQL, Caching and Choosing the Right Data Store · 6 topics

Learners work with document stores and caches and learn to select storage technology for different needs.

1 Introduction to NoSQL and When to Use It

Understand document, key-value, column and graph databases and their trade-offs against relational systems. Learn the basics of the CAP theorem.

NOSQL

CAP THEOREM

TRADE-OFFS

2 MongoDB Fundamentals and Document Modeling

Create collections and documents and decide between embedding and referencing. Perform basic CRUD in the Mongo shell.

MONGODB

DOCUMENTS

CRUD

3 Using MongoDB with Python through PyMongo

Connect a Python application to MongoDB and run queries and updates. Build simple aggregation pipelines for summaries.

PYMONGO

AGGREGATION PIPELINE

INTEGRATION

4 Redis Basics and Caching Strategies

Store data in Redis and use it to cache expensive queries and sessions. Learn expiry, invalidation and cache-aside patterns.

REDIS

CACHING

CACHE INVALIDATION

5 Pagination, Search and Handling Large Datasets

Implement efficient pagination and basic full-text search in an application. Avoid performance traps with large tables.

PAGINATION

FULL-TEXT SEARCH

SCALABILITY

6 Choosing the Right Database for Your Application

Evaluate project requirements and decide between relational, document and cache layers. Justify choices in a short architecture discussion.

ARCHITECTURE

DECISION MAKING

POLYGLOT PERSISTENCE

M6.6 Authentication, Authorization and Web Application Security · 6 topics

Learners implement secure login systems and protect applications against common web vulnerabilities.

1 Authentication Concepts and Secure Password Storage

Learn the difference between authentication and authorization and why passwords must be hashed. Use bcrypt or Argon2 with salting.

AUTHENTICATION

PASSWORD HASHING

BCRYPT

2

Sessions, Cookies and Django Authentication System

Understand how sessions and cookies keep users logged in. Implement registration, login and logout with Django auth.

SESSIONS

COOKIES

DJANGO AUTH

3

Token Based Authentication with JWT

Issue and verify JSON Web Tokens for stateless APIs. Handle expiry and refresh tokens and secure their storage on the client.

JWT

REFRESH TOKENS

API SECURITY

4

Role Based Access Control and Permissions

Restrict endpoints and data by user roles and object level permissions. Apply these in Django REST Framework and Flask.

RBAC

PERMISSIONS

AUTHORIZATION

5

OAuth 2.0, Social Login and Email Verification

Add Google or GitHub sign-in using OAuth 2.0 and build email verification and password reset flows. Understand the authorization code flow.

OAUTH 2.0

SOCIAL LOGIN

PASSWORD RESET

6

Protecting Against XSS, CSRF and Common Vulnerabilities

Recognize the OWASP Top 10 risks and apply defenses such as CSRF tokens, CORS rules, input escaping and HTTPS. Finish with a security checklist review.

OWASP

CSRF

XSS PREVENTION

07

SECTION 7

Machine Learning and Generative AI Integration in Applications

Learners build machine learning models and generative AI features, then integrate them into Django and Flask applications with production awareness.

6 modules · 36 topics

M7.1

Machine Learning Fundamentals with NumPy and Pandas · 6 topics

Learners prepare data and understand core machine learning concepts using Python's scientific stack.

1

Introduction to Machine Learning Concepts

Understand supervised, unsupervised and reinforcement learning with real application examples. Map common product features to the right learning type.

ML BASICS

PROBLEM FRAMING

2 NumPy Arrays and Vectorized Computation

Create and manipulate arrays, use broadcasting and replace slow loops with vectorized operations.

NUMPY

VECTORIZATION

3 Data Analysis with Pandas DataFrames

Load, filter, group and merge datasets using Pandas. Summarize data to find patterns before modeling.

PANDAS

DATA ANALYSIS

4 Data Cleaning and Missing Value Handling

Detect duplicates, outliers and missing values and apply sound cleaning strategies.

DATA CLEANING

OUTLIERS

5 Exploratory Data Analysis and Visualization

Use Matplotlib and Seaborn to visualize distributions and relationships. Draw insights that guide feature choices.

MATPLOTLIB

SEABORN

EDA

6 Feature Engineering and Data Preprocessing

Encode categories, scale numeric features and build new features that improve model performance.

FEATURE ENGINEERING

SCALING

ENCODING

M7.2 Building and Evaluating Models with Scikit-learn · 6 topics

Learners train classical models and measure their quality using proper validation techniques.

1 Regression Models for Numeric Prediction

Train linear regression models to predict values such as prices. Interpret coefficients and errors.

LINEAR REGRESSION

PREDICTION

2 Classification with Logistic Regression and KNN

Build classifiers for yes or no and multi-class problems and compare their behavior.

CLASSIFICATION

LOGISTIC REGRESSION

KNN

3 Decision Trees and Random Forests

Train tree-based models and understand feature importance. See how ensembles reduce overfitting.

DECISION TREES

ENSEMBLES

4 Model Evaluation Metrics and Cross Validation

Use accuracy, precision, recall, F1 and ROC curves with k-fold validation to judge models fairly.

METRICS

CROSS VALIDATION

F1 SCORE

5 Hyperparameter Tuning and Pipelines

Automate preprocessing and training with pipelines and optimize settings using grid and random search.

PIPelines

GRIDSEARCH

TUNING

6 Clustering and Recommendation Basics

Group data with K-Means and build a simple similarity-based recommender for an application.

K-MEANS

RECOMMENDERS

UNSUPERVISED

M7.3 Deep Learning and Natural Language Processing Essentials · 6 topics

Learners explore neural networks and text processing techniques that underpin modern AI features.

1 Neural Network Fundamentals

Understand neurons, layers, activation functions and backpropagation at an intuitive level.

NEURAL NETWORKS

BACKPROPAGATION

2 Building Neural Networks with TensorFlow and Keras

Define, compile and train a feed-forward network on a tabular dataset using Keras.

TENSORFLOW

KERAS

TRAINING

3 Image Classification with Convolutional Networks

Train a CNN to classify images and learn how convolution and pooling extract visual features.

CNN

COMPUTER VISION

4 Transfer Learning with Pretrained Models

Reuse pretrained models to get strong results with little data and short training time.

TRANSFER LEARNING

FINE-TUNING

5 Text Processing and Sentiment Analysis

Tokenize, vectorize and classify text to build a sentiment analysis feature.

NLP

TF-IDF

SENTIMENT

6 Embeddings and Semantic Similarity

Convert text into embeddings and compare meaning using cosine similarity for smarter search.

EMBEDDINGS

SEMANTIC SEARCH

M7.4 Generative AI and Large Language Model Foundations · 6 topics

Learners understand how large language models work and how to control their output effectively.

1 How Large Language Models Work

Learn transformers, tokens, context windows and why models generate fluent but sometimes wrong text.

LLMS

TRANSFORMERS

TOKENS

2 Prompt Engineering Techniques

Write clear prompts using roles, examples and constraints to get reliable outputs.

PROMPTING

FEW-SHOT

3 Working with LLM APIs in Python

Call hosted model APIs, manage keys securely and handle parameters like temperature and token limits.

LLM API

TEMPERATURE

API KEYS

4 Structured Outputs and Function Calling

Get JSON responses and let models trigger application functions to connect AI with business logic.

JSON OUTPUT

FUNCTION CALLING

5 Streaming Responses and Conversation Memory

Stream tokens to users and manage chat history within context limits.

STREAMING

CHAT MEMORY

6 Image and Speech Generative AI Services

Use image generation and speech-to-text services and decide where multimodal features fit in a product.

MULTIMODAL

SPEECH

IMAGE GENERATION

M7.5 Retrieval Augmented Generation and AI Agents · 6 topics

Learners ground model answers in private data and build tool-using assistants.

1 Introduction to Retrieval Augmented Generation

Understand why grounding answers in documents reduces hallucination and how the RAG pipeline flows.

RAG

HALLUCINATION

2 Document Loading and Text Chunking

Load PDFs and web content, then split text into chunks that preserve meaning for retrieval.

CHUNKING

DOCUMENT LOADERS

3 Vector Databases and Similarity Search

Store embeddings in a vector database and retrieve the most relevant chunks for a query.

VECTOR DB

FAISS

RETRIEVAL

4 Building a Question Answering Chatbot

Combine retrieval and an LLM to answer questions over custom documents with source citations.

LANGCHAIN

CHATBOT

CITATIONS

5 AI Agents and Tool Usage

Build agents that choose tools such as search, calculators and databases to complete multi-step tasks.

AGENTS

TOOLS

PLANNING

6 Evaluating and Improving AI Responses

Test answer quality, tune retrieval settings and add guardrails against unsafe or off-topic output.

EVALUATION

GUARDRAILS

M7.6 Integrating AI Features into Full Stack Applications · 6 topics

Learners deploy models behind APIs and connect them to React frontends responsibly.

1 Saving and Loading Trained Models

Serialize models with joblib and Keras formats and version them for reuse in applications.

JOBLIB

MODEL VERSIONING

2 Serving Predictions with Flask and Django APIs

Wrap trained models in REST endpoints with input validation and clear error responses.

MODEL SERVING

REST

VALIDATION

3 Connecting AI Endpoints to the React Frontend

Build chat and prediction interfaces in React with loading states and streamed output.

REACT

FETCH API

AI UI

4 Background Tasks and Caching for AI Workloads

Use task queues and caching to handle slow model calls and reduce API cost.

CELERY

CACHING

PERFORMANCE

5 Responsible AI, Privacy and Cost Control

Address bias, data privacy, rate limits and token costs when shipping AI features.

AI ETHICS

PRIVACY

COST MANAGEMENT

6

Capstone: Deploying an AI Powered Web Application

Build and deploy an end-to-end app combining a trained model, an LLM feature and a React interface.

CAPSTONE

DEPLOYMENT

INTEGRATION

08

SECTION 8

Deployment, Docker, Cloud and Interview Preparation

Learners package, deploy and monitor full stack AI applications using Docker, CI/CD and cloud platforms, then prepare for technical interviews and job placement.

6 modules · 36 topics

M8.1

Production Readiness for Full Stack Applications · 6 topics

Learners prepare Django, Flask and React applications for safe and stable production use.

1

Environment Variables and Configuration Management

Separate secrets and settings from code using environment files and settings modules. Configure development, testing and production environments cleanly.

ENVIRONMENT VARIABLES

CONFIG MANAGEMENT

SECRETS

2

Production Settings for Django and Flask Apps

Set debug flags, allowed hosts, static file handling and secure cookies for production. Understand common misconfigurations that cause outages.

DJANGO SETTINGS

FLASK CONFIG

SECURITY

3

Web Servers with Gunicorn and Nginx

Run Python applications behind Gunicorn and use Nginx as a reverse proxy. Learn how requests flow from browser to application.

GUNICORN

NGINX

REVERSE PROXY

4

Building and Optimizing the React Production Bundle

Create optimized React builds and serve them with proper caching. Connect the frontend to the deployed backend API.

REACT BUILD

CACHING

API INTEGRATION

5

Logging, Error Handling and Application Monitoring

Add structured logging and error tracking to catch issues early. Review logs to debug live problems.

LOGGING

ERROR TRACKING

DEBUGGING

6

Performance Tuning and Security Hardening Basics

Apply caching, query optimization, HTTPS and CORS rules to improve speed and safety. Check applications against common OWASP risks.

PERFORMANCE

HTTPS

OWASP

M8.2

Containerization with Docker · 6 topics

Learners package full stack and AI applications into portable containers.

1

Containers and Docker Fundamentals

Understand images, containers and how Docker differs from virtual machines. Install Docker and run your first containers.

DOCKER BASICS

IMAGES

CONTAINERS

2

Writing Dockerfiles for Python Applications

Write Dockerfiles for Django and Flask apps with proper base images and dependency layers. Build and run the resulting images locally.

DOCKERFILE

PYTHON IMAGES

BUILD

3

Multi-Stage Builds and Image Optimization

Reduce image size and build time using multi-stage builds and dockerignore files. Follow best practices for secure images.

MULTI-STAGE BUILD

OPTIMIZATION

IMAGE SECURITY

4

Docker Compose for Multi-Service Applications

Run the backend, frontend and database together using a single Compose file. Manage networks, ports and service dependencies.

DOCKER COMPOSE

NETWORKING

SERVICES

5

Volumes, Networking and Persistent Data

Persist database data using volumes and connect containers through custom networks. Troubleshoot common container communication issues.

VOLUMES

DOCKER NETWORKS

PERSISTENCE

6

Containerizing Machine Learning and GenAI Services

Package a model-serving API with its dependencies into a container. Handle model files, environment keys and startup commands.

ML CONTAINERS

MODEL SERVING

GENAI DEPLOYMENT

M8.3 CI/CD Pipelines and Automation · 6 topics

Learners automate testing, building and releasing code using modern pipelines.

1 Introduction to CI/CD Concepts and Workflows

Learn why continuous integration and delivery matter for team projects. Map out a pipeline from commit to production.

[CI/CD](#)
[WORKFLOWS](#)
[DEVOPS](#)

2 Automated Testing with Pytest for Backend Code

Write unit and API tests for Django and Flask endpoints. Measure coverage and run tests automatically.

[PYTEST](#)
[API TESTING](#)
[COVERAGE](#)

3 Building GitHub Actions Workflows

Create workflow files that run linting and tests on every push or pull request. Read pipeline logs to fix failures.

[GITHUB ACTIONS](#)
[LINTING](#)
[AUTOMATION](#)

4 Building and Publishing Docker Images in Pipelines

Automate Docker image builds and push them to a container registry. Tag images with versions and commit hashes.

[REGISTRY](#)
[IMAGE TAGGING](#)
[PIPELINE BUILD](#)

5 Automated Deployment and Release Strategies

Deploy automatically after successful builds and understand rolling releases and rollbacks. Manage secrets safely inside pipelines.

[AUTO DEPLOY](#)
[ROLLBACK](#)
[PIPELINE SECRETS](#)

6 Frontend Testing and Quality Checks in Pipelines

Add React component tests and code quality checks to the pipeline. Block merges when checks fail.

[REACT TESTING](#)
[QUALITY GATES](#)
[PULL REQUESTS](#)

M8.4 Cloud Deployment of Full Stack AI Applications · 6 topics

Learners deploy complete applications to cloud platforms and manage them in production.

1 Cloud Computing Fundamentals and Service Models

Understand IaaS, PaaS and serverless along with regions, pricing and shared responsibility. Choose the right model for a project.

[CLOUD BASICS](#)
[IAAS AND PAAS](#)
[PRICING](#)

2 Deploying Backend APIs on Render and Railway

Deploy Django and Flask apps to beginner-friendly platforms with managed databases. Configure environment variables and custom domains.

[PAAS DEPLOYMENT](#)
[MANAGED DATABASE](#)
[DOMAINS](#)

3 Hosting React Frontends on Vercel and Netlify

Deploy the React application with automatic builds from Git. Configure redirects and environment variables for API access.

VERCEL

NETLIFY

STATIC HOSTING

4 Deploying on AWS EC2 with Docker

Launch a virtual server, install Docker and run the application stack. Configure security groups, SSH access and elastic IPs.

AWS EC2

SSH

SECURITY GROUPS

5 Cloud Storage, Databases and Managed Services

Use S3 for file storage and managed relational databases such as RDS. Connect them securely to the deployed application.

AWS S3

RDS

MANAGED SERVICES

6 Domains, SSL Certificates and Cost Management

Connect custom domains, enable HTTPS and set up budget alerts. Learn practical ways to keep cloud costs low.

DNS

SSL

COST CONTROL

M8.5 Capstone Project and Professional Portfolio · 6 topics

Learners build, deploy and present an end-to-end AI full stack project that showcases their skills.

1 Capstone Planning, Requirements and Architecture Design

Select a project idea and define features, user stories and system architecture. Plan milestones and the tech stack.

REQUIREMENTS

ARCHITECTURE

PLANNING

2 Capstone Build Sprint: Backend and AI Integration

Develop the API, database and AI features for the capstone with mentor guidance. Apply testing and version control practices.

BACKEND BUILD

AI INTEGRATION

GIT WORKFLOW

3 Capstone Build Sprint: Frontend and User Experience

Build the React interface and connect it to the backend. Improve usability, responsiveness and error states.

REACT UI

UX

API CONNECTION

4 Capstone Containerization and Cloud Deployment

Dockerize the complete project and deploy it with a CI/CD pipeline. Verify the live application end to end.

DOCKER

CI/CD

LIVE DEPLOYMENT

5 Writing Documentation and Professional README Files

Document setup steps, architecture and API usage clearly. Add diagrams, screenshots and demo links to the repository.

DOCUMENTATION

README

TECHNICAL WRITING

6 Project Demo and Presentation Skills

Present the capstone to a panel, explain design decisions and answer technical questions. Receive structured feedback for improvement.

DEMO

PRESENTATION

FEEDBACK

M8.6 Interview Preparation and Career Launch · 6 topics

Learners prepare for technical interviews and build a strong job search strategy.

1 Python and Data Structures Interview Questions

Practice commonly asked Python, OOP and data structure questions with clear explanations. Improve speed and accuracy on coding rounds.

PYTHON INTERVIEWS

DATA STRUCTURES

CODING ROUNDS

2 Full Stack and System Design Interview Basics

Answer questions on REST APIs, databases, authentication and React. Learn to explain simple system designs step by step.

FULL STACK QUESTIONS

SYSTEM DESIGN

REST

3 AI and Machine Learning Interview Questions

Review core machine learning concepts and GenAI integration topics asked in interviews. Explain model choices and project trade-offs.

ML CONCEPTS

GENAI QUESTIONS

TRADE-OFFS

4 Resume, LinkedIn and GitHub Profile Optimization

Build an ATS-friendly resume and a strong LinkedIn and GitHub presence. Highlight projects with measurable outcomes.

RESUME

LINKEDIN

GITHUB PROFILE

5 Mock Technical Interviews and Feedback

Take live mock interviews covering coding, projects and concepts. Review performance and fix weak areas.

MOCK INTERVIEW

FEEDBACK

PROBLEM SOLVING

6 HR Interviews, Salary Negotiation and Job Search Strategy

Prepare for behavioral questions and learn how to negotiate offers. Plan job applications, networking and referrals.

HR INTERVIEW

NEGOTIATION

JOB SEARCH

07 · SUBJECT MATTER EXPERTS



Meet Our SME Board Members

Expert professionals guiding your journey through real-world industry knowledge and practical experience.



SME BOARD MEMBER · SENIOR SME

Krishna Peketi

Data Analyst

Amazon · Subject Matter Expert

Analyzing organizational data, generating insights, and developing institutional reports to support informed business decisions.

Analyzing datasets, identifying trends, and transforming complex data into actionable insights. Developing dashboards, preparing analytical reports, and supporting data-driven decision-making to improve business performance, operational efficiency, and strategic planning across organizational functions.



SME BOARD MEMBER · SENIOR SME

Karteek Ch

Associate

Tata Consultancy Services · Subject Matter Expert

Technology professional, mentor, and engineer specializing in Oracle, AI, IoT, and emerging technologies.

Technology professional and TCS Engineer specializing in Oracle, Java, Python, AI, and IoT. An experienced Subject Matter Expert and Mentor at Intern Stump, passionate about practical learning, emerging technologies, and industry-ready student development.



SME BOARD MEMBER · ASSOCIATE SME

Vinay Jannu

Finance Manager

The Big Day by Chai · Associate Subject Matter Expert

Provides financial planning, investment guidance, budgeting, and strategic advisory to support informed financial decisions.

Vinay is a dedicated Financial Consultant specializing in financial planning, investment advisory, budgeting, and wealth management. He helps individuals and businesses make informed financial decisions, manage risks, optimize resources, and develop effective strategies to achieve financial stability and long-term growth.



SME BOARD MEMBER · SENIOR SME

Harshavardhan Kumar

Lead Consultant

Intern Stump Solutions · Subject Matter Expert

Computer Science professional specializing in AI, software development, education, and UI/UX consulting.

Harshavardhan Kumar Jajam is a technology professional, educator, and Lead UI/UX Consultant at Intern Stump Solutions, specializing in Artificial Intelligence, Python, Data Science, Full-Stack Development, cloud computing, and industry-oriented technical education.



SME BOARD MEMBER · SENIOR SME

Ruman Aktar

Founder

Rumandev.com · Subject Matter Expert

Guiding backend API integration, full-stack database architecture, and scalable Web application pipelines.

Technology professional specializing in backend API integration, full-stack database architecture, and scalable web application pipelines, with a focus on system performance, seamless data flow, application reliability, and efficient software development practices.

08 · TOOLS AND TECHNIQUES



Key Tools and Techniques

Master industry-standard methodologies used by top-tier professionals, gaining practical exposure to tools, frameworks, and strategies for results.

INDUSTRY STANDARD

01

Python

The core programming language used for backend logic, automation and AI development throughout the course.

BACKEND CORE

02

Django

A powerful web framework for building secure, scalable backends with built in ORM and admin tools.

MOST IN DEMAND

03

React

A popular JavaScript library for building fast, component based user interfaces.

PRODUCTION READY

04

PostgreSQL

A reliable open source relational database used to store and query application data with SQL.

API BUILDING

05

Django REST Framework

A toolkit for building well structured REST APIs with serialization, permissions and authentication.

ML ESSENTIALS

06

Scikit-learn

A machine learning library for training, tuning and evaluating models on structured data.

GENERATIVE AI

07

LLM APIs and Prompt Engineering

Techniques for integrating generative AI models into applications with well designed prompts.

TEAM WORKFLOW

08

Git and GitHub

Version control tools for tracking changes, collaborating with teams and showcasing your code.

DEPLOYMENT SKILL

09

Docker

A containerisation tool that packages applications so they run consistently across any environment.

Many More Tools Integrated in Curriculum

09 · COURSE PROJECTS



Projects You Will Work On

Build a strong portfolio by working on industry-relevant projects that simulate real job scenarios, helping you gain confidence and career-ready skills.

INDUSTRY-LED PROJECT

01

Python Expense Tracker CLI

Build a command line app that records, categorises and summarises daily expenses. Store data in files and practice functions, OOP and error handling.

Duration: 6-10 hours

INDUSTRY-LED PROJECT

02

Responsive Portfolio Website

Design a personal portfolio using HTML, CSS and JavaScript. Make it mobile friendly and host it on GitHub Pages.

Duration: 6-10 hours

INDUSTRY-LED PROJECT

03

React Task Manager

Create a task management interface with components, state and hooks. Add filtering, editing and local storage persistence.

Duration: 8-12 hours

INDUSTRY-LED PROJECT

04

Blog Platform with Django

Develop a full blog with user login, post creation, comments and an admin panel. Use Django ORM with a relational database.

Duration: 10-15 hours

INDUSTRY-LED PROJECT

05

REST API for an E-commerce Store

Build secure REST APIs for products, carts and orders using Django REST Framework. Add JWT authentication and test endpoints with Postman.

Duration: 10-15 hours

INDUSTRY-LED PROJECT

06

House Price Prediction Model

Train a regression model using Scikit-learn on a real housing dataset. Evaluate it and serve predictions through a Flask API.

Duration: 8-12 hours

INDUSTRY-LED PROJECT

07

AI Chatbot Using LLM API

Create a chatbot that uses a large language model API with prompt design and conversation memory. Connect it to a React chat interface.

Duration: 10-14 hours

INDUSTRY-LED PROJECT

08

Resume Analyzer with NLP

Build a tool that extracts skills from uploaded resumes and scores them against job descriptions. Present results in a clean dashboard.

Duration: 10-15 hours

INDUSTRY-LED PROJECT

09

Capstone Full Stack AI Application

Plan, build and deploy an end to end product combining React, Django, a database and an AI feature. Containerise it with Docker and host it on the cloud.

Duration: 20-30 hours



10 · COURSE ENROLLMENT

Python AI Full Stack Development

Learn Python, web development, databases and AI from scratch in live mentor-led sessions. Build 9 real projects and become job-ready as a full stack developer with AI skills.

Starts	To be announced	Enrolment closes	Batch details soon
Duration	6 Months	Timings	To be announced
Batch	Batch details updating soon	Batch status	Batch open · Batch dates announced soon

Your plan

One-time fee · inclusive of GST

Plan 1

Course + Live Work Experience

Build real projects and earn a live work experience certificate.

₹7,999

₹30,000 · 73% OFF

Live classes & completion certificate	✓ Included
---------------------------------------	------------

Live projects	✓ Included
---------------	------------

Live work experience certificate	✓ Included
----------------------------------	------------

8C appraisal report	✓ Included
---------------------	------------

SME AI (Insyrio)	✓ Included
------------------	------------

Live chat with SME	✗ Not included
--------------------	----------------

Placement assistance	✗ Not included
----------------------	----------------

Plan perks	15% off across Intern Stump - coins, self-paced courses, masterclasses, mock interviews 20% off membership plans
------------	---

Also included

- Live classes with industry SMEs
- Course completion certificate
- Live projects reviewed by your SME

Good to know

Completion certificate

Verifiable ID - a sample certificate can be previewed on the course page.

8C Appraisal System

How the eight Cs work - see the 8C Appraisal section of this brochure.

Part-pay with iSure Coins

Use your coin balance at checkout.

Need help choosing?

See the FAQs or book a free demo.

11 · 8C APPRAISAL



World's First Ed-Tech with 8C Appraisal

Verified 8C Appraisal Certificate: every candidate receives an official corporate readiness report shared directly with hiring partners upon course completion.

The 8C Appraisal is a proprietary evaluation matrix measuring holistic performance across 8 career pillars for authentic proof-of-work:

Communication

Configuration

Chronicle

Completeness

Cognizance

Conception

Comprehension

Case Study



Frequently Asked Questions

Find answers to common questions about our learning tracks, placement assistance, and 8C evaluation model, or connect directly with an academic consultant.

Q1 Do I need experience to join Python AI Full Stack Development?

No. The course starts from Python basics and gradually moves to advanced full stack and AI topics, so complete beginners can join comfortably.

Q2 What is the duration of the course?

The course runs for 6 Months with live mentor-led sessions, assignments and project work throughout.

Q3 How much is the course fee?

The current fee is ₹7,999, reduced from ₹30,000, which gives you a saving of 73%. Check the page for any ongoing offers.

Q4 When does the next batch start?

The upcoming batch is the next batch and begins on soon. Classes are held at to be announced.

Q5 Are the classes live or recorded?

Classes are taught live by industry mentors so you can ask questions in real time. Session recordings are shared for revision.

Q6 What projects will I build?

You will complete 9 hands-on projects, from a Python CLI app and React interfaces to REST APIs, an AI chatbot and a deployed capstone application.

Q7 Will I get a certificate after completion?

Yes. You receive a course completion certificate from Intern Stump once you finish the required assignments and projects.

Q8 Does this Technical course help with placements?

Yes. You get resume building, portfolio guidance, mock interviews and job preparation support. These improve your readiness, though final selection depends on your performance.

Ready to take the next step?

Python AI Full Stack Development · Live Course



Scan to view details & enrol

internstump.com/courses/python-ai-full-stack-development



WEBSITE
internstump.com



EMAIL
info@internstump.com



CALL / WHATSAPP
+91 7396168247



Intern Stump Solutions · Generated on 25 Oct 2026, 02:55 PM IST from the live details on internstump.com. Prices, batches and schedules can change.