# Python Lectureflow

| **Module-1) SE - Overview of IT Industry** | **5** |
|---|---|

- Session 1 - Welcome to the IT World Explain careers in IT (developer, tester, designer, data analyst, AI engineer). Discuss product vs service companies; real-world examples (Google, Zoho, TCS). Overview of how teams collaborate using GitHub, Slack, Jira. Trainer Demo: Show an open-source GitHub project (e.g., “Awesome-Java”) and walk through contributors’ roles. AI Tool Tip: Ask ChatGPT to list top emerging tech jobs (to discuss in class).
- Session 2: How Software Works Define software application and its components. Explain software architecture layers: UI ? Business Logic ? Database. Compare desktop, web, and mobile apps. Explain client-server communication with example (browser request ? backend ? database ? response). Trainer Demo: Use Draw.io or Lucidchart AI to visually show data flow (e.g., for “Instagram feed refresh”).
- Session 3: Internet &amp;amp; Networking Basics Internet Fundamentals: How computers communicate. Protocols: Define HTTP, HTTPS, FTP, SMTP (overview only, 5 mins each). Domain &amp;amp; DNS: Explain how domain names map to IP addresses. Demonstrate using nslookup google.com. Explain DNS hierarchy: root, TLD, domain, subdomain. IP Addressing: What is an IP (IPv4 vs IPv6)? HTTP Lifecycle: Request ? Response ? Status Codes (focus on 200, 404, 500). Trainer Demo: Use browser DevTools ? Net
- Session 4: Software Development Lifecycle (SDLC) Explain SDLC phases: Requirement Design Development Testing Deployment Maintenance. Discuss Agile vs Waterfall, Scrum roles (PO, Scrum Master, Dev Team). Relate each phase to a real-world app (e.g., “Spotify new feature rollout”). Trainer Demo: Show a Jira board mock-up with stories in “To Do / In Progress / Done.”
- Session 5: Git &amp;amp; Version Control Why version control is needed. Git concepts: Repository, Commit, Branch, Merge, Push, Pull. GitHub overview: Forks, Pull Requests, Collaboration. Trainer Demo: Initialize a local Git repo (git init, git add ., git commit) Push to GitHub. AI Integration: Use GitHub Copilot Chat to explain git errors in real-time. Depth: Students should confidently push their first small project to GitHub.

| **Module-2) SE- Introduction to Web Technologies - HTML and CSS** | **12** |
|---|---|

- Session 1: Introduction to Web Design Coverage: How browsers render HTML + CSS. Difference between front-end &amp;amp; back-end. Developer tools overview (Chrome DevTools). Demo: Inspect elements on Instagram home page. Depth: Conceptual understanding of DOM structure.
- Session 2: HTML Page Structure Coverage: HTML boilerplate (&amp;lt;!DOCTYPE&amp;gt;, &amp;lt;html&amp;gt;, &amp;lt;head&amp;gt;, &amp;lt;body&amp;gt;). Tags: Headings, paragraphs, lists, anchors, images, tables. Demo: Create personal “About Me” page.
- Session 3: Forms &amp;amp; Inputs Coverage: Form elements (input, select, textarea), labels, validation attributes. Demo: Registration form for Social Bio Link App. Depth: Show POST request preview in DevTools Network tab.
- Session 4: Semantic HTML 5 Coverage: Semantic tags (header, nav, article, footer). Demo: Rebuild a YouTube-style page layout.

- Session 5: Introduction to CSS 3 Coverage: Inline, Internal, External styles; Selectors, Colors, Font families. Demo: Color-theme switch between light and dark mode.
- Session 6: Box Model &amp;amp; Positioning Coverage: Margin, Padding, Border, Display properties, Position (relative, absolute, fixed). Demo: Create a profile-card layout.
- Session 7: Flexbox &amp;amp; Grid Coverage: CSS Flexbox (container, justify, align) and Grid (template columns &amp;amp; rows). Demo: Responsive gallery of social icons.
- Session 8: Responsive Design Coverage: Media queries, Viewport, Breakpoints. Demo: Make the profile card mobile-friendly.
- Session 9: Bootstrap Basics Coverage: Bootstrap Grid System, Containers, Rows, Columns. Demo: 2-column layout with profile image and link list.
- Session 10: Bootstrap Components Coverage: Navbar, Buttons, Cards, Carousel, Forms. Demo: Navbar for Social Bio Link App.
- Session 11: Bootstrap Utilities &amp;amp; Customization Coverage: Spacing, Typography, Colors, Custom CSS overrides. Demo: Theme switcher using Bootstrap classes.
- Session 12: Integration Review Coverage: Integrate HTML + CSS + Bootstrap for a cohesive layout. Demo: Static mock-up of Bio Link landing page.
- Mini Project 6 – "Social Bio Link Page – UI Design" Objective: Build a personal bio link page (links to Instagram, LinkedIn, Spotify, etc.) with responsive design. Trainer Notes: Guide students to implement a form that accepts link inputs and shows preview cards.

| **Module 3) SE - Introduction to Programming** | **15** |
|---|---|

- Session 1: What is Programming? Topics: Define "program" and "programming language." Explain compilers and interpreters. Discuss importance of algorithms and flowcharts. Trainer Activity: Draw a flowchart for "Ordering food online."
- Session 2: Writing Your First C Program - Topics: Setting Up Environment -Installing a C Compiler (e.g., GCC), Choosing an IDE (DevC++. VS Code, Codeblocks, etc) Basic structure of C program (header, main, return). Compile and run using GCC or VS Code terminal. Example: "Hello, TOPS Technologies!"
- Session 3: Data Types and Variables Topics: Data types (int, float, char, double). Variable naming rules. Constants, literals. Example: Store and print student's name, age, marks.
- Session 4: Operators and Expressions Topics: Arithmetic, Logical, Relational, Increment/Decrement. Operator precedence. Example: Simple calculator for addition, subtraction, etc.
- Session 5: Conditional Statements Topics: If, Else If Ladder, Nested If, Switch Case. Example: "Study Mood Bot" – takes user's input and prints motivation message
- Session 6: Loops (For, While, Do-While) Topics: Difference between entry-controlled and exit-controlled loops. Use cases (e.g., menu-driven programs). Example: Countdown timer using loops.
- Session 7: Nested Loops &amp;amp; Pattern Printing Topics: Using nested loops for grids or triangle patterns. Example: Star pyramid pattern.
- Session 8: Functions Topics: Declaration, Definition, Calling. Pass-by-value vs pass-by-reference.
- Session 9: Arrays Topics: 1D &amp;amp; 2D Arrays, Array Traversal. Example: Record 7-day screen time &amp;amp; calculate average.
- Session 10: Strings Topics: Declaring and handling strings. Common functions: strlen(), strcpy(), strcmp(). Example: Create a username from name input.

- Session 11: Pointers Topics: Pointer declaration, dereferencing, pointer arithmetic. Example: Swap numbers using pointers. Trainer Demo: Use memory diagram to explain address referencing.
- Session 12: Structures Topics: Declaring, initializing structures. Nested structures. Example: Store student details (name, marks, grade).
- Session 13: File Handling Topics: Open, write, read, close files. Modes: r, w, a. Example: Write and read "daily goals.txt."
- Session 14: Debugging &amp;amp; Best Practices Topics: Common syntax/runtime errors. Importance of commenting &amp;amp; indentation.
- Mini Project 1: "Student Productivity Tracker" Objective: Combine loops, arrays, and file handling to build a console-based app that logs daily study hours and generates a weekly progress report. Trainer Coverage: Guide students to design menu-driven interface. Discuss file I/O for persistence.

| **Module-4) Introduction to OOPS Programming** | **8** |
|---|---|

- Session 1: Procedural vs Object Thinking Topics: Limitations of C (global data, no encapsulation). Why OOP solves these. Example: Compare "Task List" written in C vs OOP style.
- Session 2: Classes and Objects Topics: Define class &amp;amp; object, syntax, member functions. Example: Class Note (title, date, isDone).
- Session 3: Constructors &amp;amp; Destructors Topics: Default, Parameterized, Copy constructors. Destructor lifecycle. Example: Auto-save user data upon exit.
- Session 4: Inheritance Topics: Single, Multilevel, Hierarchical inheritance. Example: ContentCreator ? YouTuber / Podcaster.
- Session 5: Polymorphism Topics: Compile-time (overloading) vs Runtime (overriding). Example: uploadContent() works differently for YouTube vs Instagram.
- Session 6: Encapsulation &amp;amp; Abstraction Topics: Access modifiers: public, private, protected. Abstract classes &amp;amp; pure virtual functions (conceptual). Example: Class Platform with abstract upload() method.
- Session 7: File Handling Topics: Streams (ifstream, ofstream), reading/writing files. Example: Save and retrieve creator analytics data.
- Mini Project 2: "Creator Dashboard Lite" Objective: Create a console app for digital creators to manage content ideas. Trainer Guidelines: Implement Content class (title, platform, views, status). Allow user to add, edit, delete content items. Store data using file handling.

| **Module-5) SE - Database Management – SQL &amp; PL/SQL** | **9** |
|---|---|

- Session 1: Introduction to Databases Trainer Coverage: What is a database, DBMS, RDBMS (with examples). Difference between MySQL, PostgreSQL, Oracle, SQLite. Tables, rows, columns, and relationships. Demo: Show data stored in MySQL Workbench.
- Session 2: SQL Basics &amp;amp; Commands Trainer Coverage: SQL syntax rules. DDL commands: CREATE, ALTER, DROP. Creating basic users and expenses tables. Demo: Execute commands in MySQL CLI / Workbench.
- Session 3: DML Operations Trainer Coverage: Insert, Update, Delete records. WHERE and LIMIT usage. Example: Insert 5 sample users and edit their data.

- Session 4: Data Retrieval with SELECT Trainer Coverage: SELECT queries, filtering, sorting, aliases. Aggregate functions (SUM, AVG, COUNT). Demo: Generate “total expenses” by category.
- Session 5: Joins and Relationships Trainer Coverage: Foreign keys and primary keys. Inner Join, Left Join, Right Join. Demo: Link users with their transactions table.
- Session 6: Grouping and Subqueries Trainer Coverage: GROUP BY, HAVING. Subqueries and nested queries. Example: “Show users whose average daily spend &amp;gt; ?500.”
- Session 7: Constraints &amp;amp; Views Trainer Coverage: Constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY. Creating and querying VIEWS. Example: Create “ActiveUserView” showing users with &amp;gt;5 transactions.
- Session 8: Introduction to PL/SQL Trainer Coverage: PL/SQL block structure (DECLARE, BEGIN, EXCEPTION, END). Variables and cursors. Control structures (IF, LOOP). Demo: Simple stored procedure to calculate user’s total spend.
- Session 9: Transactions &amp;amp; Error Handling Trainer Coverage: COMMIT, ROLLBACK, SAVEPOINT. Triggers: use case “Auto update user’s balance after expense entry.” AI Integration: Use ChatGPT to simulate test triggers and rollback scenarios.
- Mini Project 3 – “Expense Tracker DB” Description: Design the backend for a Personal Expense Tracker App. Create users, transactions, and categories tables. Write SQL scripts for CRUD operations. Add a stored procedure for monthly summary generation. Trainer Deliverables: Students should submit .sql file with queries. Test queries via MySQL Workbench live.

| **Module 6) Python - Fundamentals of python language** | **12** |
|---|---|

- Session 1 — Introduction to Python What is Python? Why Python? Use cases Installing Python &amp; VS Code Running first Python script Python code execution model
- Session 2 — Variables &amp; Data Types Creating variables Naming rules Primitive &amp; non-primitive types type(), id() Examples: ? Create variables for a student profile.
- Session 3 — Input, Output, Typecasting input() &amp; print() int(), float(), str() conversions Escape characters Examples: Basic calculator.
- Session 4 — Conditional Statements if, else, elif Nested if Examples: ? Grade calculation ? Blood donation eligibility
- Session 5 — Loops (for) for loop basics Looping through strings, lists Examples: ? Print fruit names ? Count characters in string
- Session 6 — Loops (while) + Patterns while loop Infinite loop cases Star patterns
- Session 7 — Break, Continue, Pass break usage continue usage pass usage Examples: ? Skip “Banana” ? Break at “Banana”
- Session 8 — Strings Indexing Slicing String methods Examples: reverse string, substring extraction.
- Session 9 — Lists Creating lists Append, insert, remove, pop Sorting
- Session 10 — Tuples Tuple creation Accessing values immutability Examples: ? Tuple slicing ? Tuple concatenation
- Session 11 — Dictionaries Key/value pairs Adding/updating entries keys(), values(), items()
- Session 12 — Functions + Lambda Defining functions Arguments Return values Lambda functions + map, filter, reduce

| **Module 7) Python - Collections, functions and Modules in Python** | **12** |
|---|---|

- Session 1 — List Deep Dive Creating list with multi-data type Updating values Removing values
- Session 2 — List Iteration Patterns for loops on lists range with lists
- Session 3 — List Transformations round(), sorted(), sort() zip() usage
- Session 4 — Tuples Advanced Creating mixed tuples Slicing rules Tuple ? list conversion
- Session 5 — Dictionaries Deep Dive Updating values Nested dictionaries
- Session 6 — Converting Lists ? Dictionary Using zip() Using loops
- Session 7 — Counting Characters &amp; Words freq dictionary word count program
- Session 8 — Functions Revisited Keyword &amp; Positional arguments Default values
- Session 9 — Anonymous Functions Lambda with multiple expressions
- Session 10 — Modules (Custom) Creating student.py Importing module in main file
- Session 11 — math &amp; random modules sqrt, ceil, floor randint, random
- Session 12 — Packages Creating package folder init.py usage
- session 13 - List comprehension - nested list
- session 14 - dynamic nested dictionary creation

| **Module 8) Python - Advance python programming** | **10** |
|---|---|

- Session 1 — File Handling Basics open(), read(), write() File modes: r, w, a
- Session 2 — File Pointer &amp; Cursor Control tell(), seek() Reading line-by-line
- Session 3 — Exception Handling Basics try, except Multiple exceptions
- Session 4 — Custom Exceptions raise keyword Creating user-defined exception AI Tool: ChatGPT ? explain error tracebacks.
- Session 5 — Classes &amp; Objects Class structure init method Creating objects
- Session 6 — Inheritance Single Multilevel Multiple Example: Family class inheritance.
- Session 7 — Polymorphism Method overloading (conceptual in Python) Method overriding
- Session 8 — Modules &amp; Packages import module Creating own module math, random modules
- Session 9 — Regex (re module) match() search() findall()
- Session 10 — GUI Programming with Tkinter Window Labels Buttons Layout
- session11 : Database connecting with python - pymysql or mysql connectivity

| **Module 9) Python - DB and Python Framework** | **20** |
|---|---|

- SESSION 1 — Introduction to Django &amp; Setup Topics What is Django? Why Django vs Flask Django features MVT Architecture Installing Django Creating first Django project Running Django server Trainer Activities Create project: doctorfinder_project Explore project structure
- SESSION 2 — Apps, Settings &amp; URL Routing Topics Creating Django app Project vs App settings.py, urls.py, wsgi.py Mapping URL ? View HttpResponse basics Activities Create app: accounts Add it to INSTALLED_APPS Create first view and URL
- SESSION 3 — Templates &amp; Static Files Topics Templates folder structure Template inheritance (base.html) Loading static files (CSS, JS) Using template tags Activities Build base.html layout Add Bootstrap via CDN

- SESSION 4 — Django Models &amp; Database Connectivity Topics Creating models Model fields SQLite ? MySQL migration Migrations: makemigrations, migrate ORM basics Activities Create model: Doctor, Specialization Connect Django to MySQL using pymysql
- SESSION 5 — Django Admin Customization Topics Enabling admin Adding models to admin Custom list display Search fields, filters Activities Customize admin for Doctor model Include filters for specialization
- SESSION 6 — Django Forms Basics Topics Django Form class GET vs POST CSRF token Handling form submission Activities Build “Add Doctor” form Display validation errors
- SESSION 7 — Django Model Forms Topics ModelForm benefits Automatic form generation Saving form data to model Activities Create DoctorForm Save doctor details from form
- SESSION 8 — User Authentication System Topics User model Signup Login Logout Password hashing Authentication backend Activities Build Register ? Login ? Logout Display logged-in username in navbar
- SESSION 9 — Email Sending in Django Topics SMTP setup Django email functions HTML email templates Activities Send welcome email on signup
- SESSION 10 — Forgot Password + Session Management Topics Django sessions Reset password with OTP link Session expiry Middleware basics Activities Implement forgot password Store OTP in session
- SESSION 11 — Working With Media Files &amp; User Profile Topics MEDIA_ROOT and MEDIA_URL Upload profile picture Serving uploaded files Activities Profile edit page (update photo, name, phone)
- SESSION 12 — CRUD Operations Topics Create, Read, Update, Delete Editing using ModelForm Delete confirmation page Activities CRUD for Doctor: Add ? View ? Edit ? Delete
- SESSION 13 — CRUD Using AJAX Topics AJAX introduction Fetch API Sending JSON request Handling JSON responses Activities Implement delete doctor using AJAX Show success message dynamically
- SESSION 14 — Django ORM: QuerySet Deep Dive Topics filter(), exclude(), annotate() select_related(), prefetch_related() Aggregations Q() objects Activities Search doctors by multiple filters Paginate doctor listing
- SESSION 15 — Django Authentication (Advanced) Topics Login required decorator Role-based access (doctor vs patient) Permissions Groups Activities Restrict admin pages Create role-based dashboards
- SESSION 16 — Django Integrations Topics Social Login (Google) Email APIs (Mailchimp / Mailgun) SMS APIs (Twilio) Activities Integrate Google Login for patient Send OTP via SMS
- SESSION 17 — Payment Gateway Integration Topics Paytm/Stripe/PayPal integration Payment initiation Transaction callback Activities Create “Book Appointment” ? “Pay Now” flow
- SESSION 18 — Maps &amp; Geolocation API Topics Google Maps API Geocoding (address ? latitude, longitude) Embedding map in Django Activities Show doctor clinic location on map Search doctors by distance
- SESSION 19 — GitHub Deployment &amp; PythonAnywhere Hosting Topics Push Django project to GitHub PythonAnywhere deployment Static &amp; media files config Database migration on server Activities Fully deploy Doctor Finder project Test live routes
- SESSION 20 — Capstone: Doctor Finder Project (Integration &amp; Cleanup) Includes Registration, Login, OTP, Forgot Password Profile Management Search Doctors Filters (specialization, location)

Appointment booking Google Maps integration Payment gateway Admin management Live deployment Deliverables Final code on GitHub Final deployment URL Project documentation

| Module 10)Django REST Framework (REST API Development) | 6 |
|---|---|
| • SESSION 1 — Introduction to REST &amp; DRF Setup Topics Covered What is an API? What is a REST API? (Principles: stateless, resource-based, CRUD) JSON vs XML Difference between Django &amp; Django REST Framework Installing DRF Adding rest_framework to INSTALLED_APPS Creating a new Django app: api Serializer vs ModelSerializer overview First test API endpoint Trainer Demonstration Create project folder /api Create simple “Hello API” endpoint using APIView or function-base<br>• SESSION 2 — Serializers + CRUD API Development Topics Covered Creating models for API Writing serializers (Serializer + ModelSerializer) Creating CRUD endpoints: Create (POST) Read (GET) Update (PUT/PATCH) Delete (DELETE) Using APIView / GenericAPIView / Mixins JSON responses Status codes (HTTP_200, 201, 404, 400) Trainer Demonstration Build CRUD API for Doctor model: /api/doctors/ /api/doctors/&lt;id&gt;/<br>• SESSION 3 — ViewSets, Routers &amp; Pagination Topics Covered ViewSets (ModelViewSet) Routers &amp; automatically generated URLs Default Router vs SimpleRouter Pagination types: PageNumberPagination LimitOffsetPagination CursorPagination Ordering and filtering basics Trainer Demonstration Convert CRUD API to ModelViewSet Implement pagination /api/doctors/?page=2 Practice Add ordering by name or specialization.<br>• SESSION 4 — Authentication &amp; Permissions Topics Covered BasicAuthentication TokenAuthentication SessionAuthentication Permissions: AllowAny IsAuthenticated IsAdminUser Custom Permissions JWT Overview (Optional light introduction) Trainer Demonstration Protect /api/doctors/ using authentication Generate auth token for user (using rest_framework.authtoken) AI Integration ChatGPT ? Explain “Token vs Session Auth” for interview prep.<br>• SESSION 5 — Third-Party API Integrations Topics Covered Integrating Popular External APIs: OpenWeatherMap API Google Maps Geocoding API REST Countries API GitHub API Twitter API (theoretical due to OAuth limits) Trainer Demonstration Create endpoint /api/weather/&lt;city&gt;/ DRF view calls external weather API Show JSON response to frontend AI Tool Integration Postman AI ? test API endpoints ChatGPT ? generate Python requests code snippet.<br>• SESSION 6 — Advanced Features + Deployment Topics Covered Email Sending API (MailChimp, Mailgun) SMS API (Twilio) Payment APIs (Stripe/PayPal) Social Authentication (Google Login) Custom Response Formatting Exception Handling in DRF Deploying REST API on PythonAnywhere Versioning your API (v1, v2) Trainer Demonstration Integrate sample Twilio SMS API (demo level) Deploy REST API live Test endpoints using Postman | |