1. Introduction to Full Stack Development

A. What is Full Stack Development?
B. Client-Server Architecture & HTTP Basics
C. Components of Python Full Stack: Frontend, Backend & APIs
D. Setting up development environment (Python, IDEs, Git/GitHub)
E. Software development lifecycle overview

2. Python Programming Fundamentals

A. Python installation & interpreter use
B. Variables, data types, operators
C. Input/output and basic scripting
D. Control structures (if, loops)
E. Functions & modules
F. Exception handling

3. Advanced Python

A. Object-Oriented Programming (classes, objects, inheritance, polymorphism)
B. Data structures (lists, tuples, dicts, sets)
C. Iterators, generators, lambda functions, comprehensions
D. File handling & standard libraries
E. Virtual environments & package management (pip)

4. Web Fundamentals (Frontend Basics)

A. HTML 5 — structure, elements & forms
B. CSS 3 — styling, selectors, layout fundamentals
C. JavaScript essentials — syntax, DOM, events
D. Frontend interaction with backend servers
E. Responsive design basics

5. Backend Development with Python

A. Python Back-end frameworks overview
B. Flask — setting up applications
C. Routing & URL handling
D. Request/response cycle
E. Templates (Jinja2) and static files
F. Form handling & validation
G. Django introduction (optional comparison)

6. Database Management & Data Integration

A. Introduction to databases (RDBMS & NoSQL)
B. SQL basics: CRUD with MySQL/PostgreSQL
C. ORM (Object-Relational Mapping) with SQLAlchemy or Django ORM
D. MongoDB basics & integration (optional)
E. Database migrations & schema design

7. API Development & RESTful Services

A. Introduction to web services & APIs
B. RESTful API principles
C. Building APIs using Flask/Django REST Framework
D. JSON handling and serialization
E. Authentication & Authorization basics (token/JWT)
F. API versioning & documentation

8. Automation with Python

A. Scripting tasks & system automation
B. File system automation (OS & pathlib)
C. Web automation (Selenium basics)
D. Scheduling jobs (Cron, Windows Task Scheduler)
E. Handling emails, reports & logs programmatically

9. Testing, Debugging & Deployment

A. Debugging tools (pdb, IDE debuggers)
B. Unit testing & integration testing (unittest/pytest)
C. Preparing for deployment
D. Deployment platforms (Heroku, AWS, PythonAnywhere)
E. Environment variables and config management

10. Project Work & Capstone

A. Planning a full stack application
B. Building a complete backend with frontend integration
C. Real-world API integrations (e.g., payment, email)
D. Performance considerations
E. Presentations, documentation & demo

Practical:

- Hands-on exercises include:
- Python programming labs (fundamentals to advanced)
- Frontend integration with backend
- Database CRUD exercises
- REST API creation & testing
- Automation scripts
- Deployment of an end-to-end full stack application