1. **Introduction to Python**

* Variables and data types
* Numeric data types: integers, floating-point numbers, and complex numbers
* Boolean data type: True or False
* Sequence data types: strings, lists, tuples, and range objects
* Mapping data type: dictionaries
* Set data type: sets and frozen sets
* NoneType data type: None
* Control structures

Conditional statements:

* if statements
* if-else statements
* if-elif-else statements

Loops:

* for loops
* while loops

Control statements:

* break statements
* continue statements
* Exception handling:
* try-except statements
* finally statements
* Functions
* Functions
* Recursive function calls
* Anonymous (lambda) functions
* Object-oriented programming
* Input/output
* Error handling
* try-except statements
* finally statements
* Libraries and modules

1. **Django Fundamentals**

* Django project and app structure
* Models
* Views
* Templates
* URL routing
* Settings

1. **Practical Examples**

* Blog Platform
* Student Record Management
* Job Board

1. **Connecting Business Logic with Database Knowledge**

* Understanding the business logic and how to translate it into tables, columns, and relationships
* Object-relational mapping (ORM)
* Querying data

1. **Performance Tuning, Normalization, and Security of Applications**

* Database performance tuning
* Long logic process breakdown
* Normalization (1NF, 2NF, 3NF, etc.)
* Security in 3 levels (Middlewares, Decorators, and Logics)
* Logging and debugging

1. **Django Cutting-edge Concepts**

* Restful API concepts (Restful, Serialization, Swagger)
* Graphql Concepts (Model-based and Object-Based Fields)
* Request and Response Built-in-API

1. **Practical Examples**

* Upgrading Blog Platform
* Upgrading Student Record Management
* Upgrading Job Board

1. **Data Visualization and Reporting**

* Finding or defining the purpose of the report
* Choosing an appropriate visualization
* Ensuring data accuracy
* Export-based Report (PDF, Excel, CSV, and Word)

1. **Introduction to System Decoupling Segmentation (Microservices)**

* Monolithic vs. microservices architecture
* Event-driven architecture
* API gateway
* Service discovery
* Identifying bounded contexts
* API design
* Data management
* Containerization and orchestration

1. **Advancing to Microservices with Python (Django) and Node (API gateway)**

* Federation-ready Django Application
* Best practice project structure
* Step-by-step project setups
* Internal request
* Message broker
* Resolvers functions and Utils
* DTO and DTO Builders
* Internal Object referencing and Object fusion
* External Object referencing and fusion

1. **Managing Integrations and External Communication**

* Point of Integration (POI) management
* Data dependency with external systems
* Authentication to and from external systems

1. **Take Home**

* What to be done to accelerate your knowledge level
* What are the challenges
* How to handle changes and upgrades without disturbing other logics