SOFTWARE CONSTRUNCTION AND DEVELOPMENT

(SE 5A BATCH 2021)

CONTENT DEV BY:

SE ZARYAB KHAN CLASS OF 2020 DSU

Course Outline (20 Weeks)

Week 1: Course Introduction & Overview

- **Topics:**
  - Course objectives, grading, and policies
  - Overview of software construction & development
  - Introduction to web frameworks and Flask's role in Python web development
- **Activities/Deliverables:**
  - Setup development environment (Python, IDE, Git)

Week 2: Python Refresher & Project Setup

- **Topics:**
  - Brief Python essentials (data types, functions, OOP)
  - Structuring a Python project
  - Introduction to virtual environments
- **Activities/Deliverables:**
  - Practice exercises on Python syntax and OOP
  - Create and activate a virtual environment for the project

Week 3: Flask Basics & Application Structure

- **Topics:**
  - Installing and configuring Flask
  - The concept of app.py or a main application script
  - Basic “Hello World” Flask app and routes
- **Activities/Deliverables:**
  - Build a simple Flask “Hello World” application
  - Explore flask run and debug mode

Week 4: Routing & Request Handling

- **Topics:**
  - URL routing in Flask (@app.route)
  - HTTP request types (GET, POST, PUT, DELETE)
  - Handling query parameters and form data
- **Activities/Deliverables:**
  - Simple routes with dynamic URL segments
  - Implement a basic form submission

Week 5: Jinja2 Templating

- **Topics:**
  - Integrating Jinja2 with Flask
  - Templating basics, variables, control structures, loops
  - Inheritance with template layouts (base.html)
- **Activities/Deliverables:**
  - Create a multi-page Flask app using templates and layout inheritance

Week 6: Static Files & Front-End Integration

- **Topics:**
  - Managing static assets in Flask (CSS, JavaScript, images)
  - Flask’s url_for usage
  - Basic front-end design considerations (Bootstrap or another CSS framework)
- **Activities/Deliverables:**
  - Add styling and basic JavaScript to templates

Week 7: Introduction to Databases & SQLite

- **Topics:**
  - Relational databases overview
  - SQLite features and setup
  - Database schema design basics

Activities/Deliverables:

  - Install SQLite or verify it’s available
  - Create a simple schema (e.g., users, posts)

Week 8: Flask-SQLAlchemy Basics

- **Topics:**
  - Configuring Flask-SQLAlchemy
  - Defining models and relationships
  - Connecting Flask app to SQLite DB
- **Activities/Deliverables:**
  - Implement your first model (e.g., a “User” or “Post” table)
  - Simple CRUD operations (Create, Read)

Week 9: Advanced Queries & Migrations

- **Topics:**
  - Writing queries (filtering, joining, ordering)
  - Database migrations (intro to Flask-Migrate or alternatives)
  - Handling one-to-many or many-to-many relationships
- **Activities/Deliverables:**
  - Practice advanced queries
  - Optional: Integrate Flask-Migrate for schema changes

Week 10: Form Handling & Validation

- **Topics:**
  - Dealing with WTForms or simple form handling in Flask
  - Client-side vs server-side validation
  - Displaying error messages in templates
- **Activities/Deliverables:**
  - Build a registration or contact form with validation checks

Week 11: CRUD Functionality & Jinja2 Integration

- **Topics:**
  - Full CRUD operations in the Flask app (Create, Read, Update, Delete)
  - Displaying and editing database entries with Jinja2
  - Pagination and search
- **Activities/Deliverables:**
  - Implement a feature (e.g., blog posts) with full CRUD
  - Add search and/or pagination to the application

Week 12: Authentication & Authorization

- **Topics:**
  - Introduction to user authentication (sessions, cookies)
  - Using Flask-Login or custom session management
  - Role-based access control (basics)
- **Activities/Deliverables:**
  - Implement user login, logout, and protected routes

Week 13: Application Structure & Best Practices

- **Topics:**
  - Organizing larger Flask projects (blueprints, packages)
  - Configuration management (dev, test, production)
  - Security considerations (SQL injection, CSRF, etc.)
- **Activities/Deliverables:**
  - Refactor your project into a more modular structure
  - Enable CSRF protection for forms

Week 14: Testing & Quality Assurance

- **Topics:**
  - Unit testing in Python (pytest or unittest)
  - Testing Flask routes and database operations
  - Continuous integration concepts
- **Activities/Deliverables:**
  - Write basic unit tests for models and routes
  - Integrate a simple CI workflow (optional)

Week 15: Error Handling & Logging

- **Topics:**
  - Custom error pages (404, 500, etc.)
  - Logging configuration and levels in Flask
  - Monitoring and debugging strategies
- **Activities/Deliverables:**
  - Implement custom error pages in the application
  - Set up a logging system

Week 16: Performance & Optimization

- **Topics:**
  - Database indexing and query optimization
  - Profiling Python applications
- **Activities/Deliverables:**
  - Identify potential performance bottlenecks in your project

Week 17: Packaging & Deployment

- **Topics:**
  - Building a production-ready build (environment variables, config files)
- **Activities/Deliverables:**
  - Explore deployment to a free hosting service or local server

Week 18: Project Work & In-class Presentations

- **Topics:**
  - Students work on a capstone-style project integrating all learned components
  - In-class peer review and progress checks
- **Activities/Deliverables:**
  - Present progress on final project
  - Receive feedback and refine

Week 19: Final Project Demo & Evaluation

- **Topics:**
  - Complete final project
  - Demonstration or presentation of project to class/instructor
  - Code reviews and lessons learned
- **Activities/Deliverables:**
  - Submit final project
  - Live demo session

Week 20: Course Wrap-up & Revision

- **Topics:**
  - Review of key concepts (Flask, Jinja2, SQLite, ORM, best practices)
  - Discussion of industry trends, next steps, and further resources
  - Final Q&A and feedback session
- **Activities/Deliverables:**
  - Course evaluation