IS 218 – Building Web Applications

Section: 001
CRN: 93750
Semester: Fall 2025
Credits: 3 **Delivery**
Mode: Face-to-Face
Days/Times: Tuesday & Friday, 2:30 – 3:50 PM
Location: CKB 313

Instructor Information

- **Instructor:** Keith Williams
- **Office:** GITC 3420
- **Email:** kwilliam@njit.edu (Discord preferred)

Course Description

Students will gain experience in the development of Web based systems using an object-oriented programming language and SQL. Students will learn to develop a web-based system through an intensive hands-on project that requires students to apply real-world problem-solving skills to meet the challenge of developing a web-based information system. Students will learn the basic principles of web-based applications, MVC application design, how to apply object-oriented design patterns, design a relational database, and write SQL queries to create, retrieve, update, and delete information in a database.

Prerequisites: (IS 117 or IT 202) and (CS 100, CS 113, or CS 115)

Learning Outcomes

By the end of this course, students will be able to:

1. Apply **professional Python practices** including OOP, design patterns, and modular design.
2. Follow a **test-driven development (TDD) workflow** using **pytest**.

3. Improve code quality with **linters and formatters** (flake8, black).
4. Design relational databases and implement **SQL CRUD operations**.
5. Integrate **SQLAlchemy ORM** into Python projects for object-relational mapping.
6. Build and document **REST APIs** with FastAPI and Swagger.
7. Containerize and deploy applications with **Docker**.
8. Collaborate using **GitHub workflows** (branches, pull requests, commit history).
9. Demonstrate readiness for **junior developer internships**, applying professional software engineering practices in a team environment.

Course Materials

- Online and instructor-created content
- Cloud services (may require AWS or Azure account verification)

Technologies

- **Python** (OOP, FastAPI, pytest, flake8, black)
- **SQL & SQLAlchemy ORM**
- **Docker**
- **Git & GitHub**
- **REST API tools** (Swagger UI, Postman, cURL)
- **Minimal HTML templating** (Jinja2)

Course Concepts

- Professional Python development practices
- Test-driven development (TDD) with pytest
- Code quality and style enforcement (linters)
- MVC back-end architecture with FastAPI
- Relational data modeling and CRUD operations
- ORM and database integration
- REST API design and documentation
- Unit testing and CI/CD basics
- DevOps (Docker, containerization, deployment)
- GitHub collaboration workflows

Course Format

- **12 Modules**
- New content released **Thursday by noon** each week
- Each module includes: unit intro, hands-on coding exercise, readings
- Communication via **Slack** (Canvas for posting only)

Tentative Course Outline

- Week 1: Course intro, Git workflows, Python project setup
- Week 2: Python OOP, modules, packages, code organization
- Week 3: Test-driven development (pytest), using linters (flake8/black)
- Week 4: MVC and FastAPI basics
- Week 5: Databases & SQL fundamentals
- Week 6: SQLAlchemy ORM and schema design
- Week 7: Authentication & Authorization (JWT, sessions)
- Week 8: Midterm Exam
- Week 9: REST API design and documentation with Swagger/Postman
- Week 10: Docker basics and containerizing Python projects
- Week 11: CI/CD concepts, unit testing pipelines
- Week 12: Capstone project integration (data + API + deployment)
- Week 13: Team collaboration, GitHub branching, pull requests
- Week 14: Project presentations, peer code review
- Week 15: Final Exam

Grading

- **Midterm:** 25%
- **Final:** 25%
- **Homework:** 50% (complete/incomplete)
- **Extra Credit:** 0–3% (Team Collaboration Report)

Grading Scale

- A: 94–100
- B+: 87–93
- B: 80–86
- C+: 74–79
- C: 66–73
- D: 60–65

- F: Below 60

Policies

- **Late Work:** Homework not accepted >2 days late. Projects lose 10% per day, cutoff after 4 days.
- **Incompletes:** Only for serious medical/military cases with documentation.
- **Academic Integrity:** NJIT Honor Code strictly enforced. All commits must be authentic and documented. Work without meaningful Git history will receive zero credit.

Attendance

Students are expected to attend all classes, participate in Slack, and complete assignments on time.

University Policy Statements

- **Academic Integrity:** NJIT Honor Code applies.
- **Disability Support Services:** Students requiring accommodations must contact DSS (Kupfrian Hall 201, dss@njit.edu).
- **Title IX:** NJIT faculty must report gender-based discrimination or harassment to the Title IX Coordinator.
- **Emergency/Contingency Plan:** If in-person meetings are disrupted, course will continue online via Canvas and WebEx/Slack.

Instructional Philosophy

The course emphasizes **professional software development practices**:

- Write tests first (TDD mindset)
- Maintain code quality with linters and style checkers
- Commit regularly with meaningful Git messages
- Treat every project as a **mini-internship simulation**
- Respect the workload: projects may take **40+ hours**
- Goal: graduate ready to function as a **junior developer on internship or entry-level teams**