# IS601: Web Systems Development

## Spring 2025 Syllabus

### Welcome to IS601: Web Systems Development!

Welcome to IS601! This course is designed to equip you with the skills and knowledge necessary to excel in modern web systems development.

#### Key Policies:

- Generative AI: A subscription to ChatGPT or Claude AI is required for course activities.
- **Git Commit Policy**: Follow the established guidelines to track your progress and model professional developer behavior.
- **Academic Integrity**: Strict adherence to original work and proper collaboration documentation is mandatory.
- **Late Submissions**: Homework and projects have specific deadlines with penalties for late submissions.

Let's embark on this journey to build your technical expertise and professional acumen!

## **Course Overview**

#### **Course Information**

- Course Number: IS601
- Course Title: Web Systems Development
- Credit Hours: 3

#### Sections:

- Section 002: Tuesday 10:00 AM 12:50 PM, CKB 303
- Section 004: Thursday 1:00 PM 3:50 PM, CKB 215 (Students must bring their own device for this section)

### Instructor Information

#### **Contact Details**

- Name: Keith Williams
- Office: GITC 3420
- Email: <u>kwilliam@njit.edu</u> (Discord Preferred)

- **Discord**: Primary communication method join link provided in Canvas
- Virtual Office: Zoom Link

Day	Time	Format	Designated Course
Tuesday	1:00 PM - 2:20 PM	In-Person/Discord	IS601
Thursday	4:00 PM - 5:30 PM	In-Person/Discord	IS421
Friday	4:00 PM - 5:30 PM	In-Person/Discord	IS322

### Office Hours (GITC 3420)

Additional Support:

- Available on Discord throughout the week
- Quick questions can be answered anytime on Discord
- Virtual meetings can be scheduled outside office hours by appointment
- Message on Discord before visiting office hours
- All students are welcome during any office hour time slot

## Learning Outcomes

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

- 1. Develop and deploy web applications using Python and the FastAPI framework.
- 2. Implement and interact with various data sources, including CSV files, SQL databases, and RESTful APIs.
- 3. Utilize Git for version control, demonstrating effective commit practices and collaboration.
- 4. Apply object-oriented programming principles and design patterns in project development.
- 5. Containerize applications using Docker and manage full-stack development workflows.
- 6. Implement user authorization and authentication mechanisms and understand the web security model.
- 7. Conduct unit testing and ensure code quality through continuous integration practices.
- 8. Effectively use Generative AI tools to enhance learning and development processes.

## **Course Description**

This comprehensive course immerses students in the world of data programming with Python and web development, focusing on hands-on projects to develop critical problem-solving skills for addressing complex information system requirements. Students will gain proficiency in Python programming, working with various data sources like CSV files, SQL databases, and REST-based web services. The course emphasizes professional competencies, including:

- Industry standards
- Git for version control
- Code standards
- Object-oriented programming
- Design patterns
- Docker
- FastAPI
- SQLAlchemy
- User authorization and authentication
- Web security model

#### Prerequisites

None

## **Generative AI Policy**

Subscription Requirement

- All students must purchase a subscription to either **ChatGPT** or **Claude Al** to access advanced features necessary for coursework.

#### Permitted Use

- 1. **Learning Assistance**: Use GenAl to understand course materials, clarify concepts, and accelerate learning.
- 2. Idea Generation: Brainstorm ideas for projects and assignments.
- 3. **Debugging Help**: Seek assistance in identifying and resolving coding issues.
- 4. **Code Snippets**: Obtain code snippets to understand specific functionalities, but integrate and modify them to fit project requirements.

#### **Prohibited Use**

- 1. **Direct Code Generation**: Do not use GenAl to generate complete code solutions for assignments or projects.
- 2. **Plagiarism**: Any GenAl-generated code or content must be properly cited.
- 3. Academic Dishonesty: Using GenAI to bypass learning objectives is strictly prohibited.

#### Transparency and Ethical Use

- Acknowledgment: Document GenAl use in project documentation or README files.
- **Commit Documentation**: Reflect GenAl usage in Git commits.

- **Respect Privacy**: Do not input sensitive information into GenAI tools.
- **Review and Understand**: Always review and comprehend GenAl-generated content before integration.

## **Git Commit Policy**

#### **Commit Guidelines**

- 1. Commit Frequency:
  - Commit changes frequently
  - Break work into small, manageable commits

#### 2. Commit Messages:

- Clear and descriptive
- Format: <type>: <short description>
- Types: feat, fix, docs, style, refactor, test, chore

### Examples

Feature implementation

git checkout -b feature/user-authentication

git commit -m "feat: add user authentication module"

### Bug fixing

git checkout -b fix/data-parsing-bug

git commit -m "fix: resolve data parsing error in API"

### Documentation update

git checkout -b docs/update-readme

git commit -m "docs: update README with installation instructions"

## Grading

### Grade Breakdown

Category	Percentage
Weekly Hands-on Assignments	50%
Midterm Project	25%
Final Project	25%

### **Grading Scale**

Grade	Range	Significance
А	94-100	Superior
В+	87-93	Excellent
В	80-86	Very Good
C+	74-79	Good
С	66-73	Acceptable
D	60-65	Minimum
F	< 60	Inadequate

### **Grading Policy**

- No grade increases will be considered
- Homework graded complete/incomplete
- Projects graded on:
  - Minimum 50% for substantive effort
  - Additional 50% based on functionality, design, and code quality

## Late Policy

- Homework: No submissions accepted after 2 days late
- **Projects**: 10% penalty per day, maximum 4 days late
- **Exceptions**: Only with medical/military documentation

## Academic Integrity

**Collaboration and Original Work** 

- Original work is required
- Referencing tutorials is allowed; copying is not
- Violations will be reported to the Dean of Students

### **Collaboration Documentation**

- Every commit from another student must be documented in the collaboration report
- Collaboration report must show:
  - Authorization by both students
  - Authentic nature of collaboration
- Clear commit messages and project work history are essential

#### Consequences

- First violation: Project/homework receives 0
- Subsequent violations: Automatic course failure
- Potential disciplinary actions include:
  - Referral to Dean of Students
  - Committee on Professional Conduct review
  - Possible Disciplinary Probation
  - Permanent record marking
  - Possible dismissal
  - Course grade of 'F'

## Course Calendar

Key Dates and Deliverables

Week	Dates	Topics & Deliverables
1	Jan 21-24	Python Environment Setup, Git Review
		**Lab 1**: Development Environment
2	Jan 27-31	Python Fundamentals, Functions, OOP
		**Lab 2**: Python Basics

Week	Dates	Topics & Deliverables
3	Feb 3-7	Data Structures, File Operations
		**Lab 3**: Data Processing
4	Feb 10-14	Database Fundamentals, SQL **Lab 4**: SQL Basics **Proiect 1 Assigned**
5	Feb 17-21	ORM Patterns
		**Lab 5**: ORM Implementation
6	Feb 24-28	FastAPI Basics **Lab 6**: API Development
		**Project 1 Due**
7	Mar 3-7	Advanced API Features
		**Lab 7**: Complex APIs
8	Mar 10-14	Testing Principles **Lab 8**: Unit Testing
		**Midterm Project Assigned**
-	Mar 16-22	Spring Recess
9	Mar 24-28	CI/CD with GitHub Actions
		**Lab 9**: Automation
10	Mar 31-Apr 4	Advanced Python Patterns **Lab 10**: Design Patterns
		**Midterm Project Due**
11	Apr 7-11	Security Best Practices **Lab 11**: API Security

Week	Dates	Topics & Deliverables
		**Final Project Assigned**
12	Apr 14-18	Performance Optimization
		**Lab 12**: Optimization
13	Apr 21-25	Project Workshop
		Code Reviews
14	Apr 28-May 2	Production Deployment
		Final Project Development
15	May 5-7	Final Project Presentations
		**Final Project Due**

### **Important Notes**

- All assignments due at 11:59 PM EST
- Weekly hands-on assignments build towards project requirements
- Lab assignments due at time posted in Canvas
- Project demonstrations during class times

## **Course Resources**

- NJIT Library: Academic resources
- Stack Overflow: Programming Q&A
- GitHub Learning Lab: Git tutorials
- Python Documentation: Official docs
- FastAPI Documentation: Comprehensive guide
- NJIT Tutoring Services: Academic support

## Attendance and Participation

### **Attendance Policy**

- Highly recommended to attend all scheduled classes
- More than three unexcused absences may negatively impact grade

### **Participation Requirements**

- Active engagement in class discussions, group work, and Discord channels
- Contribute to collaborative learning environments

#### **Excused Absences**

- Medical or military absences require documented verification
- Notify instructor in advance when possible

## Accessibility and Accommodations

NJIT is committed to an inclusive learning environment. Students requiring accommodations due to disability should contact the NJIT Disability Resource Center early to discuss needs confidentially.

### Inclusion and Diversity

We value diversity and strive to create an inclusive environment. Discrimination or harassment will not be tolerated. Report any incidents to the instructor or NJIT Office of Diversity and Inclusion.