Course Syllabus: IT202 - Internet Applications

Matt Toegel – matthew.toegel@njit.edu

1. General Information

Course Number:	IT202
Course Title:	Internet Applications
Section(s):	008, 010
Semester:	Spring 2025
Date & Time:	**008:** Monday/Wednesday 01:00 pm - 02:20 pm FMH 207 **010:** Monday/Wednesday 11:30 am - 12:50 pm CKB 222
Modality:	Face-to-Face
Credits:	3
Office Hours:	**CKB Public Area/Lounge Main Floor:** Monday/Wednesday 10 am - 11:20 am General availability on Discord via a provided communication channel

2. Overview

This course will discuss concepts and implementations of a web application covering the frontend, backend, and data layers. Topics will range from markup and styling using HTML5 and CSS3, frontend/client-side scripting using JavaScript and jQuery (with usage of AJAX), backend/server-side scripting using PHP running on Apache, and data storage using MySQL. This course will be heavily hands-on with a milestone approach for a final project. Class participation and questions are strongly encouraged. Git will be heavily used for recording work and submitting assignments on Canvas.

3. Course Catalog

Prerequisites: CS 100 (https://catalog.njit.edu/search/?P=CS%20100) or CS 113 (https://catalog.njit.edu/search/?P=CS%20113) or CS 115 (https://catalog.njit.edu/search/?P=CS%20115) or a course in a high-level programming language as approved by department. This course presents the concepts and software technologies that underline web-oriented, three-tier software architectures and applications. The enabling software mechanism include the markup languages (HTML5 and CSS3) used by browsers, client-side scripting languages and libraries (Javascript and AJAX), web servers and server-side-scripting languages (Apache, PHP, HTTP protocol), and background databases (SQL, MySQL). The course uses a hands-on, guided development approach with substantial assignments to illustrate the fundamental computing concepts systems, and technologies considered and to provide direct experience in their use.

4. Instructor

Matt Toegel
Email: matthew.toegel@njit.edu
Discord: MattToegel

Github: MattToegel

5. Prerequisites

This course is a full stack development course that'll utilize HTML, CSS, JavaScript, PHP, and MySQL. Prior knowledge is beneficial but not required however it's anticipated that students have a reasonable computing background (in addition to the Course Catalog Prerequisites). Students should be familiar with file system structure, running programs from the command line, installing programs, editing text files, etc.

If you are not well versed in these subjects, here are some resources that may help you get up to speed:

GCFGlobal - Basic Computer Skills (https://edu.gcfglobal.org/en/basic-computer-skills/)
Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems (https://www.amazon.com/Debugging-Indispensable-Software-Hardware-Problems/dp/0814474578)
codeacademy - Operating Systems: Filesystems (https://www.codecademy.com/learn/operating-systems-filesystems)
Visual Studio Code - Getting started with Visual Studio Code (https://code.visualstudio.com/docs/introvideos/basics)
simplilearn - What is Client-Server Architecture? Everything You Should Know (https://www.simplilearn.com/what-is-client-server-architecture-article)

6. Attending Class

6.1. Synchronous

Class will be held in the rooms and times given per your schedule from the registrar. Mostly, I'll be sharing my screen with everyone and going over the topics either via the classroom projector or a screen-sharing service. There will commonly be time in class to practice the topic for that day and/or get a headstart on homework. We'll be using Respondus for exams and everyone should ensure the software runs on at least 1 device (anticipate webcams will be required even in the classroom).

It's highly encouraged to ask questions and express any doubts/concerns throughout the course. I want to give everyone the opportunity to raise any concerns or ask any questions to make sure they're on track for the semester. Make sure to always keep in communication with me if there are any concerns about the class or anything related, this can be done via Discord (preferred), email, Canvas Inbox, etc.

7. Learning Outcomes

1. Students can use PHP as a general purpose programming language.
    a. Students can install and run PHP.
    b. Students can design an algorithm to solve a problem.
    c. Students can write and run PHP scripts.
    d. Students can follow standard conventions for software development.
2. Students can utilize PHP to handle user interactions and backend logic on a running web server.
    a. Students can run the built-in php webserver for local development and testing.
    b. Students can run their programs/projects on a publicly accessible web server such as Heroku.
        i. Or any host that supports apache/apache-like web hosting.
    c. Students can connect PHP to MySQL to invoke SQL CRUD operations.
3. Students can utilize CSS to apply styles/theming to web pages.

    a. Students can utilize CSS libraries/frameworks such as Bootstrap 5.
    b. Students can make mobile-friendly and cross-platform designs.
4. Students can structure/layout pages via HTML5 tags/semantics.
    a. Students can structure valid HTML specification documents.
    b. Students can combine language content into the same document.
        i. Students can determine which languages will be interpreted based on the file extension.
    c. Students can reference other files (include/require).
5. Students can utilize MySQL for persistent data storage and retrieval.
    a. Students can remotely connect to a MySQL instance via their IDE, development tools, and hosting platform.
    b. Students can construct SQL queries to create tables.
    c. Students can construct SQL queries to read data from tables.
        i. Students can construct complex SQL queries such as JOINs.
    d. Students can construct SQL queries to add rows/records to tables.
    e. Students can construct SQL queries to update rows/records to tables.
    f. Students can construct SQL queries to delete rows/records from tables.
6. Students can modularize their code to make reusable components to aid development.
7. Students can problem solve and debug.
    a. Students can utilize the error output/logs and other debugging techniques.
    b. Students can formulate their own code to solve complex problems.
    c. Students can analyze, interpret, and integrate course-provided code.
8. Students can utilize version control.
    a. Students can use git commands via the terminal/command line.
    b. Students can use GitHub for managing histories of their work.
9. Students can gain experience with a milestone approach of building a complex project.
    a. Students can incrementally develop a large project in a sprint-like structure (based on provided guidelines).
    b. Students can extend existing code to add new features.
    c. Students can add new files to solve requirements.
    d. Students can leverage course concepts effectively.

8. Illustrative Schedule

The schedule is a guideline and is subject to change to fit the particular instance of the class. All topics in general are planned to be covered. Some may have more focus than others and per class interest, other topics may be included.
Note: Some modules may span more than one week, but in general they'll be about 1 week in length and extra time later in the semester will go towards Project Topics and Questions.

Each Milestone will generally have 2 weeks to work on the specific requirements.

Module	Topics
Module 1:	• Overview/Introduction of Course • Environment Setup (VS Code/PHP/Heroku) • Git/GitHub Intro
Module 2:	• PHP and JavaScript Intro • Data Types • Flow Control • Loops and Iterations
Module 3:	• HTML Intro • Forms • JavaScript/CSS Selectors
Module 4:	• SQL Basics • Project and SQL Setup • User Authorization (Registration) • Navigation with dynamic state changes • User Authorization (Login) • PHP Templating / Conditional HTML Concepts • **[Project Data Source Chosen]** RapidAPI.com (https://rapidapi.com/)
Module 5:	• Regex Intro • Flash Messages and User feedback • Alternative Login (username) • User Profiles • **[Midterm]**
Module 6:	• Authorization (User Roles) • Login Enhancement • Dynamic Content - Data Driven Development • **[Project Milestone 1 Introduced]**
Module 7:	• Ajax Intro • Bootstrap • Reusability • Working with APIs

Module	Topics

Module 8:	• jQuery Intro and Comparison
	• Using API data via PHP
	• CRUD Operations of Core Project Data
	  ◦ API Data
	  ◦ Custom User entered data
	• **[Project Milestone 2 Introduced]**

Module 9:	• Entity Associations (API/Custom data to Users) • Milestone 3 Concepts • **[Project Milestone 3 Introduced]**
Module 10:	• Other data associations and analytics • Public User Profiles • Automatic fetching/refreshing of API data • Continued Milestone 3 Concepts
End of Semester:	• Final Milestone Due (deliverable) • Final Demo of Project Due (presentation)

8.1. Schedule Notes

Although some modules aim to isolate certain concepts, due to the nature of the concepts being fullstack there will commonly be usage of most of the noted programming languages in later Modules.

When applicable, security and best practices will be discussed in each Module.

9. Assignments

Each week there will be coding samples related to the current week's topics. Additionally, there will be supplemental online resources as well as recordings available to support your learning.
Assignments generally are graded out of 10 points unless otherwise noted.

9.1. Semester-Long Project

There will be a semester-long project that each student will incrementally develop as new topics are learned. A set of requirements/objectives will be given via a Proposal document at the start of the semester.

The project will be based on an agreed-upon proposal and will cover the material discussed in class. During the semester, there will be milestone deliverables for groups of features from the project. These milestones will cover the gist of the features; there commonly is some time between the last milestone and the final demo/deliverable where the remaining features can be implemented and/or cleaned up.
Milestones are graded out of 10 points.

Students are expected to utilize the course material and structure to work on the projects as this aids/eases learning/debugging and enhances perspective for anyone already familiar with a particular design/structure.

9.2. Quizzes

There will be weekly online quizzes that'll test the current module's topics. These will be taken via Canvas online by the designated due date (typically the following Monday night of the particular module). Each quiz requires a passphrase, generally has one attempt unless otherwise noted, and will generally be capped at 10 minutes. Scoring is out of 10 points. These are expected to be closed notes and taken individually. Part of the goal is to highlight the topics that require extra review/study as they'll come up again in the semester.

9.3. Exams

All exams will use Respondus, so be sure to bring a compatible device with you on the day of the assessment. The midterm will take place during a regular class period and will cover the material from modules one to four. Exams will be closed book and must be taken in the classroom if the class is meeting face-to-face.
Any exams will be graded out of 100 points.

10. Grading

10.1. Breakdown

Midterm: 20%
Quizzes: 15%
Participation/Attendance: 5%
Assignments: 10%
Milestones (1-2): 25%
Final Project Deliverable: 25% (Includes last milestone and remaining features)
Final Demo: Included in final project deliverable

All points will be converted to a final percentage and letter grade at the end of the semester. Canvas will already have the weights applied.

10.2. Extra credit

Extra credit may be given for exceptional programming projects at the discretion of the instructor.

10.3. Grading Scale

Grade	Percentage Range
A	100% to 89.5%
B+	<89.5% to 84.5%
B	<84.5% to 79.5%
C+	<79.5% to 74.5%
C	<74.5% to 69.5%
D+	<69.5% to 64.5%

Grade	Percentage Range
D	<64.5% to 59.5%
F	Below 59.5%

11. Materials/Technologies

1. To better match industry trends and standards, this class will utilize online resources in lieu of a traditional textbook. Resources will be provided as the course progresses, but some primary resources are listed here:
    a. w3schools - PHP Tutorial (https://www.w3schools.com/php/)
    b. w3schools - CSS Tutorial (https://www.w3schools.com/css/)
    c. w3schools - JavaScript Tutorial (https://www.w3schools.com/js/)
    d. w3schools - MySQL Tutorial (https://www.w3schools.com/mysql/)
2. This class utilizes Canvas learning management system. You can find assignments, assessments, and learning resources there.
3. This class utilizes a platform for assignment evidence gathering and submission (provides clear objectives and generates a file to upload to Canvas for review).
    a. Students will access the worksheets through Canvas to fill in the required deliverables and the final submission will be uploaded to Canvas.
    b. The worksheets aim to maintain clarity of requirements for the student and for objective and fair grading.
4. This class will utilize the Respondus proctoring system for any/all exams.
5. Students are required to have a device that meets the YWCC minimum specifications (https://ist.njit.edu/student-computers-recommended-specs). They will need administrative acess to this device to install software. The device should be functional (charged, working) and brought to each class.
6. Students will utilize an installed IDE, (VS Code (https://code.visualstudio.com/Download) is recommended and will be used by the instructor).
    a. Students may use another IDE at their discretion as long as it supports the necessary structure that'll be utilized in addition to any required extensions.
7. This class has a GitHub page (https://github.com/MattToegel/IT202-2025) where you can find out how the course materials are built and have changed overtime.
8. Students will install and utilize the latest PHP 8.4 (https://www.php.net/downloads.php).
9. Students will utilize Heroku to host a dev/qa version of their content and a production version of their hosted content.
    a. The dev/qa is for testing/verifying that things work outside of localhost and is meant to be more of a sandbox before production.
    b. The production instance is what will be graded against and should always be working.
    c. With the GitHub Student Developer pack and getting the the Heroku Student Offer (https://www.heroku.com/github-students) students will receive over $300 of credit for 2 years.
    d. The course will be utilizing the Eco Tier which is **$5 for 1000 VM hours/month**.
        i. Eco Tier VMs sleep after 30 minutes of inactivity giving students a larger pool of resources than what appears on paper. The 1000 hours is shared across all VMs on the same account.

ii. A remote MySQL connection will be provided for each student to utilize across their environments.

10. Students will utilize git and github for tracking development progress

    a. The first module's lessons will be dedicated to the setup of these resources and any naming conventions to follow

    b. git for windows (https://git-scm.com/download/win)

    c. git for mac (https://git-scm.com/download/mac) (Homebrew is recommended)

    d. git for linux (https://git-scm.com/download/linux)

    e. Github Education (https://education.github.com/discount_requests/application) (the education package is recommended especially to get free hosting credits, but a regular free account is sufficient)

12. Policies

12.1. Academic Integrity

The work done is expected to be your own, any group work should clearly distinguish ownership of tasks. Use of snippets/material from others should be kept to a minimum and the source should be accredited where applicable.

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: academic integrity code (http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf).

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing, or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu.

12.2. Requesting Accomodations

If you are in need of accommodations due to a disability please contact the Office of Accessibility Resources & Services (OARS) (https://www.njit.edu/studentsuccess/accessibility), Fenster Hall Room 260 to discuss your specific needs. A Letter of Accommodation Eligibility from the OARS authorizing your accommodations will be required.

12.3. Resources for NJIT Students

NJIT Services for Students (https://docs.google.com/document/d/1xGO2qcVEF1tsOgZn-_W1LjSOKn_jhEVs9IWI_6jeuPs/edit?usp=sharing), including Technical Support

12.4. Class Etiquette

Students who are the most successful attend and participate in class. If you have questions, please ask them. This makes the class more dynamic and interesting for everyone.

12.5. Proctoring

NJIT policy requires that all midterm and final exams must be proctored, regardless of delivery mode, in order to increase academic integrity. Note that this does not apply to essay or authentic based assessments. Effective beginning Fall semester 2019, students registered for a fully online course section (e.g., online or Hyflex mode) must be given the option to take their exam in a completely online format, with appropriate proctoring.

Exams will be given in-person using Respondus. Be sure to bring your charged laptop and charger on the day of exams.

12.6. Late Policy

All deliverables will be eligible for a 5% penalty per day late, generally Canvas automatically has this set and will automatically apply it as grades are entered. Late assignments will automatically be marked by Canvas as a 0, but will be updated once grades are entered for the particular item.

Missed Exams/Quizzes will result in a 0.

If you are going to miss a class/material and cannot hand in an assignment, it's your responsibility to let me know as soon as possible so the situation can be handled.
There also will be no make-up exams (except, at the discretion of the instructor in the case of a documented medical or family emergency from the Dean of Students).
For any emergency please reach out to the **Dean of Students** (https://www.njit.edu/dos/student-absence-verification) so they can send out an official notice.

12.7. Attendance Policy

It's anticipated that students attend each scheduled session. Attendance will be recorded each class.

12.8. AI Usage Policy

The expectation of this course is for students to work through the course without assistance from any type of artificial intelligence to better develop their own skills in this content area. While, artificial intelligence (AI) is permitted in this course as a tool/aid, it is not permitted to be used to blindly complete assignments and/or generate full "solutions" in place of your own analysis and implementation. Additionally, if and when students use artificial intelligence in this course, the AI **must** be cited as is shown within the NJIT Library AI citation page for AI (https://researchguides.njit.edu/AI/cite). If and when used, failure to clearly disclose the usage of AI within the constraints of this policy will result in a zero for the entire assignment. Generally if the course contains code, there is an expected style, library, toolkit, etc (course material) provided that must be utilized.

13. Closing Notes

Syllabus is subject to change, attend class to stay current.

Last updated 2025-02-05 02:26:32 UTC