

Course Syllabus

Course Title

Web Development Using Application Frameworks

Course Number

IS439 – A-Section

Semester

Spring 2024

Instructor

Kevin Trainor

Teaching Assistant

Colleen Heinemann

Class Times

Online meetings will be held on Tuesdays from 12:30 PM till 2:30 PM in CIF 1038. I look forward to seeing you there.

Optional Lab Sessions

I hold optional lab sessions two times per week. Please join me to ask a question, to discuss solutions to previous assignments, to get help with the current assignment, to discuss the final project, or just to say hello. I will work with students on a first come, first served basis.

The first optional lab session of the week takes place immediately following our regular class session. Since class sessions typically end earlier than their scheduled two-hours, I devote the remainder of the time available to helping students. Often, I can stay beyond the two-hour period if many students need help.

On Friday afternoons from 3:00 PM till 4:00 PM, I will be holding an optional online lab session using Zoom. I look forward to you joining us. Please use a headset.

Office Hours

I do not hold conventional open office hours. Students needing help with assignments are encouraged to join in the optional lab sessions (see above). Students who need to discuss confidential matters can arrange for an individual meeting with me on Zoom. If you need to arrange an individual meeting, please send me an *Individual Meeting Request* via the HelpDesk for this course (see *Contacting Instructor or TA* below).

Contacting Instructor or TA

Please do not send requests or other messages to us using Canvas or our regular email addresses. We will not be responding to messages sent to us on those platforms.

Instead, please contact us by creating a ticket using the [Trainor HelpDesk](#). The HelpDesk system allows you to open a ticket and correspond back-and-forth with us by sending and receiving emails to/from the Trainor HelpDesk system.

For an introduction to the HelpDesk system, please see [About Trainor HelpDesk](#).

For assistance in creating a ticket related to this course (IS439 A-Section), please see [IS439 A-Section - Compose Email](#).

Catalog Course Description

A course in the use and evaluation of back-end Web application frameworks for system architects, designers, and developers.

Detailed Course Description

Today, many substantial Web applications are designed and built using a back-end Web application framework. These frameworks provide standard ways to build and deploy Web applications that can increase developer productivity, increase software quality, and reduce application maintenance. Back-end Web frameworks typically use architectural patterns like Model-View-Controller (MVC) to separate code that implements the data model, business rules, and user interface. Frequently, they also employ architectural patterns like a database persistence layer to greatly simplify database coding within the Web application. Other common features of back-end Web application frameworks include built-in support for Web page templating, user authentication/authorization, content caching, URL mapping, session management, Web services, and various content management features.

This course will explore the common features of back-end Web application frameworks in general while providing hands-on experience using Django, a popular open-source Web application framework based on Python. Students will complete a series of coding assignments that build Django design/development/deployment skills while providing insights into standard features of back-end Web application frameworks. In the final project, each student will design and build a full-featured Django Web application to address a problem of their own choosing.

The audience for this course includes system architects, designers, and developers who wish to consider a back-end Web application framework as a platform for creating substantial Web applications. These include business applications, digital libraries, institutional repositories, and research data repositories, as well as workflow applications for data collection, data extraction, data cleaning, and data analysis.

Prerequisites

- Experience in creating static Web sites using HTML and CSS
- Experience in Python programming (IS430, IS597-PR, or equivalent)
- Experience in creating dynamic Web sites using tools like PHP is helpful but not required.
- Experience in using relational databases is helpful but not required.

Course Outcomes

After completing this course, you should be able to:

1. Identify the advantages of using a back-end Web application framework when designing, developing, and deploying a Web application.
2. Identify common features of back-end Web application frameworks in general.
3. Explain how the specific features of Django correlate with common back-end Web application framework features.
4. Design, code, test, and deploy Web applications that use Django features for:
 - a. Models
 - b. Templates
 - c. URL Mapping
 - d. Views
 - e. Forms
 - f. User authentication/authorization
 - g. Deployment
5. Design, code, test, and deploy a full-featured Django application to solve a problem of your own choosing.
6. Evaluate the suitability of a particular back-end Web application framework based upon an anticipated set of functional and non-functional requirements.

Required Texts

Vincent, W. S. (2023). *Django for Beginners: Build websites with Python and Django (Revised for Django 4.2)*.

This book is self-published:

- An e-book (EPUB or PDF) can be purchased at the author's site: <https://wsvincent.com/books/>
- A print (or Kindle) version can be purchased from Amazon: <https://www.amazon.com/dp/1735467200/?tag=wsvincent-20>

Pinkham, A. (2016). *Django Unleashed*. Pearson Education.
Print ISBN: 9780321985071, eBook ISBN: 9780133812398
A free electronic copy of this book is available to Illinois students at: <https://www.oreilly.com/>

Oppel, A. (2010). *Data Modeling: A Beginner's Guide*. McGraw-Hill.
ISBN-10: 0071623981, ISBN-13: 978-0071623988
A free electronic copy of this book is available to Illinois students at: <https://www.oreilly.com/>

Chacon, S. & Straub, B. (2009). *Pro Git (2nd Edition)*. Apress.
A free electronic copy of this book is available at <https://git-scm.com/book/en/v2>
A free electronic copy of this book is available to Illinois students at: <https://www.oreilly.com/>

Recommended Texts

Myers, G. J., Badgett, T., & Sandler, C. (2012). *The art of software testing: Now covers testing for usability, smartphone apps, and agile development environments* (3. ed). Wiley.
A free electronic copy of this book is available to Illinois students at: <https://www.oreilly.com/>

Okken, B. (2022). *Python Testing with pytest*. Pragmatic Bookshelf.
A free electronic copy of this book is available to Illinois students at: <https://www.oreilly.com/>

Percival, H. (2024). *Test-Driven Development with Python, 3rd Edition [Early Release]*. O'Reilly Media.
A free electronic copy of this book is available to Illinois students at: <https://www.oreilly.com/>

Pajankar, A. (2021). *Python Unit Test Automation: Automate, organize, and execute Unit tests in Python*. Apress.
A free electronic copy of this book is available to Illinois students at: <https://www.oreilly.com/>

Technology Requirements

You will be completing coding assignments and conducting your final project using your own computer. I recommend that you use a computer that runs Windows (either Windows 10 or Windows 11) or a recent release of macOS. While the software that we will be using for this course does run on Linux, there will be substantially less technical support available for Linux. If you want to use a Linux computer for your coursework, please contact me first.

You will be installing the following software on your computers for this class:

- The Anaconda open data science platform for Python
- The PyCharm Professional integrated development environment (IDE) for Python
- The Git version control system
- The GitHub Desktop client for Git

All of the software and services that you will be using are free for your use during this class. I will publish brief instructions and detailed tutorial videos to help you download, install, and begin using the required software on computers running Windows (either Windows 10 or Windows 11) and recent releases of MacOS. Links to these resources will be provided in our Weekly Schedule.

The tools described above are those that I recommend for your use when completing work for this course. If you are not able to use one of these tools because of accessibility reasons, please contact me to get approval for a suitable alternative that meets your needs as a student and our needs as graders.

Course Schedule

The schedule for this course will be available via our Weekly Schedule at:

https://courseinfo.ligent.net/2024sp/_illinois/is439_a_section/index.html

The course schedule is always subject to reasonable change to account for changes in circumstances and to correct errors. When I make changes to the schedule, I will announce them via our Canvas Announcements Forum. Postings to this forum should result in you being sent an email copy of the announcement as well.

Course Elements

1. *Readings*

Required readings will be assigned from the resources listed above and from other resources that will be identified in the Weekly Schedule. Generally, readings are chosen to accompany our online lecture/discussion session for the week. So, you should expect to complete the readings before class.

Several optional textbooks have been identified for this course that deal with software testing in general, unit test frameworks for Python, unit testing for Django, and test-driven development. Much of our in-class breakout group work will involve unit testing and the challenge portion of many assignments will also involve unit testing. While there will be no fixed list of reading assignments in these texts, our discussions in class will include students making recommendations to the class regarding which of these testing resources they are finding most helpful.

2. *Lectures and Lecture Videos*

Some of the lectures for this course will be provided as video recordings. I will present others live during our class session. When a video lecture recording has been provided, I will expect you to have played the recording before our weekly class session. Before playing my lecture videos, make sure that you have previously played [Tips on Playing My YouTube Videos](#) to assure that you get the most from your viewing experience.

3. *Tutorial Videos*

I have created tutorial videos for the following use cases:

- There are a number of tutorials that explain how to do activities necessary for the course. These include installing software, reading grading rubrics, submitting assignments, and related activities.
- Many tutorials are coordinated with the exercises in your coding assignments. They represent a demonstration of the skill you will be expected to use when doing the exercise.

I recommend that you play these recordings either before or during your work on the assignment with which they are associated. Before playing my tutorial videos, make sure that you have previously played [Tips on Playing My YouTube Videos](#) to assure that you get the most from your viewing experience.

4. *Coding Assignments*

There will be weekly *Coding Assignments*. As mentioned above, coding assignments will often be paired with tutorial videos. These should allow you to complete your Coding Assignment using the same general approach that has been demonstrated in the tutorial video.

A Canvas submission activity will be provided for submitting each assignment. Instructions for each Coding Assignment and a grading rubric will be published in the Weekly Schedule.

Solutions to the regular exercises from the Coding Assignments will be posted to our Canvas site immediately before our next online class session. At the beginning of each class, we will review the solution to the most recent assignment, and I will entertain any questions that students may have regarding the assignment.

A major goal for this course is to build your proficiency in self-evaluation of your work. To build this skill, I will expect you to be able to estimate your grade on each Coding Assignment. The solutions to coding assignments posted to our Canvas site and our review of those solutions during the next class will serve as your primary feedback for the Coding Assignment.

As secondary feedback, your Coding Assignment submissions will be graded and commented upon. This feedback will be published to the Canvas assignment submission activity within 2 weeks.

For information regarding the grading of coding assignments, please see *Coding Assignment Submissions That Meet Certain Criteria Are Subject to a Minimum Score Guarantee* under *Grading Policies* below.

5. *Final Project*

You will be expected to plan, gather requirements for, design, code, and test a Web application using Django as your Final Project. Detailed instructions and grading rubric for the Final Project will be published separately. The following are highlights from those requirements:

- The Web application should fully demonstrate the back-end Web application framework features covered in the class.
- The Web application should include significant add/change/delete database functionality.
- The Web application should be sufficiently interesting to you that you are likely to continue to develop and maintain after the course is complete.

The Final Project is an individual assignment, NOT a group assignment.

6. *Attendance*

The iSchool expects students to attend all classes except in cases of emergency. See *Student Code on Attendance*: <http://studentcode.illinois.edu/article1/part5/1-501/>. Students in online classes who miss a class meeting are expected to play the recording of that class meeting. If you need help locating the recording of a class, please contact the iSchool Help Desk.

7. *Participation*

Your participation is an important element of the course. Accordingly, a significant portion of your grade for this course will be determined by your participation. Students will earn participation credit for:

- Introducing yourself by speaking during the first class meeting.
- Making speaking contributions during class.
- Actively participating in a group breakout session during class.
- Speaking for your group during debrief of group breakout session during class.

For information regarding the grading of participation, please see *Your Participation Grade Will Be Based Upon Participation Points Earned Throughout the Semester* under *Grading Policies* below.

8. *Framework Evaluation Paper*

Your paper will present the conclusions of your evaluation of the suitability of a candidate server-side Web application framework (other than Django) for a set of functional and non-functional requirements which you could reasonably expect to encounter in the workplace. The length of your paper should be 1500 to 2000 words. In addition, you will need to cite a minimum of 10 sources. Detailed instructions and a grading rubric for the Framework Evaluation Paper will be published separately.

Course Grading Policies

1. *iSchool and University Grading Policies Apply*

Many iSchool and University Academic Policies have grade implications. Please see *iSchool and University Academic Policies* below.

2. *Careful Attention to Detail is Required*

One important goal of this course is to train you to become a responsible information professional. The work of information professionals is highly detail oriented. Clients rely on information professionals to deliver a correct work product that conforms to stated requirements and best practices.

When your work is graded, deductions will be made for all deviations from the assignment instructions. Some of these deductions will be made for small deviations that may seem insignificant to you. So, it is a good practice to carefully check your work against all instructions before submitting.

3. *Assignment Resubmissions are Not Permitted After the Assignment Deadline*

While assignments may be resubmitted before the assignment deadline, they may not be resubmitted after the assignment deadline. If you should accidentally make a submission that is somehow defective, you must discover and correct this error before the assignment deadline. Resubmission of assignments after the deadline will not be permitted for any reason.

4. *Deadline Extensions Must be Requested Before the Deadline*
If you believe that you have a valid reason for a deadline extension, please submit a *Deadline Extension Request* using the Service Desk before the deadline. I have a practice of granting reasonable extension requests. I will only grant extensions beyond the beginning of our next class session in very limited circumstances.
5. *Deductions Will be Made for Late Submissions*
The grading rubrics for coding assignments, the Framework Evaluation Paper, and the Final Project include substantial deductions for late submissions. Please see the assignment grading rubrics for more details.
6. *Assignments Submitted Too Late Will Not be Graded*
Coding assignments submitted more than 14 days late will be considered *too late*. Framework Evaluation Papers and Final Projects submitted more than 7 days late will be considered *too late*. Assignments that are submitted *too late* will not be graded. These submissions will earn a grade of zero. If you are in danger of missing the *too late* deadline, and you believe that you have a valid reason for an extension, please submit a *Deadline Extension Request* using the Service Desk before the deadline.
7. *Grade Adjustments Will Be Limited to Automatic Rounding*
All grades will be awarded on 0 to 100-point scale. Fractional values will be rounded automatically. Fractional portions of grades ending in .0 through .4 will be rounded down. Fractional portions of grades ending in .5 through .9 will be rounded up.

No further adjustments will be made to grades. This policy applies even in situations where increasing a grade by just 1 point would cause a student's final letter grade for the course to cross a threshold (i.e. from B+ to A-). Regardless of the potential consequences, grade adjustments will be limited to automatic rounding.
8. *Re-Grading Requests Made Using the HelpDesk Will be Given Fair Consideration*
It is possible for one of your assignment submissions to be missed during the grading process. This is especially true for assignments that are submitted late. If this happens to you, please submit a Re-Grading Request to the HelpDesk to remind us that your submission still needs grading. Make sure to fully identify the assignment that needs attention.

Each assignment that we grade is accompanied by a grading feedback form. Please read this feedback to understand our grading decisions. If, after reading the grading feedback form, you believe that our grading decisions are somehow unfair, please submit a Re-Grading Request to the HelpDesk. Include details in your request that identify the assignment and your rationale for the re-grading request. We will give these requests fair consideration and inform you of our determination by posting back to the HelpDesk ticket.
9. *Extra Credit Opportunities are Not Available*
I have designed the grading policies for coding assignments such that poor performance on one coding assignment should not spoil your entire semester grade (see Item 11, below). Consequently, I do not offer any opportunities to submit work for extra credit.

10. *The Expected Grade for Assignment Submissions That Meet All Expectations is 95*
The grading rubrics for all assignments have been designed such that submissions that meet all expectations for the *Regular Exercises* will score 95 points.

Assignments will also include a *Challenge Exercise*. Students who choose to submit the Challenge Exercise may earn from 0 to 5 additional points. Challenge Exercises are designed to be quite difficult and the amount of help available from the instructor for Challenge Exercises will be limited to hints.

11. *Coding Assignment Submissions that Meet Certain Criteria are Subject to a Minimum Score Guarantee*

The grading rubric for Coding Assignments has been designed to promote two important behaviors:

- Submitting your work in a properly named and formatted file. This helps substantially with grading workflow.
- Submitting your work by the assignment deadline. This assures that you will get the benefit of having tried to solve the problem on your own before seeing the solutions of others.

While separate grading rubric and assignment submission instructions documents will be published, the following is a summary of the coding assignment grading rubric features:

- 10 points will be awarded for submitting a single, properly named and properly formatted file to the proper Canvas assignment submission activity.
- A minimum of 75 points will be awarded for submissions that are submitted on time, and that demonstrate a good faith effort on all parts of the assignment. Late submissions will be awarded 74 points or fewer in this category.

The implication of this grading scheme is that you can expect a score of 85 or higher on all coding assignment submissions that meet both criteria.

12. *Your Participation Grade Will Be Based Upon Participation Points Earned Throughout the Semester*

The table below lists activities for which you may earn participation points and the points earned for each instance.

Activity	Points Earned
Introducing yourself by speaking during the first class session	10
Speaking contributions during class	2
Actively participating in a group breakout session during class	5
Speaking for your group during debrief of group breakout session during class	2

Your participation grade for the course will be calculated at the end of the semester based upon the total number of participation points earned. Participation grades will be calculated using a curve. The details of this curve-based grading are presented below:

Choice of Grading Scheme

Participation Points Earned	Applicable Grading Scheme
≥ 51	Percentile-Based Scheme
≤ 50	Low-Participation Scheme

Percentile-Based Scheme

Participation Points Percentile	Participation Grade
100	100
75	95
50	90
25	80
0	75

Low-Participation Scheme

Participation Points Earned	Participation Grade
50	50
35	35
25	25
<= 24	0

13. Attendance at Class Sessions May Affect Your Grade

While attendance is not graded directly, it may have a significant impact on your participation grade. All activities that earn participation credit occur during class.

Academic Integrity Policies and Practices Specific to This Course

There are iSchool and University policies regarding academic integrity. Please see the *iSchool Academic Integrity Statement* under *iSchool and University Academic Policies* (below).

Designing, coding, and testing your own programs is the primary learning activity in this course. Most of the learning value of these activities is the result of doing them on your own. Coding your programs independently leads to expected differences between the program files submitted by different students. These differences are not only expected, but they are often celebrated, as we explore the boundaries of good programming style and good programming practice.

The program files that you submit as your own work in this course will be compared to those submitted by other students in your section and by students in other sections. Your submitted work will also be compared to the official class solution as well as solutions generated by AI tools such as ChatGPT. Instances of high correlation between the work that you submit, and the work of others will be investigated as possible instances of plagiarism or cheating. Both plagiarism and cheating are serious academic integrity infractions. See <https://studentcode.illinois.edu/article1/part4/1-402/>.

The submission of highly correlated work represents an academic integrity violation for students who copy the work of others, for students whose work is copied by others, and for students who create and submit a collaborative work when independent work is expected. So, please do your work independently. Never put yourself in a position where your work could be submitted by another student.

Our work in checking your submissions for academic integrity violations will continue throughout the course. File comparison is a time-consuming activity and late submissions create new opportunities to discover highly correlated work. So, investigations can be initiated at any time during the course – even after an assignment has been graded.

The only exceptions to this general prohibition on submission of highly correlated work come from two situations where code has been provided to you in course materials:

1. *Starter Files*

As part of coding assignments, I sometimes give you starter files. Typically, these are test data files. But, sometimes, these are programs that I expect you to include in your project before submission. You are permitted to submit any programs that I provide to you as starter files without further modification.

2. *Tutorial Code*

This course includes many coding assignments that are based on tutorials where you are expected to follow-the-leader. This includes tutorials from the Vincent text and my own recorded video tutorials. You are permitted to copy and adapt code from these tutorials to create your submission. For the best learning experience, I suggest that you type the code into your computer yourself. Nevertheless, if you have an electronic copy of the Vincent text, you may copy and paste text from that electronic copy. Apart from this very narrow set of circumstances, you are not permitted to copy text that has been created by or typed into the computer by a source other than yourself.

Basis for Determining Grade

The various components of student work will contribute to the final grade based upon the following percentages:

- Participation 15%
- Coding Assignments 35%
- Framework Evaluation Paper 20%
- Final Project 30%

Letter grades will be determined as follows:

- A+: 97 - 100%;
- A : 93 - 96%;
- A- : 90 - 92%;
- B+: 87 - 89%;
- B : 83 - 86%;
- B- : 80 - 82%;
- C+: 77 - 79%;
- C : 73 - 76%;
- C- : 70 - 72%;
- D+: 67 - 69%;
- D : 63 - 66%;
- D- : 60 - 62%;
- F : 0 - 59%;

Please note that when converting overall course number grades to letter grades, simple rounding of number grades will be used. Please see *Grade Adjustments Will Be Limited to Automatic Rounding* under *Grading Policies* above.

ISCHOOL AND UNIVERSITY ACADEMIC POLICIES

Incomplete Grades

An exceptional request for an incomplete grade is most often granted to students encountering a medical emergency or other extraordinary circumstances beyond their control. Students must request an incomplete grade from the instructor. The instructor and student will agree on a due date for completion of coursework. The student must submit an Incomplete Form signed by the student, the instructor, and the student's academic advisor to the front office:

<https://uofi.app.box.com/s/sx7arobhr0gfw12teaetmp1qq32ifdrd>

Please see the Student Code for full details:

<http://studentcode.illinois.edu/article3/part1/3-104/>

iSchool Academic Integrity Statement

The iSchool has the responsibility for maintaining academic integrity so as to protect the quality of education and research in our school and to protect those who depend on our integrity. Consequences of academic integrity infractions may be serious, ranging from a written warning to a failing grade for the course or dismissal from the University.

See the student code for academic integrity requirements:

<http://studentcode.illinois.edu/article1/part4/1-401/>

Statement of Inclusion

<https://diversity.illinois.edu/about/senate-diversity-resolution/>

As the state's premier public university, the University of Illinois at Urbana-Champaign's core mission is to serve the interests of the diverse people of the state of Illinois and beyond. The institution thus values inclusion and a pluralistic learning and research environment, one which we respect the varied perspectives and lived experiences of a diverse community and global workforce. We support diversity of worldviews, histories, and cultural knowledge across a range of social groups including race, ethnicity, gender identity, sexual orientation, abilities, economic class, religion, and their intersections.

Religious Observances

In keeping with our Statement of Inclusion and Illinois law, the University is required to reasonably accommodate its students' religious beliefs, observances, and practices in regard to admissions, class attendance, and the scheduling of examinations and work requirements.

If you anticipate the need for an accommodation, please communicate with your instructor in the first two weeks of class. If you are an undergraduate student and your instructor requires an absence letter, you must fill out the Religious Observance Accommodation Request form:

https://cm.maxient.com/reportingform.php?UnivofIllinois&layout_id=19 . Other accommodations may be available.

Accessibility Statement

To obtain accessibility-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and Educational Services (DRES) as soon as possible. To contact DRES you may visit 1207 S. Oak St., Champaign, call 333-4603 (V/TTY), or e-mail a message to disability@uiuc.edu .

iSchool COVID-19 Statement

In keeping with university and iSchool policy, all students are required to engage in appropriate behavior to protect the health and safety of our community. If you are on campus, this includes being fully vaccinated, wearing a facial covering properly when required, maintaining social distance, if requested, and using hand sanitizer as needed.

If you feel ill or are unable to come to class or complete class assignments due to issues related to COVID-19, including but not limited to testing positive yourself, feeling ill, caring for a family member with COVID-19, or having unexpected child-care obligations, you should contact the instructor immediately and cc your advisor.

Contact Hours

This course will require approximately 54 contact hours.

Last Revised

2024-01-03