Course Syllabus
University of Wisconsin – Milwaukee (UWM)
School of Information Studies (SOIS)

Course Title
Advanced Web Development Using Application Frameworks

Course Number
INFOST 691

Credits
This is a 3-credit course.

Meeting Times and Location
Online

Instructor
Kevin Trainor
Office: NWQB-3472

Contacting the Instructor
The preferred method for contacting Kevin is by visiting the Request Center portal for this course at https://askproftrainor.atlassian.net/servicedesk/customer/portal/9 . Implemented using JIRA Service Desk, this portal will allow you to submit questions and requests related to this course. I anticipate that using this portal instead of regular email will help me to respond to your requests more quickly and to track them more effectively. PLEASE, DO NOT send requests to my regular email address.

In the event that the portal is unavailable to you, you may email your request to the Request Center at the following address:
infost691advancedwebdev@askproftrainor.atlassian.net .

On an emergency basis, you may contact me using my mobile phone number:
847-650-9706.

Online Office Hours / Lab Sessions
Every Thursday evening starting at 7:30 PM, I will be holding an Office Hours/Lab Session using the GoToMeeting Platform. Please join us to review solutions to recent assignments, to ask a question, to get help with your assignments, to discuss your plans for the final project, or just to say hello. Please use a headset when joining this session. You may join the GoToMeeting session using this link.

Catalog Course Description
An advanced course in the use and evaluation of Web application frameworks for system architects, designers, and developers.

Detailed Course Description
Today, all substantial Web applications are designed and built using a Web application framework. These frameworks provide standard ways to build and deploy Web applications that increase developer productivity, increase software quality, and reduce application maintenance. 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 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 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 practice assignments that build Django design/development/deployment skills while providing insights into standard features of Web application frameworks. In the final project, each student will design and build a full-featured Django Web application to address a problem of her/his own choosing.

The audience for this course includes system architects, designers, and developers who intend to create 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
Graduate standing or senior undergraduate standing. For 500 and 600 level courses it is recommended that an undergraduate student first consult with the instructor and/or advisor concerning the applicability of this specific course.

Further prerequisites include:

- Prior experience in Python programming (COMPSCI 702, HCA 741, or equivalent).
- Prior experience in basic Web development using HTML/CSS (INFOST 240, INFOST 685 or equivalent)

Prior experience with relational databases is **helpful, but not required.**

For undergraduates wishing to take this course, it is recommended that they have successfully completed COMPSCI 481.

Course Outcomes
After completing this course, you should be able to:
1. Identify the advantages of using a Web application framework when designing, developing, and deploying a Web application.
2. Identify common features of Web application frameworks in general.
3. Explain how the specific features of Django correlate with common Web application framework features.
4. Design, code, test, and deploy Web applications that use Django features for:
    a. Models
    b. Views
    c. Templates
    d. URL Mapping
    e. Forms
    f. User authentication/authorization
    g. Session management
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 Web application framework based upon an anticipated set of non-functional requirements (graduate students only).

ALA Competencies
- 4A. Information, communication, assistive, and related technologies as they affect the resources, service delivery, and uses of libraries and other information agencies.
- 4B. The application of information, communication, assistive, and related technology and tools consistent with professional ethics and prevailing service norms and applications.
- 4C. The methods of assessing and evaluating the specifications, efficacy, and cost efficiency of technology-based products and services.
- 4D. The principles and techniques necessary to identify and analyze emerging technologies and innovations in order to recognize and implement relevant technological improvements.

Instructional Methods
- Reading
- Recorded lectures and tutorials
- Text-based online discussion via D2L forums
- Voice-based online discussion via GoToMeeting sessions
- Designing, coding, testing, and deploying a full-featured Django application

Time Expectations
This course requires a weekly time commitment. General university guidelines indicate that a 3-credit course requires a minimum 144-hour time commitment over the semester. This time commitment represents a minimum of 9-10 hours of work per week per course. Three of these hours are lectures. Students are expected to do an additional 6-7 hours per week of study and work on assignments to achieve the learning goals of this course.

Required Text
Pinkham, A. (2016). Django Unleashed. Pearson Education.
Print ISBN: 9780321985071
eBook ISBN: 9780133812398

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 10 or a recent release of MacOS. While the software that we will be using for this course does run on Linux computers, 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 PyCharm integrated development environment (IDE) for Python
- The Anaconda open data science platform for Python
- The Git version control system

You will be deploying several of your coding assignment solutions and your final project to a cloud-based Django hosting facility provided by Python Anywhere. Free accounts are available from Python Anywhere that have all of the capabilities that you will need during the course.

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 10 and recent releases of MacOS. Links to these resources will be provided in our Weekly Schedule.

Course Schedule

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

http://courseinfo.ligent.net/2018sp/_uwm/infost691_212_213/index.html

The course schedule is always subject to reasonable change to account for changes in circumstance and to correct errors. When I make changes to the schedule, I will announce them via D2L Announcements and email.

Course Elements

1. Readings

Required readings will be assigned from the resources listed above and from other resources that will be identified in the schedule. Generally, readings are chosen to accompany any lecture video or tutorial video provided for the week. So, you should expect to complete the readings before playing the videos.

Optional readings will be assigned from time to time. These will typically represent alternate expressions of the same material, or interesting supplementary topics.

2. Videos

I will be providing a series of recorded video lectures and tutorials throughout the course. Generally, you can expect lectures to supplement (rather than repeat) the content of the readings. Tutorials will typically show a skill being practiced using the PyCharm IDE and related software tools. Typically, you will be assigned a parallel coding assignment in the same week that can be accomplished using the approach demonstrated in the video.

3. 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 video.

A D2L dropbox 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 Coding Assignments will be reviewed in the next Online Office Hours / Lab Session. One or two students will be asked to present their work and we will discuss it (constructively and supportively). Then, I will present my version of the assignment solution (never perfect) and we will discuss that as well. The real learning comes from the combination of having tried the skill and the subsequent discussion. Those who have really done the work before the week-ending deadline will get that benefit. Those who wait and do the work later will get a greatly reduced benefit. Having seen our solutions, they will miss out on the benefit of having tackled one of these problems from scratch.

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. After each Coding Assignment, we will review solutions during the Online Office Hours / Lab Session. Further, I will publish a copy of my solution on our D2L site. These published solutions and solution discussions 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 D2L assignment submission activity within 2 weeks.

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 week-ending deadline. This allows you to participate fully in discussions of exercise solutions during Online Office Hours / Lab Sessions.

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

- 10 points will be awarded for submitting a single, properly named and properly formatted file to the proper D2L 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 less in this category.

The implication of this grading scheme is that you can expect a score of 85 or higher on all Coding Assignments that meet both of these criteria.

4. *Participation*
   a. Each student is expected to contribute 3 significant (200 – 300 word) posts to the discussion forums for the class. These should include:
      1) 1 post to the *Introduce Yourself Forum* during Week 1 of the semester.
      2) You are expected to make a total of 2 posts to the *Final Project Forum* during the semester. Each of these posts should discuss potential projects that you are considering for your final project. Your post should address the nature of the application, the availability of users from whom you could elicit requirements, the suitability of the application as a demonstration of Web application framework features, and the likelihood that you might continue developing the application after the semester.
   b. You are also expected to read all of the posts of other students made in all of the discussion forums and respond with short posts when appropriate.

c. Every _____ evening, I will be holding an Office Hours / Lab Session using the GoToMeeting Platform. I encourage you to drop by to review solutions to recent Coding Assignments, to ask a question, to get help with your Coding Assignments, to get help with your Final Project, or just to say hello. Please use a headset when joining this session. Note that individual meetings may be arranged on other days and at other times. To arrange such a meeting, contact me via email.

Please note that participation in Online Office Hours / Lab Sessions is optional. For those who are unable to attend, I will post links to recordings of these sessions to our Weekly Schedule on the following day.

d. Participation grades will be based upon the number of participation points earned during the semester. These are the activities through which you can earn participation points:

Activity	Points Earned
1 long-form primary post to the *Introduce Yourself* or *Final Project* forms.	3
1 post to the *Open Discussion* forum.	1
1 speaking contribution during Online Office Hours / Lab Session	1
1 presentation of your Coding Assignment solution during Online Office Hours / Lab Session	5

5. *Final Project*

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

The Web application should fully demonstrate the Web application framework features covered in the class.

The Web application should include significant add/change/delete 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.

6. *Framework Evaluation Paper (Graduate Students Only)*

   Each graduate student will write a paper presenting the conclusions of their evaluation of the suitability of a candidate Web application framework (other than Django) for an identified set of non-functional requirements. Your paper should be about 1500 to 2000 words. In addition, you need to cite a minimum of 10 sources.

   Detailed instructions and a grading rubric for the Final Project will be published separately.

Grading

Basis for Determining Grade
The grading of each assignment will be the same for both graduate and undergraduate students. Because graduate students are required to complete an additional assignment (Framework Evaluation Paper), the relative weighting of each grading component will be different for graduate students versus undergraduate students. The following weights will be used when computing final number grades:

Grading Component	Undergraduate Students	Graduate Students
Coding Assignments	40%	25%
Framework Evaluation Paper	N/A	33%
Final Project	50%	32%
Participation	10%	10%
Total	100%	100%

Letter grades will be determined as follows:

- A   93 - 100%;
- 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%;

UWM AND SOIS ACADEMIC POLICIES

The following links contain university policies affecting all SOIS students. Many of the links below may be accessed through a PDF-document maintained by the Secretary of the University: http://www.uwm.edu/Dept/SecU/SyllabusLinks.pdf. Undergraduates may also find the *Panther Planner and Undergraduate Student Handbook* useful (http://uwm.edu/studenthandbook/student-handbook/).

Students With Disabilities
If you will need accommodations in order to meet any of the requirements of a course, please contact the instructor as soon as possible. Students with disabilities are responsible to communicate directly with the instructor to ensure special accommodation in a timely manner. There is comprehensive coverage of issues related to disabilities at the Student Accessibility Center (http://www4.uwm.edu/sac/), important components of which are expressed here: http://www.uwm.edu/Dept/DSAD/SAC/SACltr.pdf.

Religious Observances
Students' sincerely held religious beliefs must be reasonably accommodated with respect to all examinations and other academic requirements, according to the following policy: http://www4.uwm.edu/secu/docs/other/S1.5.htm. Please notify your instructor within the first three weeks of the Fall or Spring Term (first week of shorter-term or Summer courses) of any specific days or dates on which you request relief from an examination or academic requirement for religious observances.

Students Called to Active Military Duty
UWM has several policies that accommodate students who must temporarily lay aside their educational pursuits when called to active duty in the military (see http://www4.uwm.edu/academics/military.cfm), including provisions for refunds, readmission, grading, and other situations.

Incompletes
A notation of "incomplete" may be given in lieu of a final grade to a student who has carried a subject successfully until the end of a semester but who, because of illness or other unusual and substantial cause beyond the student's control, has been unable to take or complete the final examination or some limited amount of other term work. An incomplete is not given unless the student proves to the instructor that s/he was prevented from completing course requirements for just cause as indicated above (http://www4.uwm.edu/secu/docs/other/S31.pdf).

Discriminatory Conduct (such as sexual harassment)
UWM and SOIS are committed to building and maintaining a campus environment that recognizes the inherent worth and dignity of every person, fosters tolerance, sensitivity, understanding, and mutual respect, and encourages the members of its community to strive to reach their full potential. The UWM policy statement (http://www4.uwm.edu/secu/docs/other/S47.pdf) summarizes and defines situations that constitute discriminatory conduct. If you have questions, please contact an appropriate SOIS administrator.

Academic Misconduct

Cheating on exams and plagiarism are violations of the academic honor code and carry severe sanctions, ranging from a failing grade for a course or assignment to expulsion from the University. See the following document (http://uwm.edu/academicaffairs/facultystaff/policies/academic-misconduct/) or contact the SOIS Investigating Officer (currently the Associate Dean) for more information.

Complaints

Students may direct complaints to the SOIS Dean or Associate Dean. If the complaint allegedly violates a specific university policy, it may be directed to the appropriate university office responsible for enforcing the policy.

Grade Appeal Procedures

A student may appeal a grade on the grounds that it is based on a capricious or arbitrary decision of the course instructor. Such an appeal shall follow SOIS appeal procedures for undergraduates as seen here: (http://www4.uwm.edu/sois/programs/graduate/mlis/policies/appeals.cfm) In the case of a graduate student, the Graduate School, (http://www4.uwm.edu/sois/programs/undergraduate/ug_appeals.cfm).

Last Revised
2018-01-25