KIEI-925

Winter Quarter 2014
Kellogg School of Management
Northwestern University
(DRAFT)

Course Description

Instructor: Jeffrey Cohen
Lectures: 3 hours, once a week; exact time TBA
Discussion Section: 2 hours, once a week; exact time TBA

This course is geared for entrepreneurs and non-programmers who want to build their own web apps and manage a software product lifecycle. Students will build a project in small steps, learning the fundamentals of computer programming and modern web app development along the way.

Knowing how to hack also means that when you have ideas, you'll be able to implement them.... It's a big advantage, when you're considering an idea like putting a college facebook online, if instead of merely thinking "That's an interesting idea," you can think instead "That's an interesting idea. I'll try building an initial version tonight."
-- Paul Graham

Learning Goals

1. The Ruby on Rails web framework for building web applications
2. Key concepts and patterns in modern web programming regardless of framework
3. Communicate more effectively and credibly with developers on your teams
4. The core principles of agile team management and best practices
5. Prioritize features more efficiently by developing a better feel for their costs

Student Expectations

This is an empowering course, but it is *not* an easy course. Students should expect to spend 8-10 hours a week on project assignments. Students are highly encouraged to work with a partner to help each other on their projects.

Attendance at lectures is paramount. Beginner-level materials for building web apps as scarce. There is no textbook that can act as a substitute. If you miss two lectures you will probably never catch up.

Students are also ***highly encouraged*** to attend a weekly discussion section led by the instructor. The discussion section is not mandatory, but is being offered in recognition of the fact that the material contains demanding technical concepts that are foreign to most Kellogg students. Students who do not attend any of the discussion sections will have a harder time completing their projects.

Homework will be due prior to the discussion section with no exceptions. Homework solutions will be presented during the discussion section.

Computer Requirements

Students must own or have access to a modern computer that is capable of running the Ruby on Rails framework. MacBooks are highly recommended, though most PCs can be made to work as well. Detailed installation instructions for both platforms will be provided at the first class.

Course Organization

Students must build a project of their choosing throughout the quarter as their final project. Weekly homework will require building the next small step of their project (no waiting until week 9 to start on your project :-)

Students are encouraged to work in pairs, but are free to work on a solo project if they prefer. You can also help each other on the homework, but you must type up your final solutions yourself (no copy/pasting).

- **Homework is worth 50% of your grade**, based on 8 weekly assignments worth 6% each. You can miss one assignment for free.
- There will be a **Midterm Exam worth 20% of your grade** around week 6 or 7. It will be a mix of multiple-choice and true/false questions.

- ✦ **The Final Project is worth 30% of your grade** and must be accompanied by a short 3-minute presentation, highlighting the purpose of your app and what you learned in the process of building it.

Note that if you simply turn in all of your homework on time, your final project will be almost complete by week 9.

Suggested Reading

Course notes will be provided, but students may wish to augment their learning with any the following text:

- ❖ *Learn to Program*, 2nd Edition, Chris Pine, Pragmatic Bookshelf 2009.

Calendar

Week	Topics	Assignment
1	Introduction to HTML and CSS. Wireframes. User stories. Prioritization.	*Meet w/instructor to develop customized homework plan*
2	More HTML/CSS. CSS Frameworks. Intro to Ruby.	*Homework #1: HTML mockup, computer setup*
3	Ruby instructions. Lists of things. Defining custom instructions. Using APIs.	*Homework #2: Use Ruby to generate a real-time weather website.*
4	Ruby objects and models. Ruby classes and class inheritance. Intro to Rails. Dynamic web pages.	*Homework #3: Rails app with one model.*
5	Databases. Forms and other user input. Database-backed models. Model lifecycle.	*Homework #4: One database-backed resource.*

Week	Topics	Assignment
6	**MIDTERM.** Models in Rails. Domain modeling. One-to-many relationships.	*Homework #5: Connect two models together.*
7	Many-to-Many modeling. Full domain modeling.	*Homework #6: Connect two models together.*
8	User accounts. User authorization and security. RubyGems.	*Homework #7: User accounts.*
9	Model validations. Goodies like pagination, callbacks, and Ruby Gems.	*Homework #8: Model validations.*
10	Surprises. TBA.	
11	Present and Celebrate!	