1. Executive summary

Wasique Ahmad and **Sajid Shaikh** formed the **Data Miners** team. We chose the project to enhance the **Educational Web** system as we saw a lot of potential in it for UIUC students. We observed the opportunity to make several changes to make the application more useful, along with adding material from more courses thus expanding the scope and reach of the application.

2. Functionality completed

The Data Miners team managed to complete the following enhancements to the Educational Web system. The source code is located at: https://github.com/sajidws/CourseProject (branch: master)

a) Trim the entries in the 'Lectures' dropdown so that the list is more usable and easier to scan quickly
   - Each entry used to start with something like '02 Week 1 02 Week 1 Lesson 01 Lesson 1...'. Removed the redundant parts and made it more intuitive

b) Make the slide material from another course (CS 425: Distributed Systems) available to the users so that users can expand their learning to other courses
   - As part of this story, added 'CS 425' in the 'Courses' dropdown list

c) Improve presentation of information the following screens to make the application more user-friendly:
   - Add week number, lecture number and topic to the current slide so that the user knows which week and lecture covers the current slide.
   - Add week number and lecture number to the list of slides shown in the 'Related slides' section so that the user knows which week and lecture covers each slide listed.
   - Add miscellaneous UX improvements such as capitalize 'CS' (for Computer Science) wherever it appeared.

d) Add a home / landing page to the application that lists the courses available to that users as soon as they 'arrive' in the application.

e) Allow users to go back to the home / landing page from any pages so that users can navigate easily between different parts of the application.

Sajid Shaikh completed the user stories a, b and c.

Wasique Ahmad completed the user stories d and e.

Detailed instructions on how to deploy and run the application have been updated in the project 'README.md', including updates for Windows 10 platform.

3. Breakdown of the tasks

Here's a breakdown of the amount of effort spent by the team in various activities of the project, including comparison with the original estimates.

#	Task	Original estimate (Hrs)	Actual effort (Hrs)	Status
1	Writing proposal, user stories	2	2	Done

2	Set up and build the code	6	20	Done
3	Understanding the existing system and design for the scope of our project	6	14	Done
	Design of enhancements	8	8	Done
	Development, code reviews and unit-testing	20	30	Done
	System / QA verification	6	5	Done
	Documentation	8	10	Done
	Demonstration preparation	4	5	Done
	Communication		5	Done
Total		**60**	**99**	

4. Notes and Experiences

As can be seen from above, we spent far more time than we had estimated. One of the primary reasons for this turned out that the project had previously been tested only on Linux and MacOS, but not on Windows and the Data Miners team was more familiar on Windows platform. We decided to take up the challenge anyway to set up and deliver the project on Windows 10.

Expectedly, we faced many hurdles as we were in uncharted territory, but with help from each other, course staff and a bit of luck, managed to complete the deployment of the project on Windows 10. We have updated the project README with detailed instructions (including various package versions, paths, etc.) so that future students have an easier time.

Even after deploying though, some of the functionality such as Explanations and Search didn't work as expected. We started to fix those, ran into more issues (e.g. after fixing the URL for explanations, the current ranking function hung). Rather than spend more time in what could have been a long and unknown effort, we decided to focus on implementing the functionality we had planned.

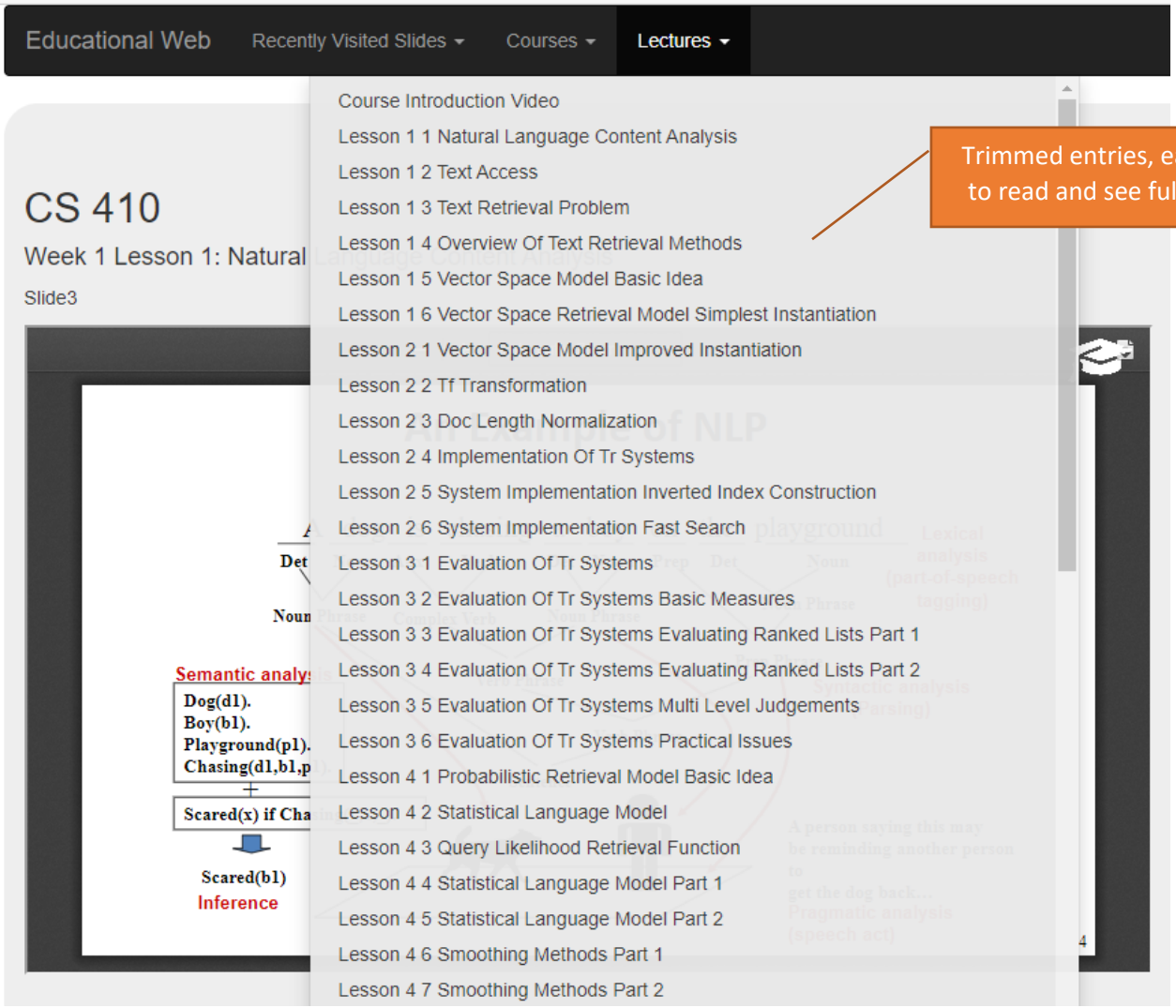
There were some other missteps along the way (e.g. tried to change the directory names to improve the lecture name entries in the dropdown, but that broke other functionality), but each time we learned something new and continued forward.

It's not unusual in any software project to have new discoveries during execution and our experience was no different. The above experience resulted in spending much more time than we had planned but it turned out to be a learning experience and we're glad if our efforts could simplify the life of future students.

5. User story details

In this section, we will dive into the details of each user story implemented, including the technical portions as well as UI screen shots of the application. A majority of the changes made to the code were in the HTML templates (*.html) and Python files, especially model.py and app.py.

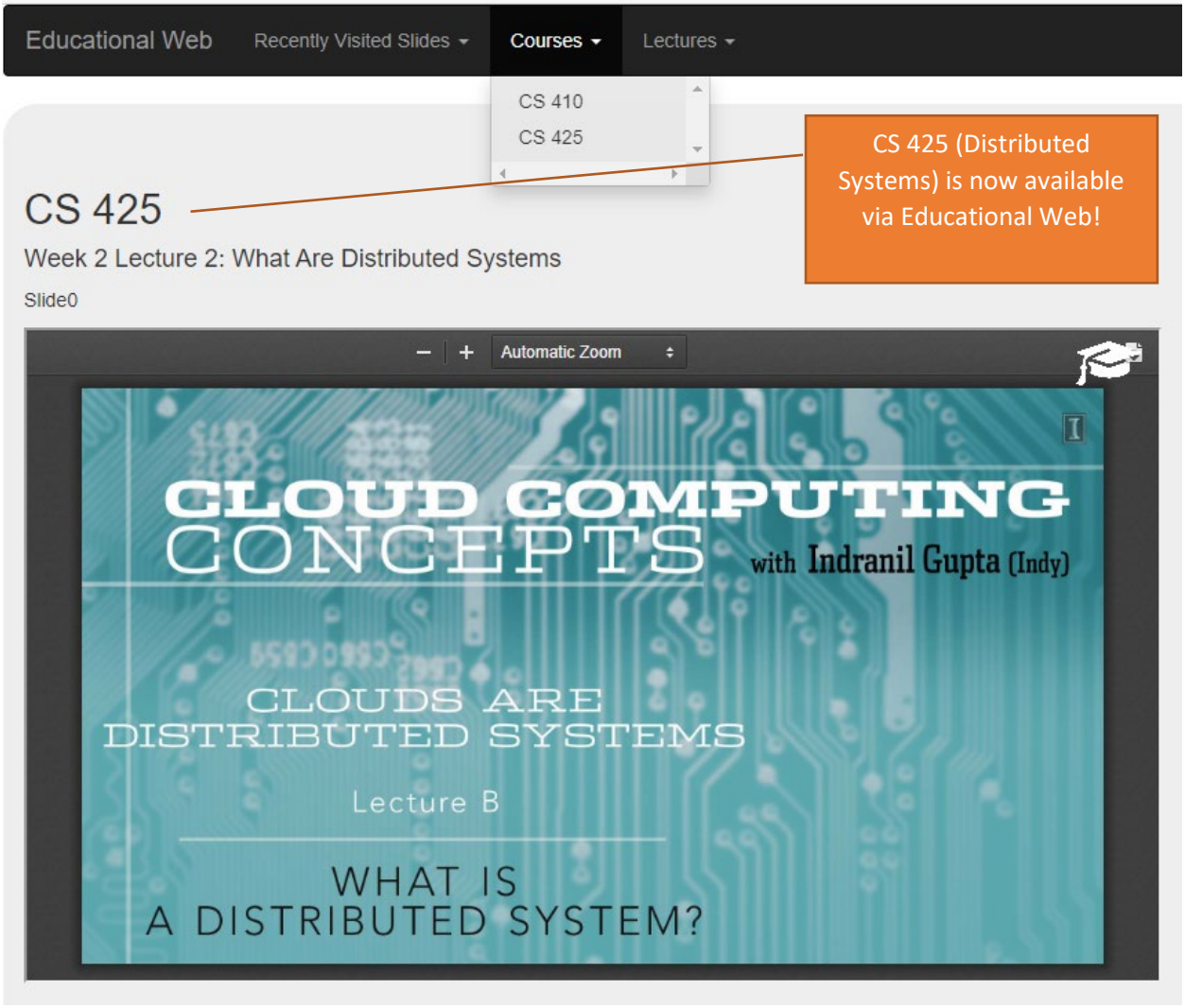
Trim the entries in the ‘Lectures’ dropdown so that the list is more usable and easier to scan quickly



Initially, we tried to simplify the directory names as that allowed us more control over the names that could be presented to the user. We then realized there are many other dependencies on the names of the directories (e.g. Related Slides weren’t appearing). Ultimately, we changed the manner in which ‘lec_names’ are displayed in the UI by changing ‘slide.html’. The highlight of the change was:

Line 7: {{' '.join(lec_names[i].split('_')[3].split('-')).title()}}

Make the slide material from another course (CS 425: Distributed Systems) available to the users



We initially thought just adding a folder such as 'cs-425' in 'static/slides' directory will do the job. However, that didn't work and we had to make changes in 'model.py' to remove hard-coded checks of 'cs-410'. Ultimately, this turned out to be a time-consuming activity. The PDF files from the CS 425 Distributed Systems lectures had to be split into individual pages, each named in a certain format. We have packaged a new archive 'cs-425.zip' and included its location and instructions to deploy in the README file.

cs425.zip is available here: https://drive.google.com/file/d/1IWxuYF1fGHlU1VZn5xfXfyCJxUV-sIRI/view?usp=sharing

Educational Web | Recently Visited Slides | Courses | Lectures

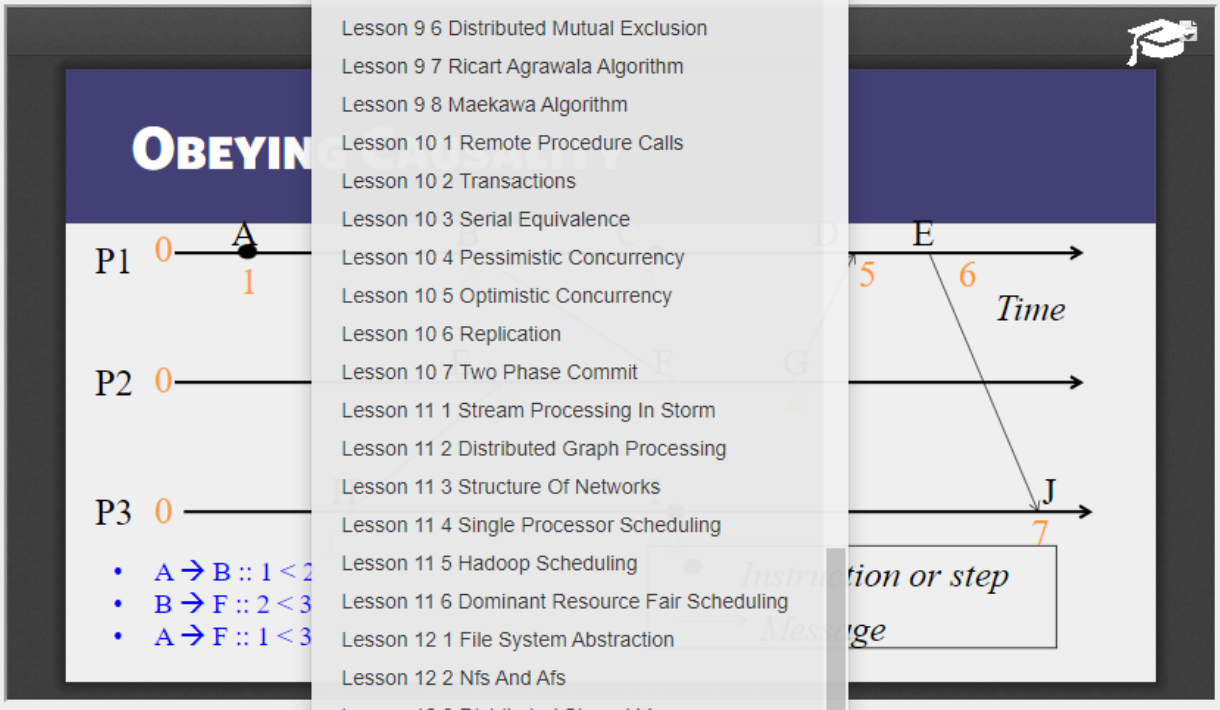
- Lesson 6 12 Paxos
- Lesson 6 13 Flp Proof
- Lesson 9 1 The Election Problem
- Lesson 9 2 Ring Leader Election
- Lesson 9 3 Chubby And Zookeeper
- Lesson 9 4 Bully Algorithm
- Lesson 9 5 Mutual Exclusion
- Lesson 9 6 Distributed Mutual Exclusion
- Lesson 9 7 Ricart Agrawala Algorithm
- Lesson 9 8 Maekawa Algorithm
- Lesson 10 1 Remote Procedure Calls
- Lesson 10 2 Transactions
- Lesson 10 3 Serial Equivalence
- Lesson 10 4 Pessimistic Concurrency
- Lesson 10 5 Optimistic Concurrency
- Lesson 10 6 Replication
- Lesson 10 7 Two Phase Commit
- Lesson 11 1 Stream Processing In Storm
- Lesson 11 2 Distributed Graph Processing
- Lesson 11 3 Structure Of Networks
- Lesson 11 4 Single Processor Scheduling
- Lesson 11 5 Hadoop Scheduling
- Lesson 11 6 Dominant Resource Fair Scheduling
- Lesson 12 1 File System Abstraction
- Lesson 12 2 Nfs And Afs
- Lesson 12 3 Distributed Shared Memory

The entire CS 425 course material!

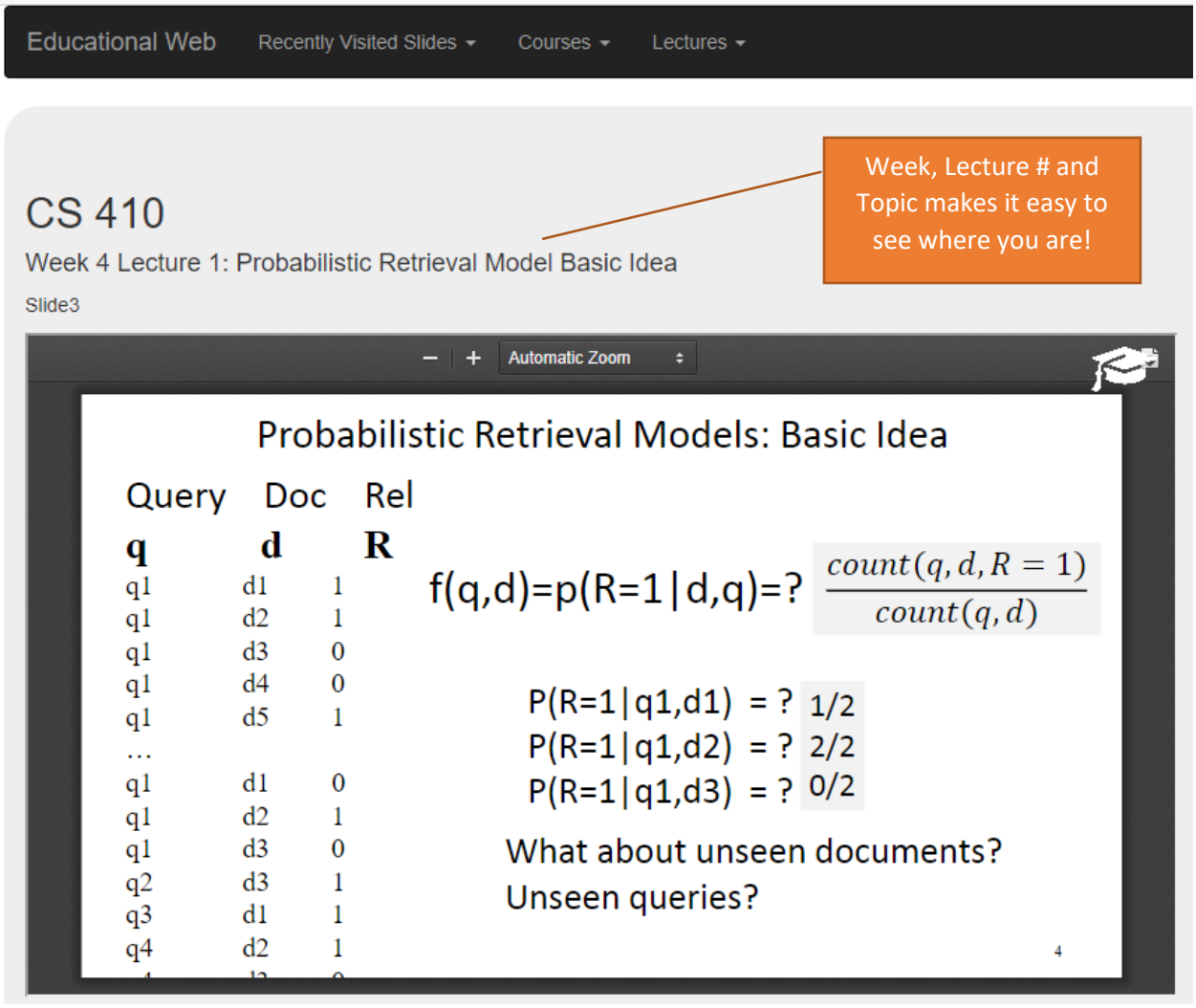
CS 425

Week 5 Lecture 9: Lampor

Slide16

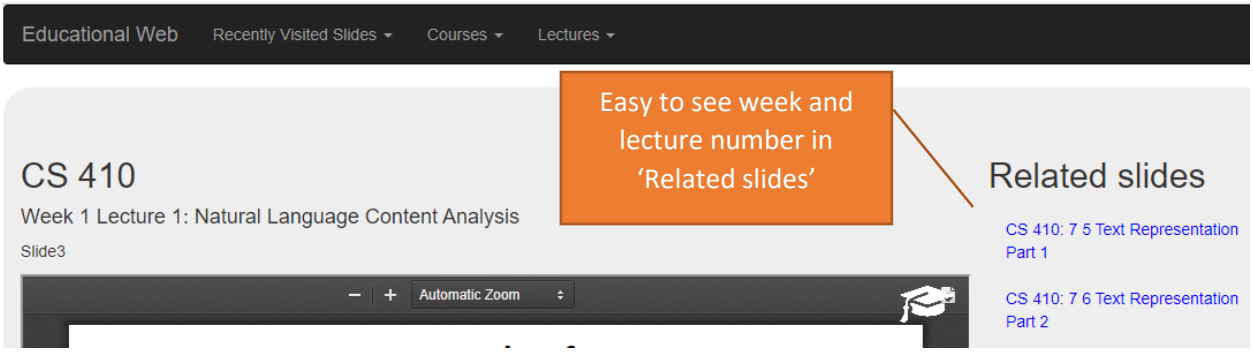


Add week number, lecture number and title to the current slide so that the user knows which week, lecture and topic covers the current slide



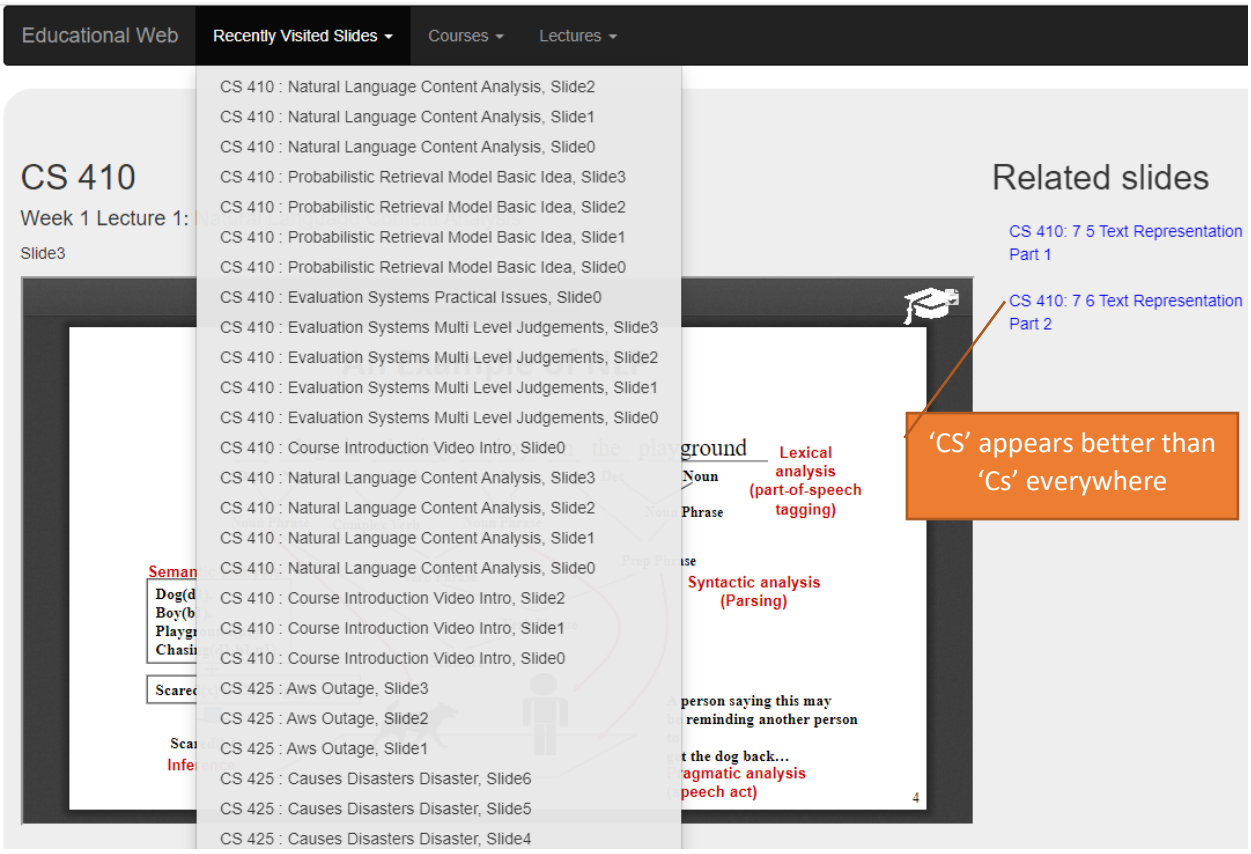
While this change was relatively straightforward as we had to extract the week and lesson number by making changes in 'slide.html', it adds quite some value to the user to see which topic the current slide is covered under and to quickly determine the week and lecture number.

Add week number and lecture number to the list of slides shown in the ‘Related slides’ section so that the user knows which week and lecture covers each slide listed



The changes made here were similar by extracting the week and lecture number in ‘slide.html’

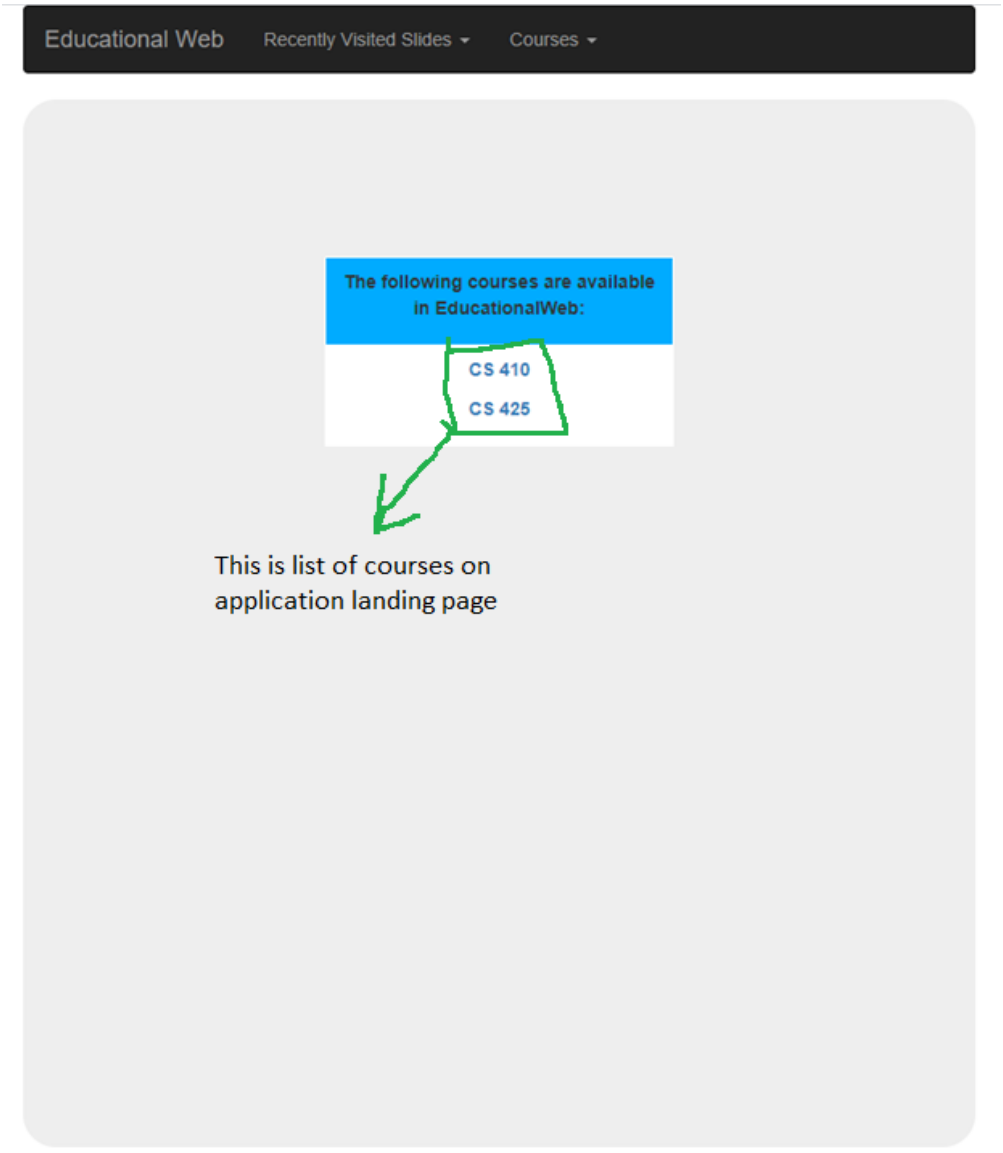
Add miscellaneous UX improvements such as capitalized ‘CS’ (Computer Science)



A relatively easy change by changing ‘title()’ to ‘upper()’ in multiple places where ‘CS’ is going to be displayed.

Adding landing page

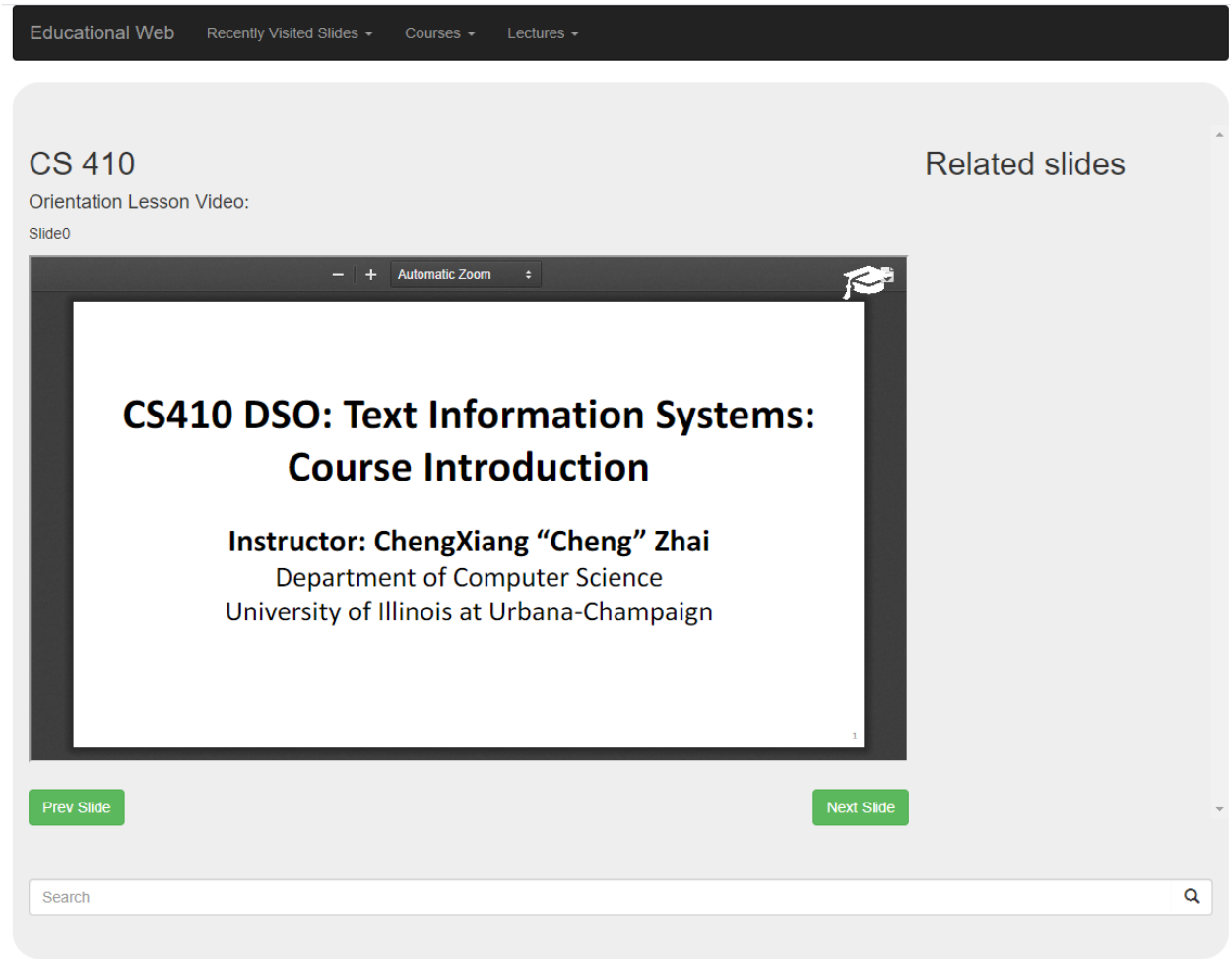
This is will be the landing page. List of the courses offered by EducationalWeb application will be shown here.



The file ‘LandingPage.html’ was added in the ‘templates’ folder to implement the above functionality. This page extends ‘base.html’.

In order to load this page when the user visits the base URL (‘/’), ‘app.py’ was modified to render ‘LandingPage.html’ instead of ‘base.html’.

This will be shown as a 1st slide after selecting course ‘CS 410’ from the above page. Clicking on ‘Educational Web’ in the top bar from any page will bring users back to the landing page.



In order to implement the above change, ‘base.html’ was modified to change the link for ‘Educational Web’ to ‘/’ instead of ‘#’.