

COMPSCI 2021: Web App Development 2- Team Project

LectureFinder

By Group 7A



LectureFinder

▶ https://github.com/Ishitanarsiker/WAD2_GroupProject

▶ Participants

1. Ishita Narsiker 2579990N
2. Amy Eden 2514468E
3. Luke Mullen 2542408M
4. Petros Kitazos 2526547K
5. SoonKwang Hwang 2572157H



LectureFinder: Overview

- ▶ LectureFinder is a platform for professors to upload study materials in the form of recorded lectures, reading materials, slides and transcripts to help students find exact instances of specific concepts being referred among one or more modules of study
- ▶ Students can perform keyword searches and get filtered results based on Professor Name / Module Name
- ▶ Student also have the flexibility to create an account where they can save searched content to their profile for future reference
- ▶ Professors will be able to upload study materials for the students to save/highlight



Target Audience

- ▶ Students wanting to revise specific concepts from previous semester/year
- ▶ Mature students with limited time to go over each and every single concept- helps them cater their study by concept
- ▶ Teaching Staff who'd like to provide ease of access to study materials in one place with student flexibility to access it



LectureFinder: Technologies Used

- ▶ Cascading Style Sheets & HTML5
 - ▶ To format the web pages and make it look polished and refined
- ▶ JavaScript
 - ▶ To keep certain functions working (e.g. animations, popups, etc.)
- ▶ Python
 - ▶ To make and store our database
- ▶ PythonAnywhere
 - ▶ To host our web application
- ▶ AJAX and jQuery
 - ▶ To make the like button functional
- ▶ JUnit tests
 - ▶ To make sure our web application functions as expected
- ▶ JSON
 - ▶ saved the secret_key which is originally in the settings.py and gmail password for lectureFinder91



Features of the Web Application

- ▶ Signup - Choose the roll between Professor/Lecturer or Student
- ▶ Login, Logout, Reset Password
- ▶ Most viewed lectures and recently uploaded lectures
- ▶ Search Lecture documents and videos with transcripts for key words
- ▶ For Students - Saved lectures
- ▶ For Professors - Uploaded lectures and Saved Lectures
- ▶ Lecture documents links (PDF or PPTX)
- ▶ Lecture video links with transcripts



Approach: Front End

▶ Styling

- ▶ Goal was to make the application look polished and refined
- ▶ So we made a separate css file to implement different styles and formatting into our application

▶ Templates

- ▶ Used the inheritance nature of templates so reduce duplicity
- ▶ Each web page inherits the base.html as to keep formatting consistant

▶ User authentication

- ▶ We used JS to help us make our sign in and log in forms functional and forms.py to actually store the user information
- ▶ We created custom css error messages that clearly state what went wrong to the user, whilst looking polished



Approach: Back End

- ▶ Models
 - ▶ Based on ER diagram we created in the design spec (without the moodle API)
- ▶ Views
 - ▶ We have separate views for each web page, pop up and lecture
- ▶ Population script
 - ▶ We used this to store user information
 - ▶ We used this to store the lecture content, saved lectures and courses



Summary of Team's Contribution

- ▶ Ishita Narsiker : Backend related with starting lecture search function
- ▶ Amy Eden : Frontend related with formatting the web pages with css and js, formatting darkmode, signup and login forms. Backend related with mapping the URLs and starting the models/forms
- ▶ Luke Mullen : Backend related with population script (to hold lecture content, user information), views, forms, unit tests, a functional search function and cleaning up code
- ▶ Petros Kitazos : Frontend related with formatting the web pages with css and js, added animations and images to clean up visually
- ▶ SoonKwang Hwang : Backend related with account reset, darkmode



Live Demonstration

