

COMP-8347-91 I/S 2020

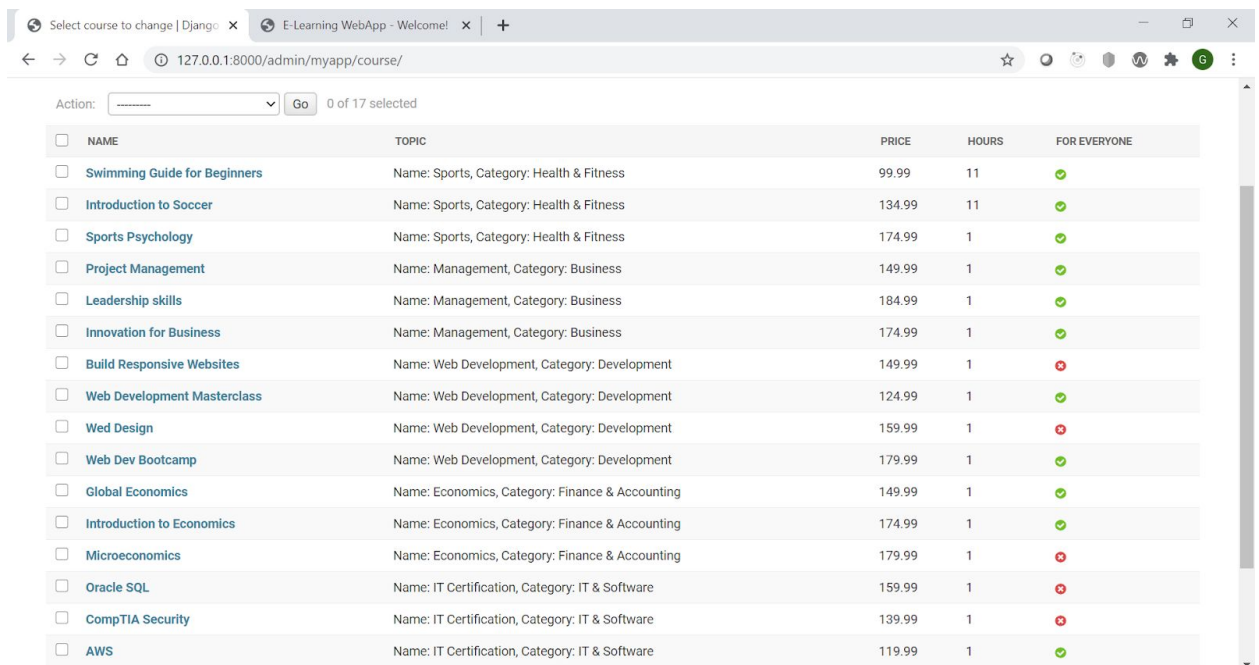
Final Project Tasks

Team Members:

Arvind Narayan Srinivasan (105187866)
Gundeep Singh (110008447)
Venkata Deepak Nannapaneni (105188586)
Vigneshwaran Balasubramani (105221382)

Required Features:

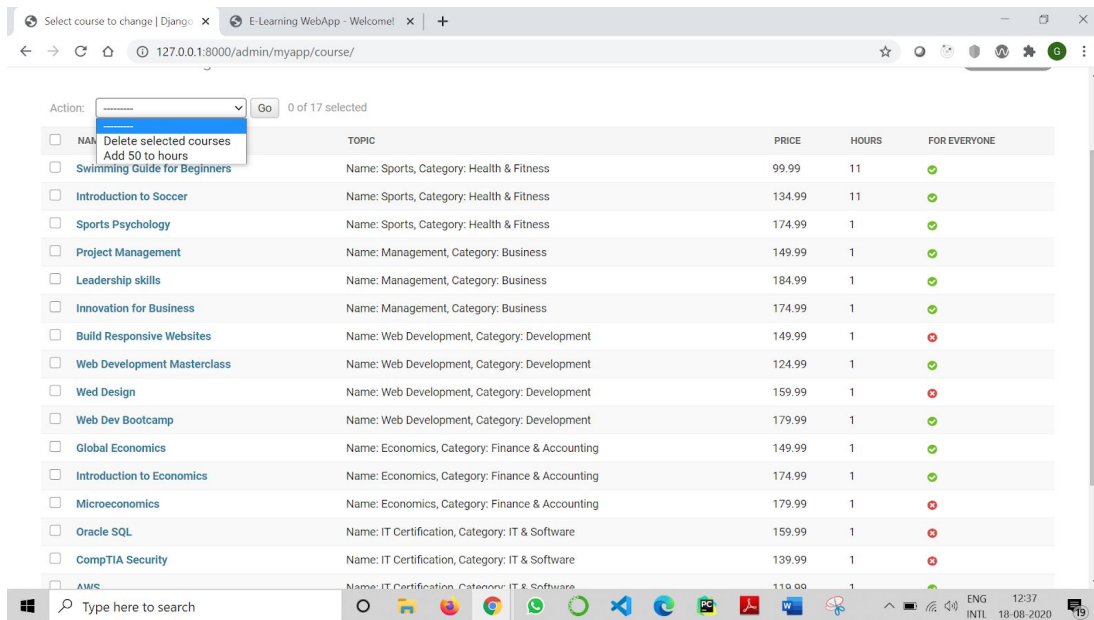
- 1) In admin.py create a class CourseAdmin(admin.ModelAdmin), register this with the admin site and show the name, topic, price, hours, and for_everyone fields, for each Course, in the admin interface page that lists all Courses. (1)



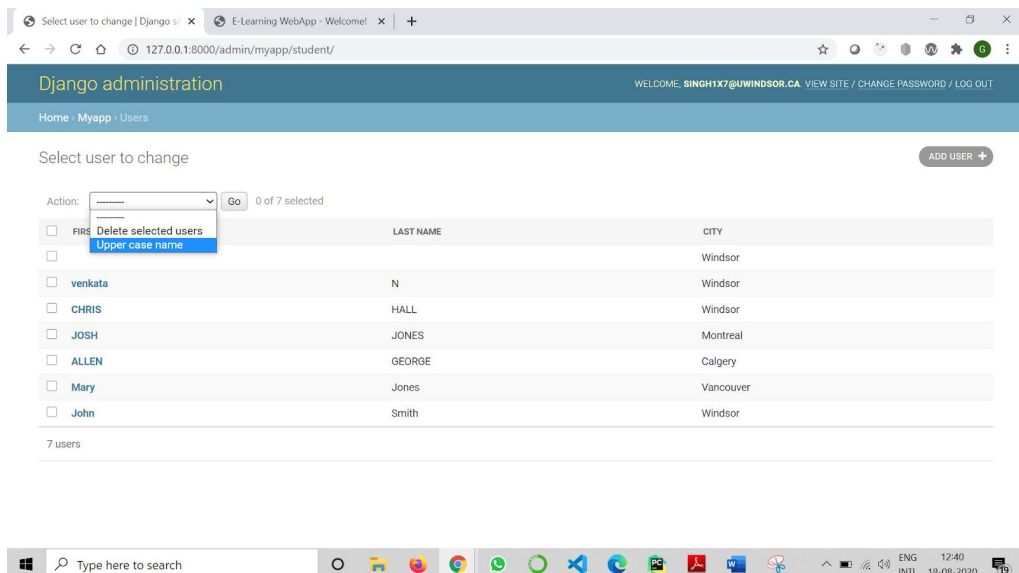
The screenshot shows the Django admin interface for a web application. The browser address bar indicates the URL is 127.0.0.1:8000/admin/myapp/course/. The interface displays a table of courses with columns for NAME, TOPIC, PRICE, HOURS, and FOR EVERYONE. There are 17 courses listed, each with a checkbox for selection. The 'FOR EVERYONE' column contains green checkmarks or red X marks.

NAME	TOPIC	PRICE	HOURS	FOR EVERYONE
<input type="checkbox"/> Swimming Guide for Beginners	Name: Sports, Category: Health & Fitness	99.99	11	✓
<input type="checkbox"/> Introduction to Soccer	Name: Sports, Category: Health & Fitness	134.99	11	✓
<input type="checkbox"/> Sports Psychology	Name: Sports, Category: Health & Fitness	174.99	1	✓
<input type="checkbox"/> Project Management	Name: Management, Category: Business	149.99	1	✓
<input type="checkbox"/> Leadership skills	Name: Management, Category: Business	184.99	1	✓
<input type="checkbox"/> Innovation for Business	Name: Management, Category: Business	174.99	1	✓
<input type="checkbox"/> Build Responsive Websites	Name: Web Development, Category: Development	149.99	1	✗
<input type="checkbox"/> Web Development Masterclass	Name: Web Development, Category: Development	124.99	1	✓
<input type="checkbox"/> Web Design	Name: Web Development, Category: Development	159.99	1	✗
<input type="checkbox"/> Web Dev Bootcamp	Name: Web Development, Category: Development	179.99	1	✓
<input type="checkbox"/> Global Economics	Name: Economics, Category: Finance & Accounting	149.99	1	✓
<input type="checkbox"/> Introduction to Economics	Name: Economics, Category: Finance & Accounting	174.99	1	✓
<input type="checkbox"/> Microeconomics	Name: Economics, Category: Finance & Accounting	179.99	1	✗
<input type="checkbox"/> Oracle SQL	Name: IT Certification, Category: IT & Software	159.99	1	✗
<input type="checkbox"/> CompTIA Security	Name: IT Certification, Category: IT & Software	139.99	1	✗
<input type="checkbox"/> AWS	Name: IT Certification, Category: IT & Software	119.99	1	✓

- 2) In admin.py write an action add_50_to_hours(...)for CourseAdmin class that will add 10 to the current value of the hours field for the selected courses and save the updated value in the database. (2)



- 3) In admin.py create a class StudentAdmin(admin.ModelAdmin), register this with the admin site and show the first_name, last_name, city fields. Define upper_case_name(...) function to display the student's first and last name in uppercase. Make the short description of this function Student Full Name. (2)



- 4) Add 'register' view that allows a user to register as a Student. Update myapp/urls.py and create register.html with suitable content. This feature involves creating a ModelForm named

RegisterForm that is based on Student model. The form should include the fields: username, password, first name, last name, city, interested_in. Design the form with labels and widgets as appropriate. Instantiate this form in register() view and pass it to register.html for display. (6)

E-learning Home Register Login

Enter username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Enter password*

First name

Last name

Enter your city*

Interested in*

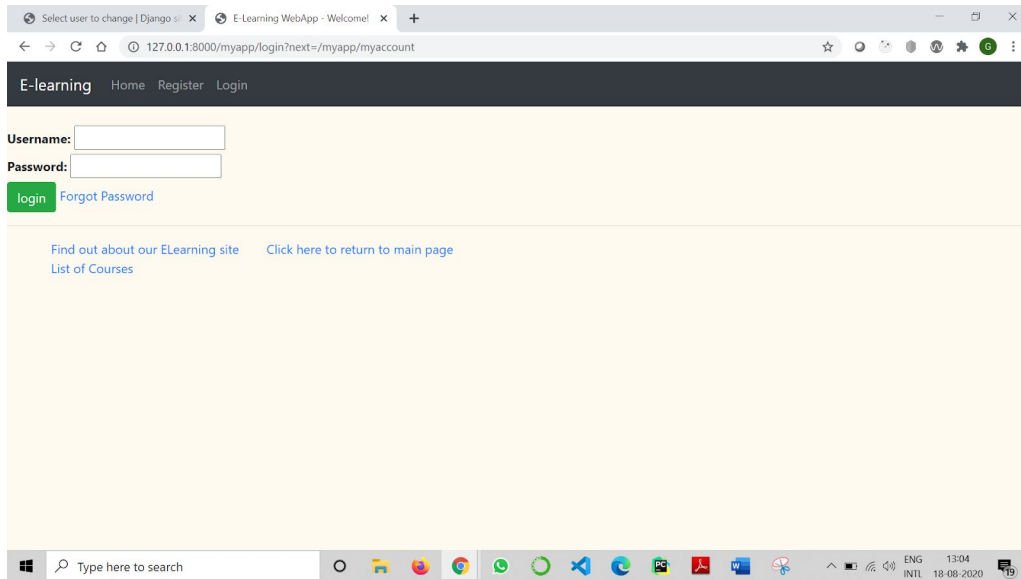
- Name: Web Development, Category: Development
- Name: Management, Category: Business
- Name: IT Certification, Category: IT & Software
- Name: Sports, Category: Health & Fitness

Select your profile picture*

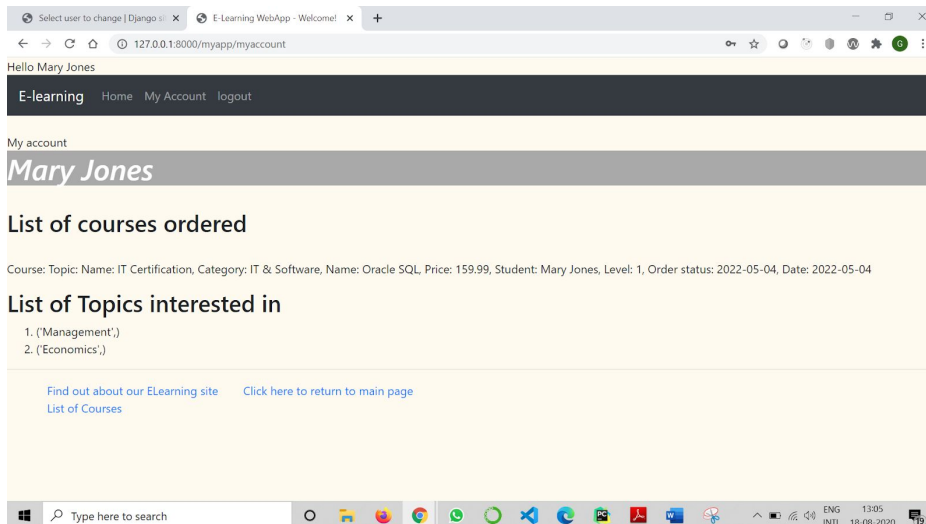
Choose File No file chosen

- 5) Update the user_login view created in Lab 8 so that if an user who is not logged in goes to url '/myapp/myaccount/' they will be directed to the login page and after successful login they will go directly to the '/myapp/myaccount/' page (instead of the main index page). (2)

On clicking the myaccount for a user who is not logged in:

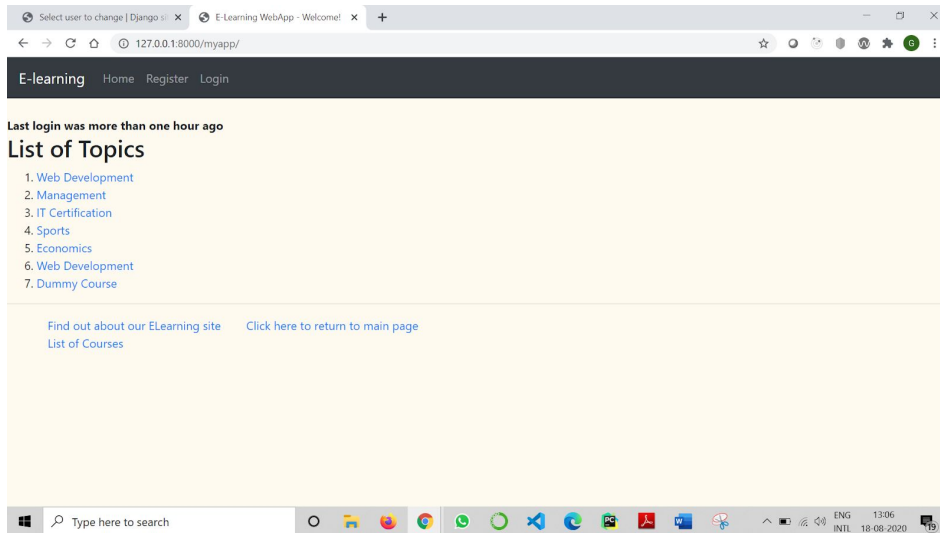


On logging in , the user lands on the myaccount page:

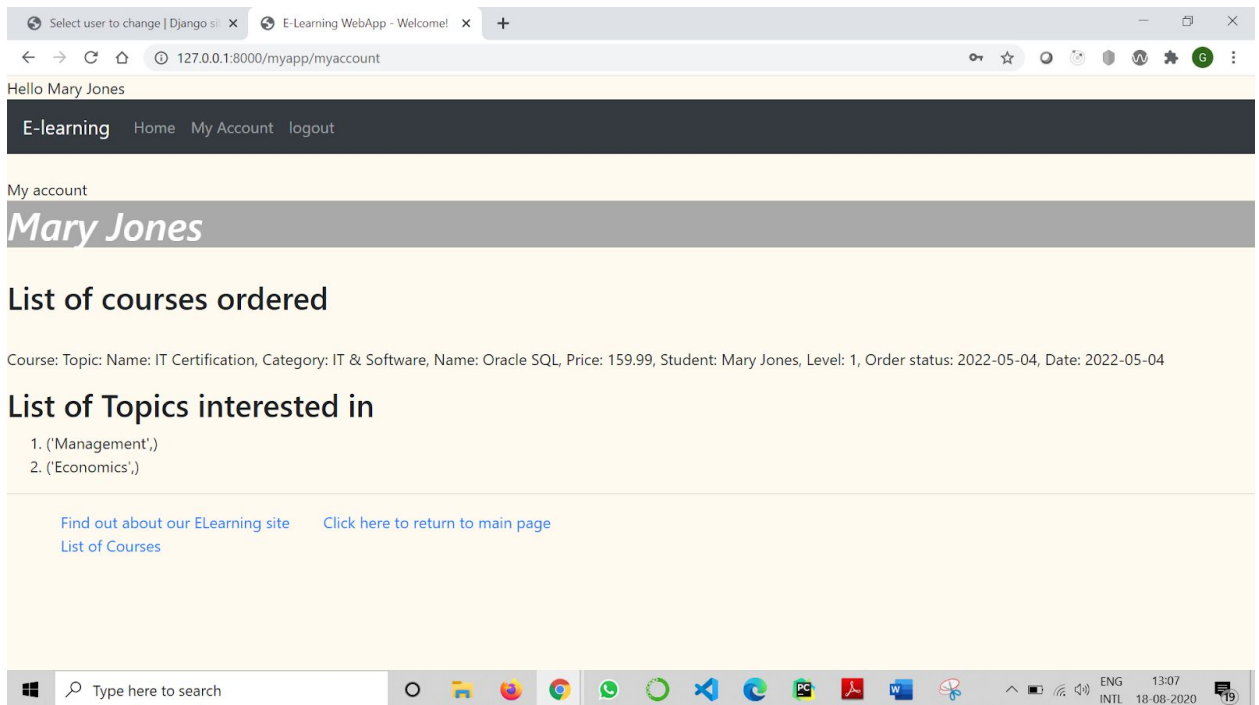


- 6) Update base.html so that if a user is logged in, it will display Logout (myapp/logout) and My Account (myapp/myaccount) links. Otherwise it will display Register Here (myapp/register) and Login (myapp/login) links. Each link should go to the corresponding view function defined earlier (in Lab 8 or in step 4 or 5 above). (3)

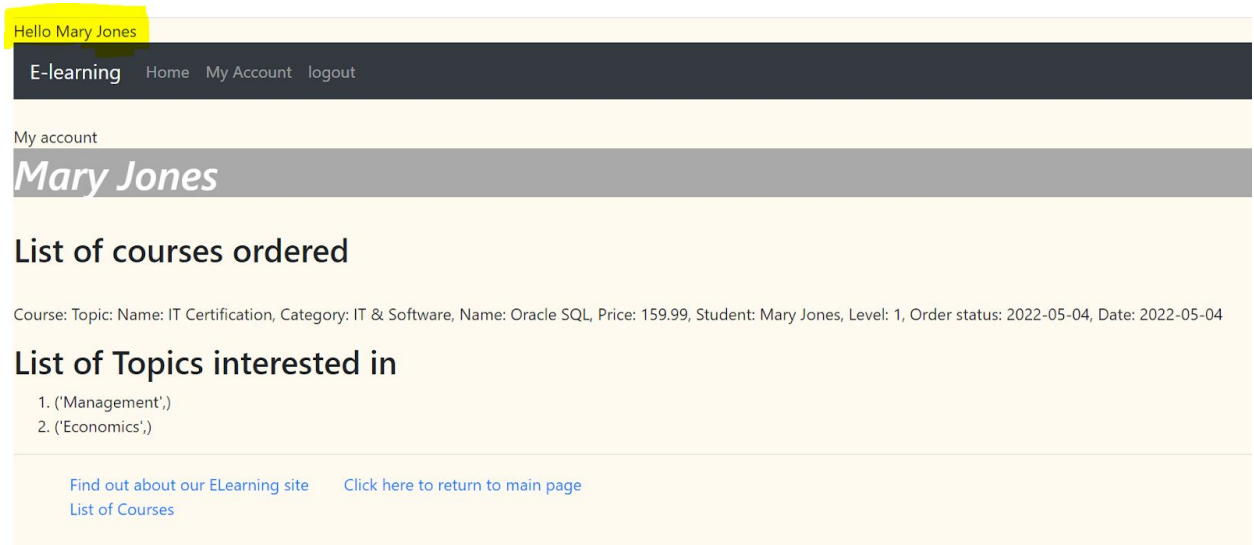
While the user is logged out:



While the user is logged in:

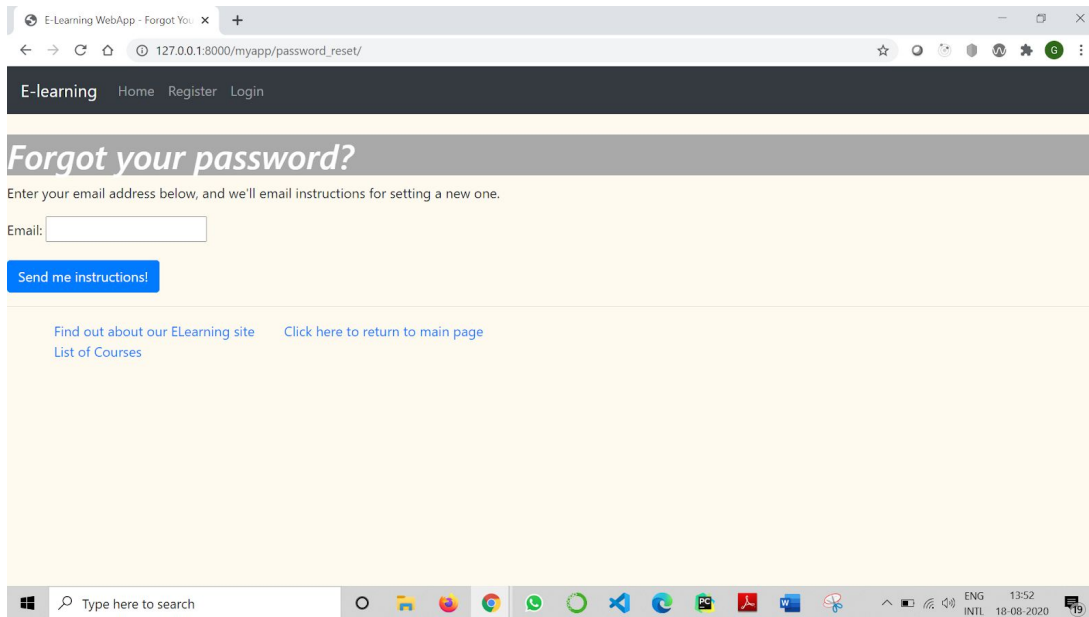


- 7) Update base.html so that if a user is logged in, it will display “Hello ” instead of “Hello User”. Here and are the first name and last name of the user that is currently logged in, respectively.
(1)



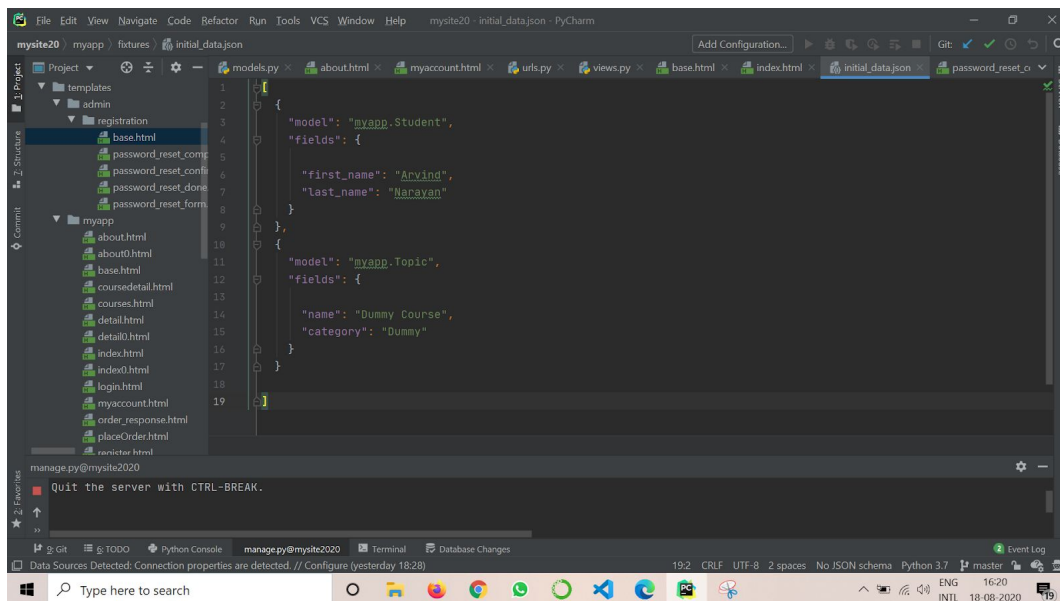
- 8) Add a ‘Forgot password’ link on login page. It should email a new password to the user. (3)

On clicking the forget password link:



Optional Features:

- 1) Save db in JSON format. Load initial data using fixtures



Above shown is the initial data provided to the fields pertaining to the model and the loading of the initial data is done using the following command:

“ `manage.py loaddata <fixturename>` ”

- 2) Add validators for price field in Course model so that it is between 100 and 200. If you're asked to add validators to one more field, which field you'll choose.

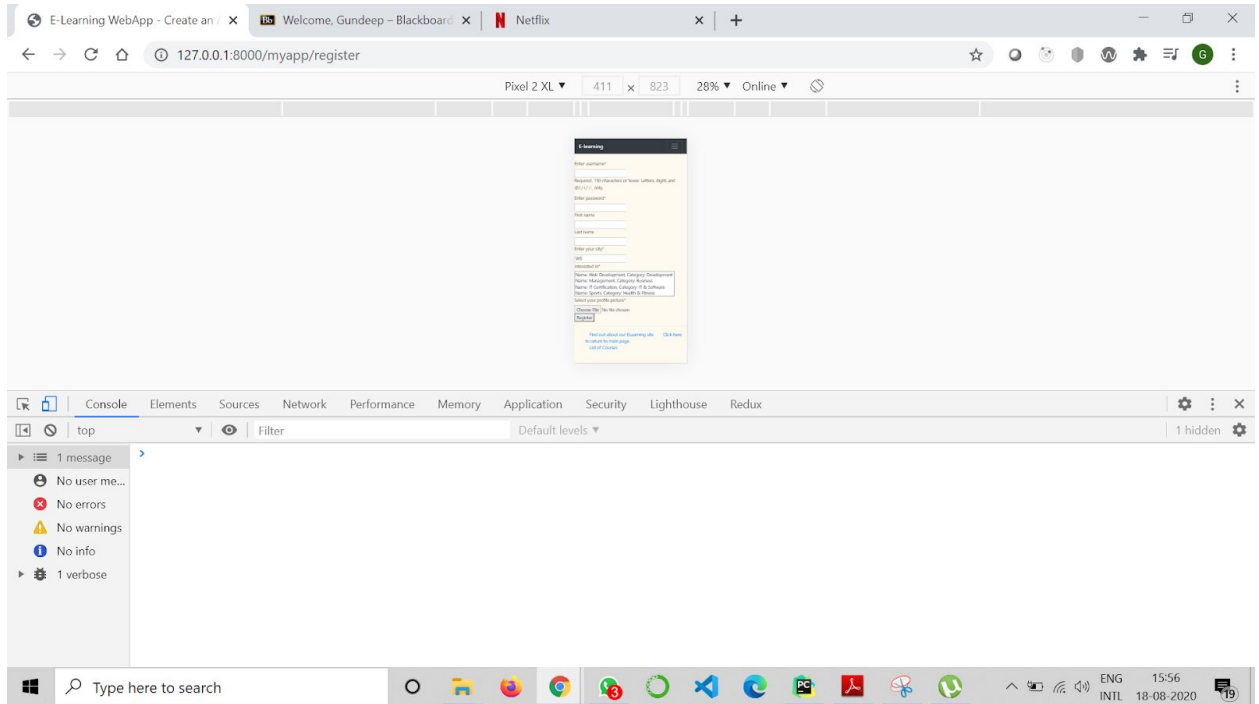
The screenshot shows the Django administration interface for adding a new course. The browser address bar indicates the URL is 127.0.0.1:8000/admin/myapp/course/add/. The page title is 'Django administration' and the user is logged in as SINGH1X7@UWINDSOR.CA. The breadcrumb trail is 'Home > Myapp > Courses > Add course'. The form is titled 'Add course' and contains the following fields: 'Topic' (a dropdown menu with 'Name: Web Development, Category: Development'), 'Name' (a text input with 'sample for price above 200'), 'Price' (a text input with '300'), 'For everyone' (a checked checkbox), 'Hours' (a text input with '1'), and 'Description' (a text area with 'sample description'). A red error message 'Please correct the error below.' is displayed above the 'Price' field, and a specific error message 'Ensure this value is less than or equal to 200.' is shown below the 'Price' input field. The Windows taskbar at the bottom shows the system time as 16:28 on 18-08-2020.

- 3) Upload image file. Add an optional field image field for a Student to upload his/her photo.

The screenshot shows a registration form with a yellow background. At the top, there is a dropdown menu for selecting a course, with two visible options: 'Name: IT Certification, Category: IT & Software' and 'Name: Sports, Category: Health & Fitness'. Below the dropdown, the text 'Select your profile picture*' is displayed. Underneath this text is a file upload field with a 'Choose File' button and the text 'No file chosen'. At the bottom of the form, there is a 'Register' button.

- 4) Use Bootstrap to style your pages.

As evident from the above screenshots as well, we have used the bootstrap 4. For a better example, below is a sample of responsive screen size achieved using bootstrap.



5) Any additional feature(s) you want to include.

We have used django crispy forms to render forms automatically. A filter used with forms variable named “|crispy” to handle div elements. “{% load crispy_forms_tags %}” is used to load the crispy tags in the current template.