

GROUP 12

Member Name

XINXIN ZHAO, 105215004

ANSHUL SHARMA, 110016388

CHINTAN PATEL, 110022265

Github Project: <https://github.com/MaxiAgrippa/IADjangoLab>

REQUIRED FEATURE COMPLETED

1. In admin.py create a class TopicAdmin(admin.ModelAdmin), register this with the admin site and show the name and length fields, for each Topic, in the admin interface page that lists all Topics. Create an TabularInline class called CourseInline based on Course model and include this under the TopicAdmin, so Courses for a particular Topic may be edited on the Topic page itself.
2. In admin.py write an action for CourseAdmin class that will reduce the price of the course by 10% for the selected courses and save the updated price in the database.
3. Create 'register' view that allows a user to register as a Student. Update myapp/urls.py and create register.html
4. Create myorders view. Define a view function myorders(request) in your views.py file. The user must be logged in to access this function. For a logged in user: if the user is a Student, return all the orders that have been placed by the user; otherwise, display a message: 'You are not a registered student!'. Update myapp/urls.py appropriately and create myorders.html to display the orders that have been placed by the student or a suitable message (if the student has not placed any orders).
5. Update the user_login view created in Lab 10 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). Note: The myaccount view function was created in lab 10.
6. Update base.html so that if a user is logged in, it will display Logout(myapp/logout) link. 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 10 or in step 3 above).
7. Update base.html so that if a user is logged in, it will display "Hello <first_name>" instead of "Hello User". Here <first_name> is the first name of the user that is currently logged in

OPTIONAL FEATURES COMPLETED

1. Save db in JSON format. Load initial data using fixtures
2. Add validators for price field in Course model so that it is between \$50 and \$500.
3. Upload image file. Add an optional image field for a Student to upload his/her photo.

4. Update index and detail view functions in views.py to use class-based views (<https://docs.djangoproject.com/en/3.1/topics/class-based-views/>).
5. Use Bootstrap to style your pages
6. In admin.py create a class StudentAdmin(admin.ModelAdmin), register this with the admin site and show the first_name, last_name, level fields and list of registered_courses for each member in the admin interface page that lists all students.
7. Add a 'Forgot password' link on login page. It should email a new password to the user.
8. Update the views.review function (created in lab 9) so that only users who are logged in and are either a 'Undergraduate' student or 'Postgraduate' student can submit a review.