

Internet Applications and Distributed Systems Project e-Book WebApp

Group Members: Deepraj Darbar, Frank Macwan, Priyanka Mamtani

Summary

In the final part of the project we have added several features and functions, which include all the required, optional features as required by the professor but also, have tried to create some interesting features ourselves.

Required features implemented:

- ✓ In admin.py create a class PublisherAdmin(admin.ModelAdmin), register this with the admin site and show the name, website, and city fields, for each Publisher, in the admin interface page that lists all Publishers.
- ✓ In admin.py write an action for BookAdmin class that will add \$10 to the current price of the book for the selected books and save the updated price in the database.
- ✓ Create 'register' view that allows a user to register as a Member. Update myapp/urls.py and create register.html
- ✓ Update the user_login view created in Lab 10 so that if a user who is not logged in goes to url '/myapp/chk_reviews/' they will be directed to the login page and after successful login they will go directly to the '/myapp/chk_reviews/' page (instead of the main index page).
- ✓ 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).
- ✓ Update base.html so that if a user is logged in, it will display "Hello " instead of "Hello User". Here is the first name of the user that is currently logged in.

Optional Features implemented:

- ✓ Save db in JSON format. Load initial data using fixtures
- ✓ Add validators for price field in Book model so that it is between \$0 and \$1000.
- ✓ Upload image file. Add an optional image field for a Member to upload his/her photo.
- ✓ Update index and detail view functions in views.py to use class based views (CBV).
- ✓ Use Bootstrap to style your pages
- ✓ In admin.py create a class MemberAdmin(admin.ModelAdmin), register this with the admin site and show the first_name, last_name, status fields and list of borrowed_books for each member, in the admin interface page that lists all members.
- ✓ Add a 'Forgot password' link on login page. It should email a new password to the user.
- ✓ Update the views.review so that only users who are logged in and are either a Regular member or a Premium member can submit a review.
- ✓ 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 Member, return all the orders that have been placed by the user; otherwise, display a message: 'You are not a registered member!'. Update myapp/urls.py appropriately and create myorders.html to display the orders that have been placed by the client or a suitable message (if the client has not placed any orders).

Any additional feature(s) you want to include:

- ✓ UserRatings page which lists the top books according to the average ratings they have received from the users
- ✓ Profile picture of the User/Member along with the greeting 'Hello User' (accessed from db, where user uploads the image)
- ✓ Book details with Images and publisher website
- ✓ Checkreviews page with list of books which redirect to average reviews of that book (easier method for user)
- ✓ Website redirected to main page when we open localhost