Exploration of all new upgrades for new iteration

**PSEDO/STEPS USER AUTHETICATION**

**Sources:** [**Source 1 ,**](https://d.docs.live.net/53a17bb9a99d641a/Documents/Source%201)[**Source 2**](Source%202)**,** [**Source 3**](Source%203)**,** [**Source 4**](Source%204)

**Models.py**

* Remove the email field from the user model.
* Using Django pre built this is already included so no need for it

**forms.py**

* Create UserRegisterForm extending UserCreationForm.
* Create UserLoginForm extending AuthenticationForm.

**views.py**

* Create the registration view that saves both
* Create a login view using Django's prebuilt system.
* Make a logout view that logs out the user. ( should see register/ login in when log out)
* Update views handling Notebook to ensure users can only update their own. ( Iteration 1 covers this but we need to be able to add Media videos or images)
* Display all notebook and be able to view the score/rounds just not able to add/delete/edit unless it’s there note book assigned to them (shouldn’t change or break have it working)

**urls.py**

* Add URL patterns for **register**, **login**, **logout**, and **notebook** views.

**templates**

* Create **register.html** links to registration.
* Create **login.html** links to user login.
* Update **notebook.html** to check for the user's notebook. ( should be good to go may have to update to allow media and pics)

**settings.py**

* Configure **AUTH\_USER\_MODEL** (should be go using pre built from Django)
* Set LOGIN\_REDIRECT\_URL and LOGOUT\_REDIRECT\_URL. ( where it sends you after you sign up and log out)

**migrations**

* Run make migrations for any changes to your models.
* Run migrationsto update them.

**PSEDO/STEPS FOR UPLOAD/DELETE MEDIA**

**Sources:** [**Source 1**](Source%201) **,** [**Source 2**](Source%202) **,** [**Source 3**](Source%203) **,** [**Source 4**](Source%204) **,** [**Source 5**](Source%205)

**Models**:

* Update golfer to store the images and videos.
* Create method and converting file formats.
* Create method to delete associated images and videos

**Forms**:

* Create from for user file uploads through the web

**Views**:

* To handle the form submissions and associate the uploaded files with the specific user.
* Implement control ensuring only authorized users can upload files to their notebook.
* handle deleting requests for files and remove the associated file from the database and file system.

**Templates**:

* The forms and to show the uploaded files (using boot strap, maybe assign the media to the right of the page and keep the scores on the left)
* Add a delete button for each file in the user's media list

**URLs**:

* Routing for the forms and files (setting urls not in app urls)
* Add a delete button for each file in the user's media list.

**UNIT TESTING PESDO/STEPS 3 unit tests**

**Source:** [**Source 1**](Source%201)**.** ,[**Source 2**](Source%202) **,** [**Source 3**](Source%203)[**Source 4**](Source%204)

**Home page status and Notebook list view**

* Confirms page status sends 200 indicating page loads correctly.
* Verifies that homepage is accessible
* Verifies that notebook\_list can render and has notebooks are included.

**Check detail view notebook**

* Sees if the notebook detail view is correctly displayed for the correct user.
* Verifies the notebook\_detail template is render and contains the correct notebook that is clicked

**SELENIUM TEST PESDO/STEPS 2 tests**

**Sources:** [**Source 1**](Source%201)[**Source 2**](Source%202)[**Source 3**](Source%203)[**Source 4**](Source%204)[**Source 5**](Source%205)

**DELETE THE ROUND**

* Logs in using a password and username already in the database
* Should see log out in nav bar if log in is successful
* Clicks the notebook list in nav bar
* Once page is loaded click on user name notebook
* once loaded delete a round using the delete button

**UPLOAD IMAGE**

* Logs in using a password and username already in the database
* Should see log out in nav bar if log in is successful
* Clicks the notebook list in nav bar
* Once page is loaded click on user name notebook
* once loaded click upload file
* navigate to file file given (something on local machine)
* submit file with the submit button
* waits to see success then close out