**User Registration part 1**

Till this point we were able to fetch the data from database and we are able to display those destinations on our index.html page. We are missing many things with this site, however the important thing is registration and login of user accounts. Discuss the registration process of any social media account(ex: facebook, twitter etc..). In this case this already have admin(superuser)account created and we are not focusing on the normal users. You can manually create x number of users, otherwise you can give a registration form to them so that they can fill the form and register themselves. Once they are registered successfully they can login(via login form) with their credentials.

In this part we are going to create 1) registration form and 2)login form

Let’s see how we can create a registration form, current layout of our home page

Graphical user interface, application

Description automatically generated

Whenever we want to work with something different, example we had created calc app, travello app, now if we want to have the user registration we want to create separate module for it. We can continue with the same app however we are going to create a new module so that we can reuse the module/app for another apps as well. So we are going to create a separate module for accounts(user accounts) .

Run the below command to create a new module accounts.

Graphical user interface, text, application

Description automatically generated

We can see that it had created a separate folder in our IDE, new app with new files, we don’t have **urls.py** for accessing this, so we will create **urls.py** as we are going to work atleast with 3 links, login, registration and logout.

Graphical user interface, text, application

Description automatically generated

Code reuse for other **urls.py** and make the changes to our **urls.py** as shown below.

* url mapping for Register, views.register and name is register

Graphical user interface, text, application, email

Description automatically generated

Post updating the urls.py for the accounts app, we need to map or link it with our main application travello urls.py as well. So, let’s update the travello’s urls.py as shown below. When someone calls the accounts page/path we will call the accounts.urls as mentioned, we have done the same for travello if we can recollect.

Graphical user interface, text, application, email

Description automatically generated

we will create a function in views.py for register and we are going to pass it as we are not giving any information at the moment for testing purpose.

Graphical user interface, text, application, chat or text message

Description automatically generated

If we want to make it work we need to call a url and that will be coming from index.html. This is where we want that button of registration, so basically we need three links login, logout and registration. We will work on registration first as it is first step. Let’s navigate to index.html and remove the about, services and news highlighted below.

Text

Description automatically generated

We will update the page to have the link for registration, register belongs to accounts app. Hence register/accounts and update the name to register as highlighted below.

Text

Description automatically generated

If you refresh the home page we can see that register link appears.

Graphical user interface, application

Description automatically generated

When you click on the register, it will call the function of register which is doing nothing rendering an error.

Graphical user interface, text, application

Description automatically generated

Discuss the error, we can’t simply say pass as it won’t work.

Graphical user interface, text, application

Description automatically generated

Let’s make some changes to our register function by calling the registration page. Return render the page by passing the request object and the page is register.html(which we need to create in templates).

Graphical user interface, text, application

Description automatically generated

Under **templates** we will have a simple **register.html** page or form. We will create a basic html sample try **ctrl+space**. Update the **title to registration**. In the body we will create the **form fields**. Form action(what it is going to call) register and since we are submitting the data we use **method “post”** and we need to find some fields.

Text

Description automatically generated

This registration will go to the server where we have the database. We don’t have the table in the database. So we need to create a table for users in database so that the data from the registration form will store in the database table. We know that when we ran migration it has created many tables in the database and one of the tables is for users. If we view all data, we can see that there are admin accounts we had created. Let’s use the columns available in the table. We will make use of the below highlighted 5 fields.

Graphical user interface, application

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Below is how our form should be like.

Text

Description automatically generated with low confidence

If you refresh our register page, we can see the below for in accounts/register page.

Graphical user interface, text, application

Description automatically generated

Let’s try to access this from the home page.

A screenshot of a computer

Description automatically generated with medium confidence

When you click on register we should see the form. Now if you submit the form it will not save the details in the database as we need to some changes for updating the table.

Graphical user interface, text, application

Description automatically generated

Recap: we had created an app(consists for registration, login, logout). We had created the page for registration so that the user can register themselves.