**IFT458/598 – Project 1**

Yuan Li & Edward Halper

**Introduction**

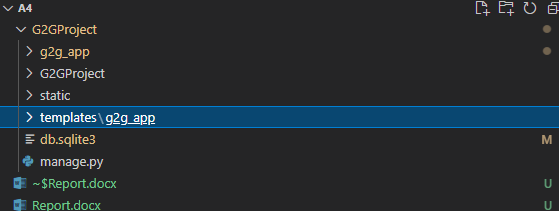
???

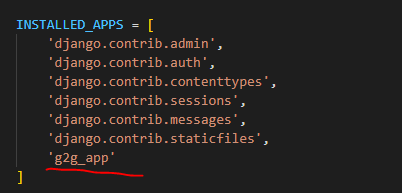
**Description of your work**

|  |  |  |
| --- | --- | --- |
| Name | Item | description |
| Yuan | Home page | This is the base page, it has all the navigation menus, header and footer. It has block tag<% block content%>, so every other page can extend from it |
| Yuan | Login page | It extends the home page. It is connected with Django default databse sqlite3. The authentication API is working. |
| Yuan | Sign Up page | It extends the home page. It is connected with Django default database sqlite3. The authentication API is working, you can create a new user and from this page, then login. You can see from one of the below screenshots, I created 3 demo users in the database. |
| Yuan | Officers page | It displays all the officers’ entities from the backend data source. Right now, the API returns the fake data. Inside the html, my code loops through all the items that returned from the query set. |
| Yuan | Event page | It displays all the event entities from the backend data source. Right now, the API returns the fake data. Inside the html, my code loops through all the items that returned from the query set. |
| Yuan | Contact Us page | It displays a static video, I had to configure static files in settings.py for the video to display. Then I had to use <% load staticfiles%> in my html page to load static file. |
| Yuan | View.py | It has all the view method which is responding to each request, connecting template and fill out the template and return |
|  |  |  |
|  |  |  |

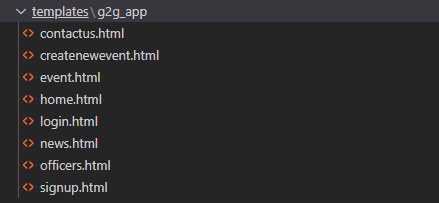
**Screenshots of code narrative from above**

**1: Created a project:G2GProject, application:g2g\_app and register app to the project**

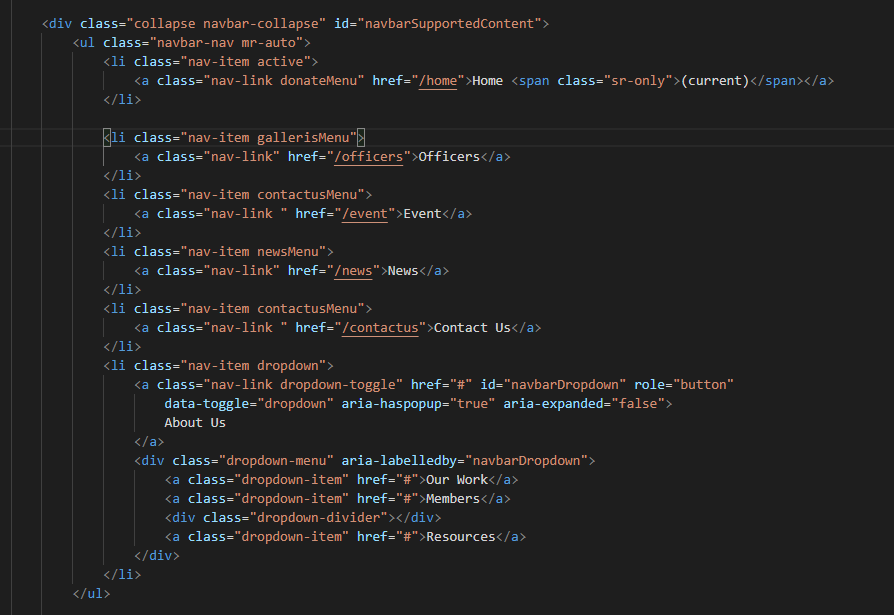




**2: Created templates, including all the html pages.**

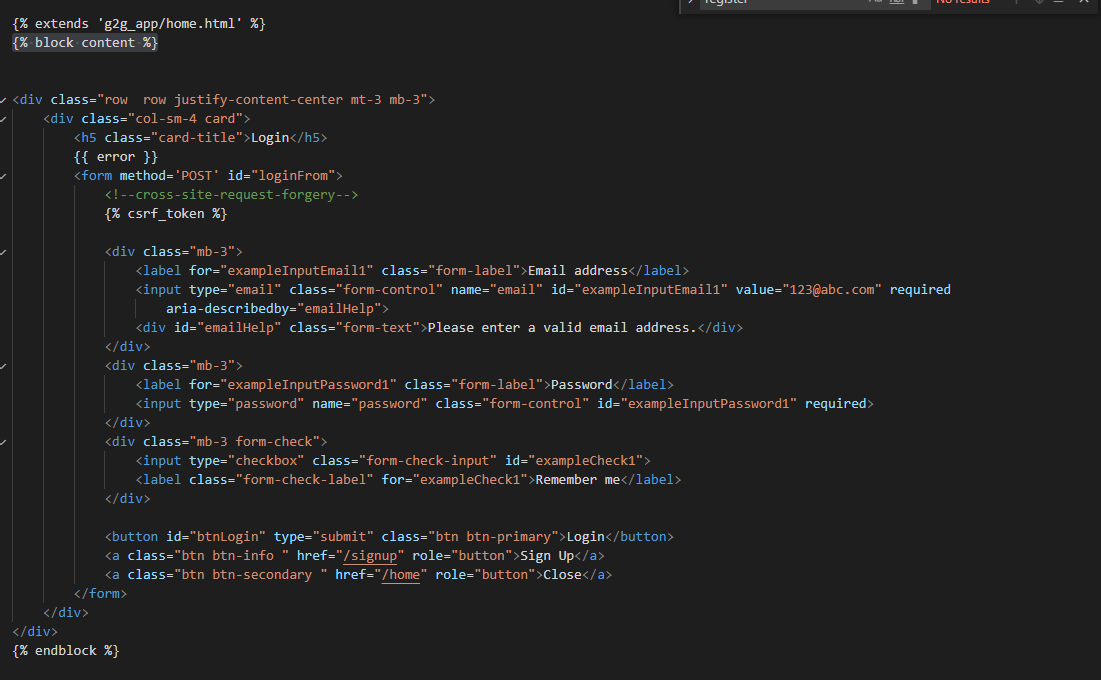


**3: Created the home.html, including nav item and block tag {% block content %}**

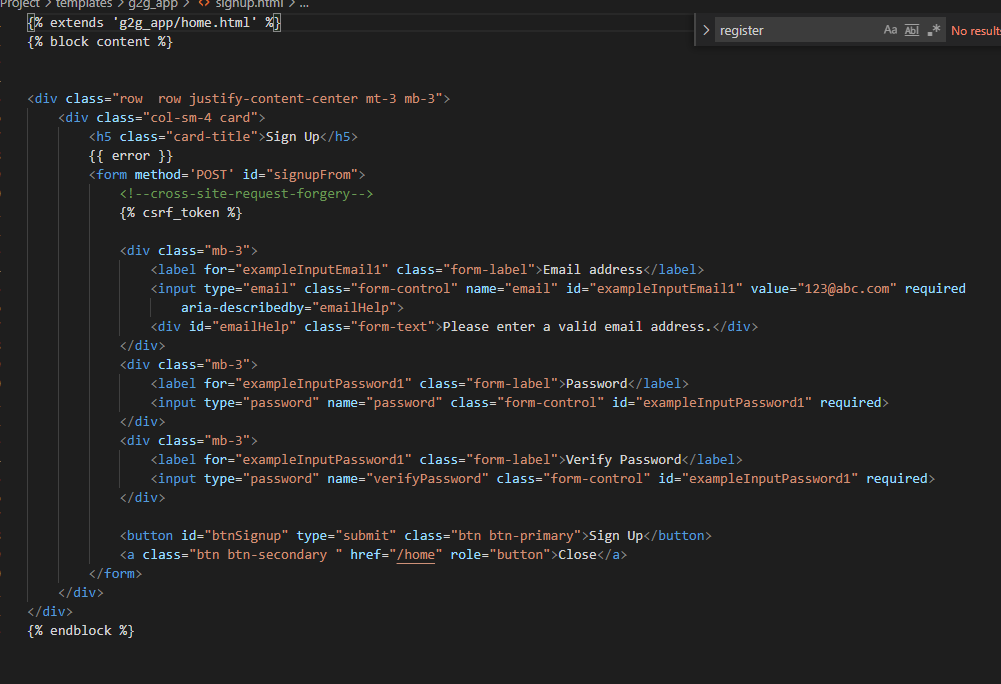




**4: Created login.html**



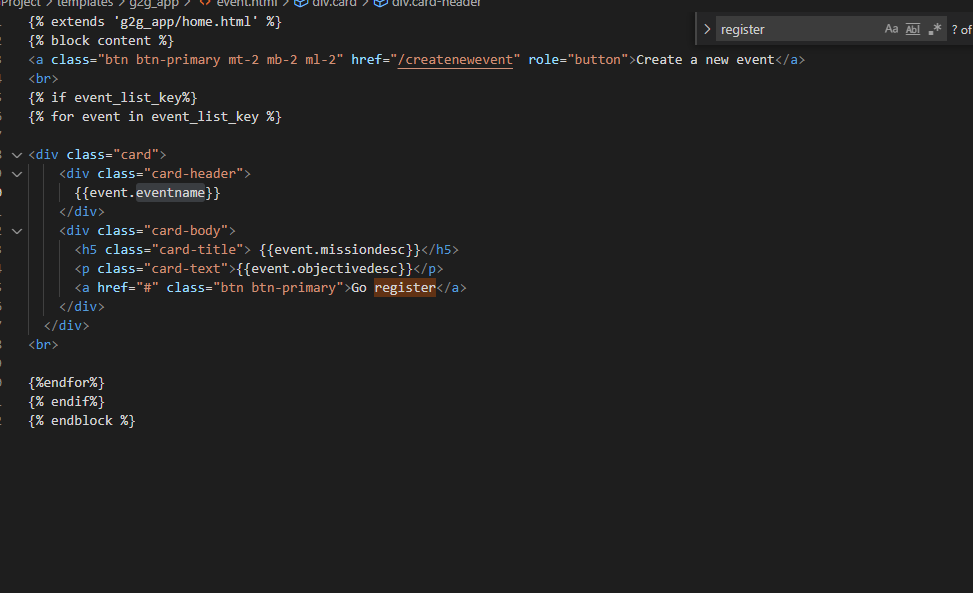
**6. Created signup.html**



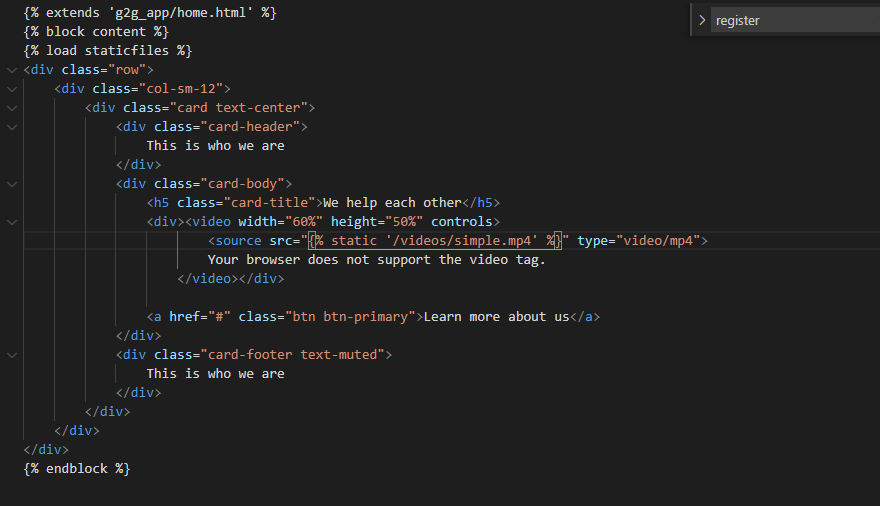
**7. Created officer.html**



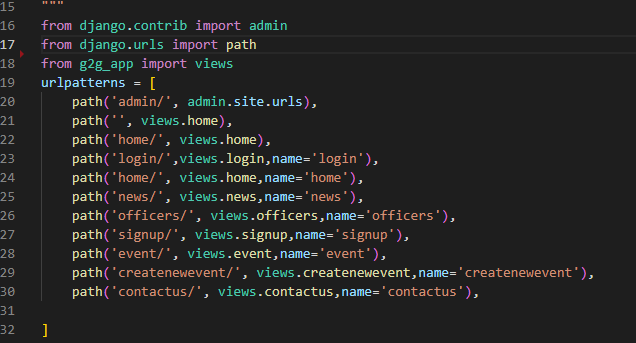
**8. Created event.html**



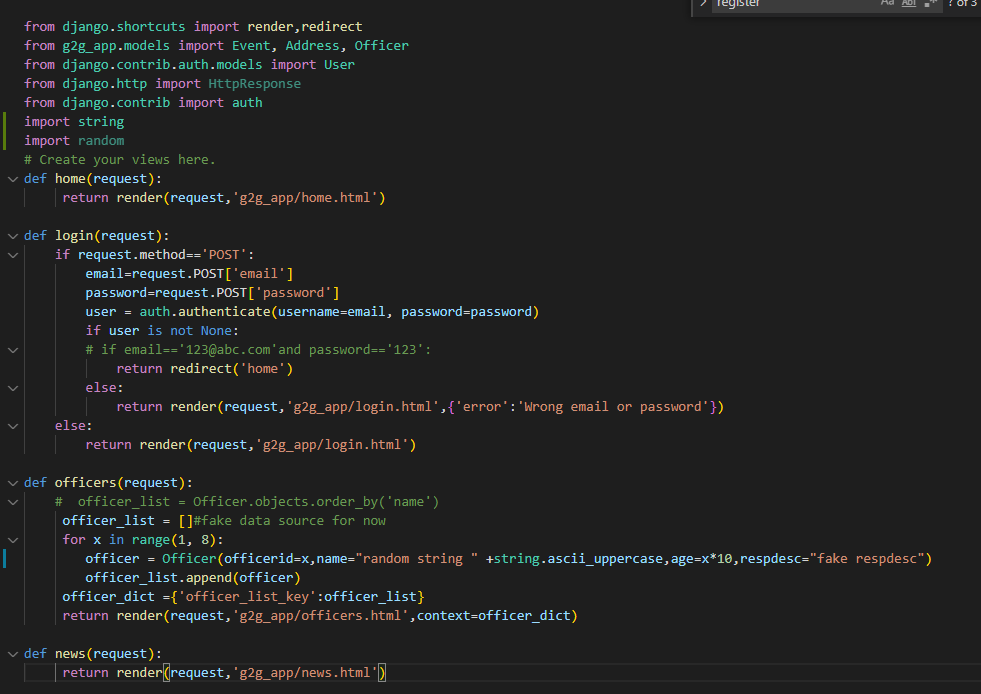
**9. Created contactus.html**

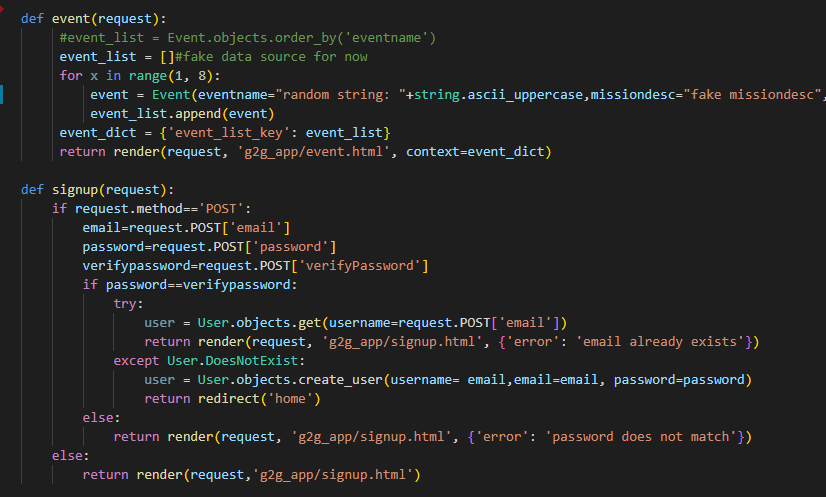


**10. config project urls.py**



**11. Created views to connect template and fill out the template.**

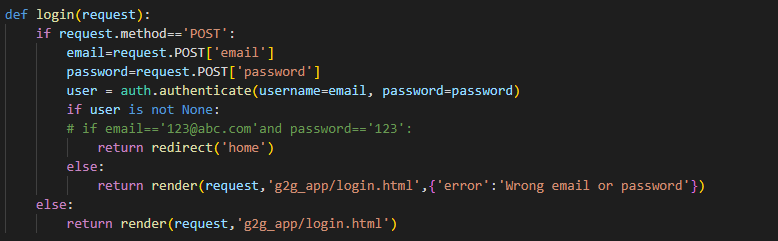






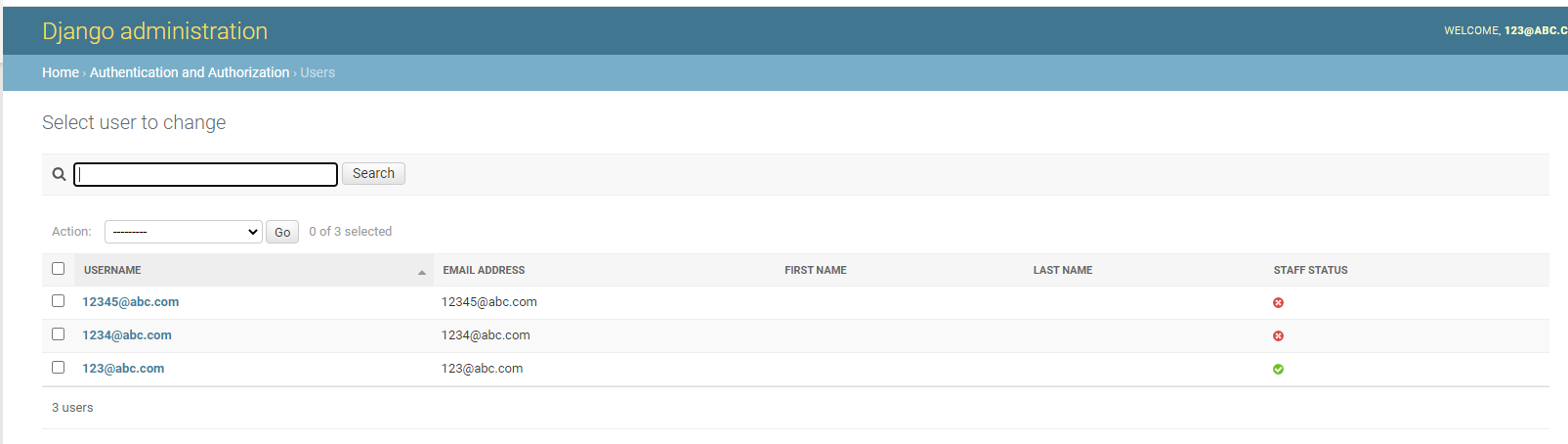
**12. Migrated default sqllist3 database, so the authentication will work**

**13. Created the authentication logic for login and signup**



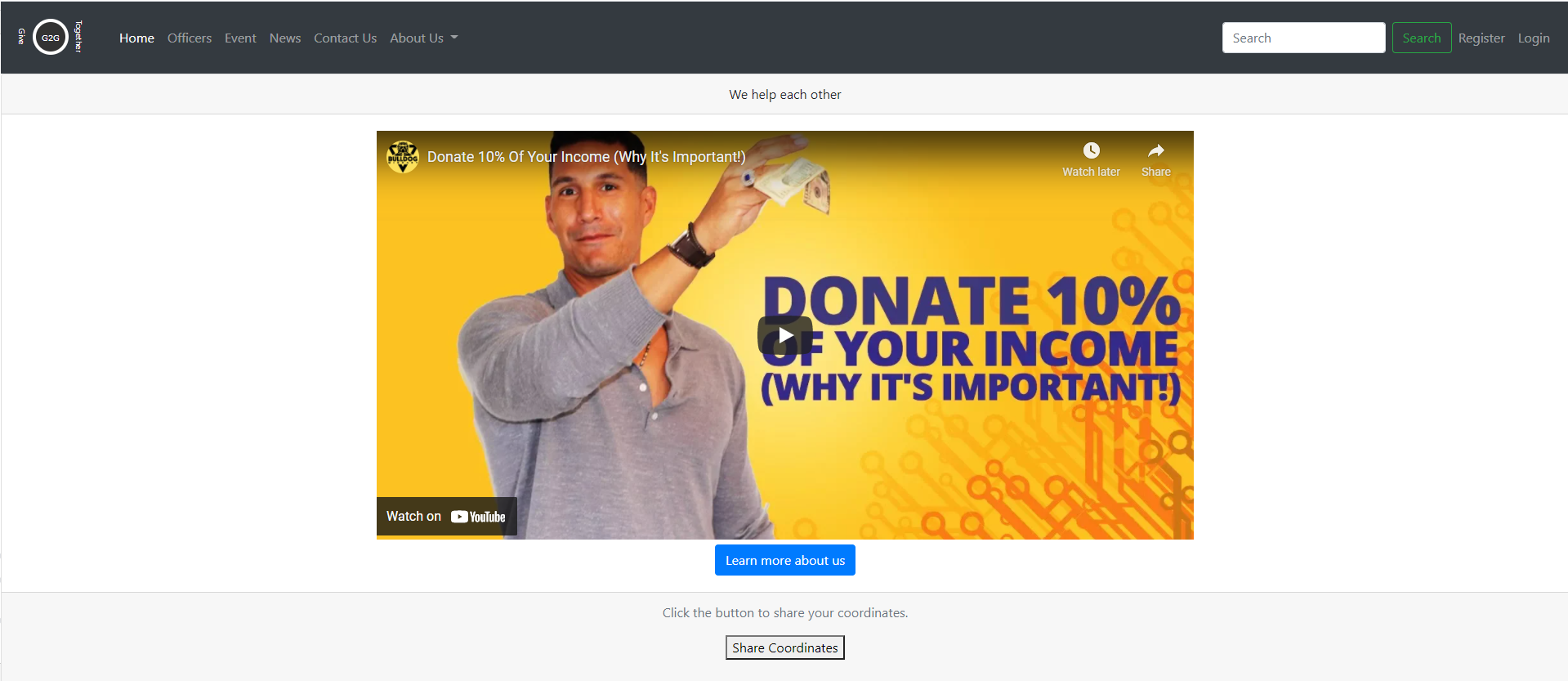


**14. I Created 3 users during my testing in my environment**

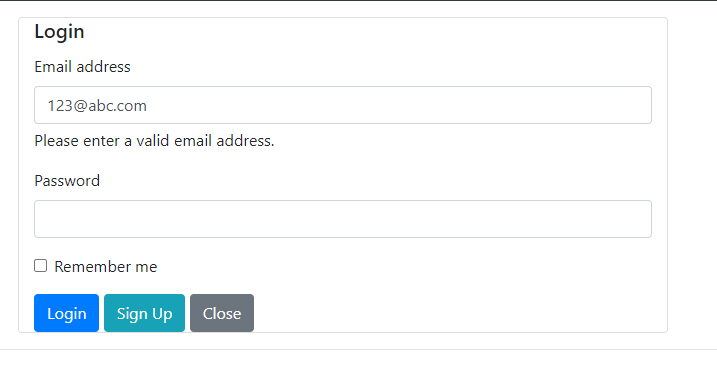


**User manual**

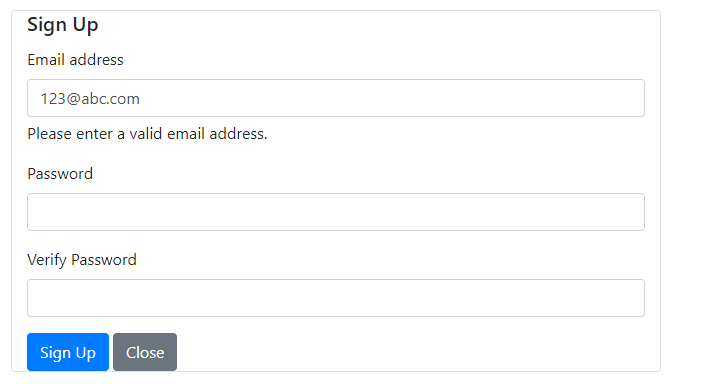
**Step1: click Home nav item**



**Step2** Click Login , use [123@abc.com](mailto:123@abc.com) as email address and 123 as password to login



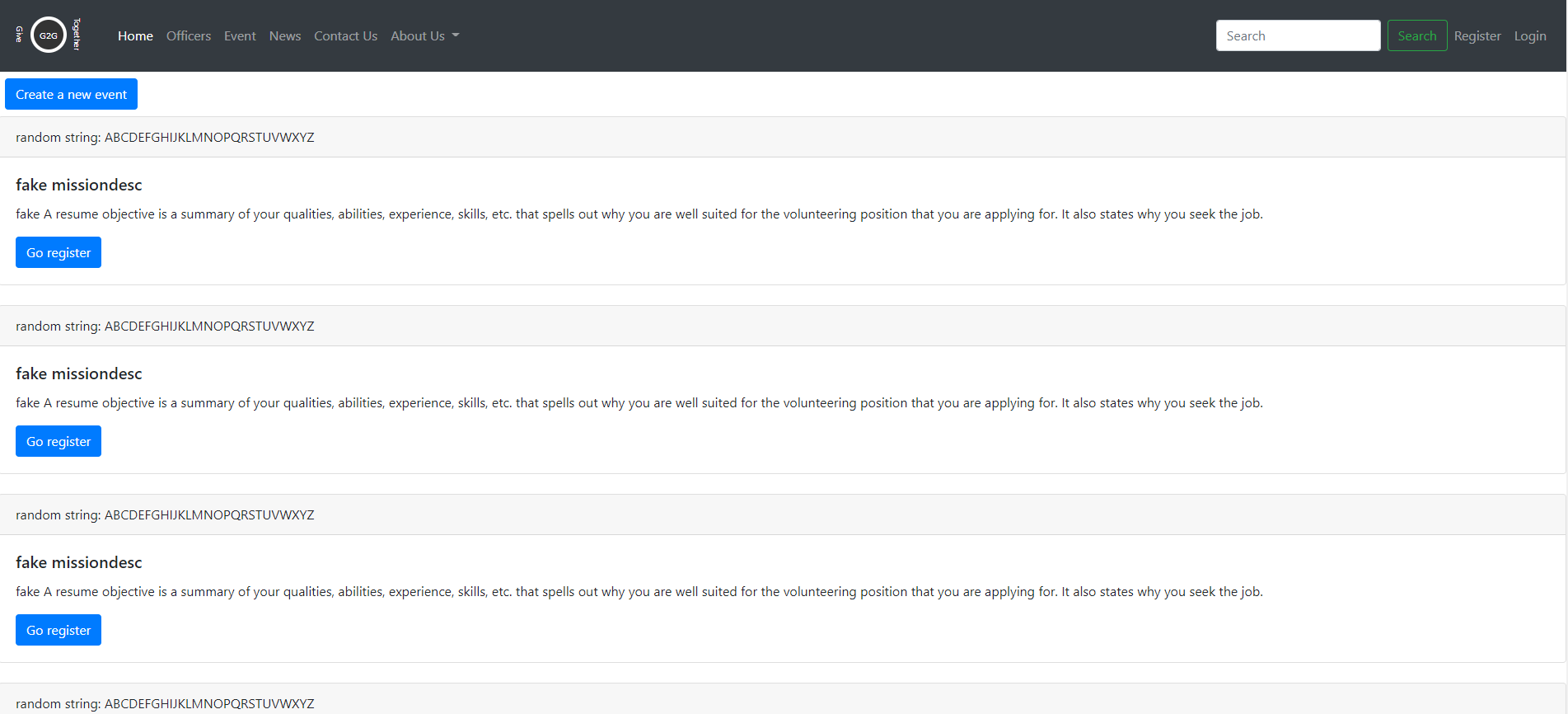
**Step3** Click Sign up to Created your own credentials.



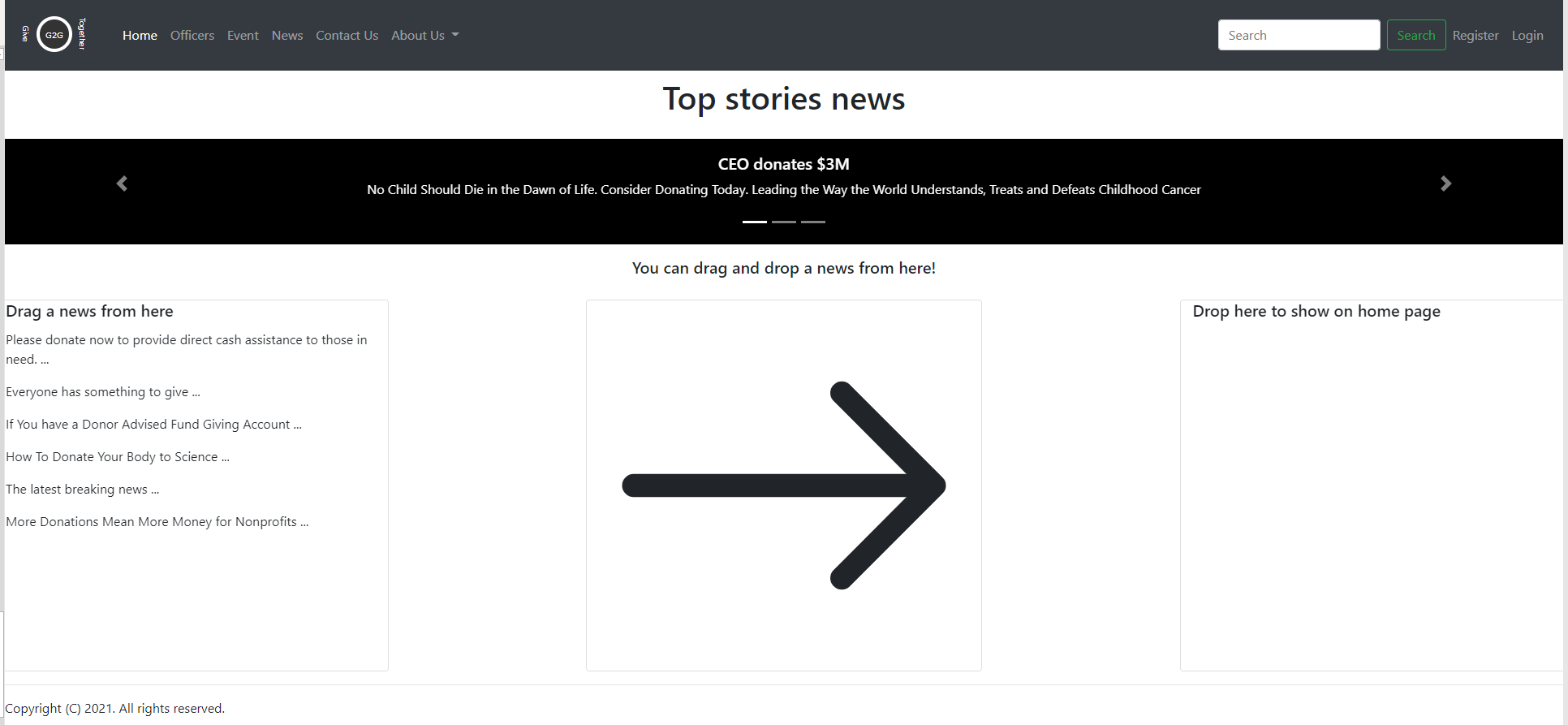
**Step4** Click officers menu item. Currently it has all fake data.



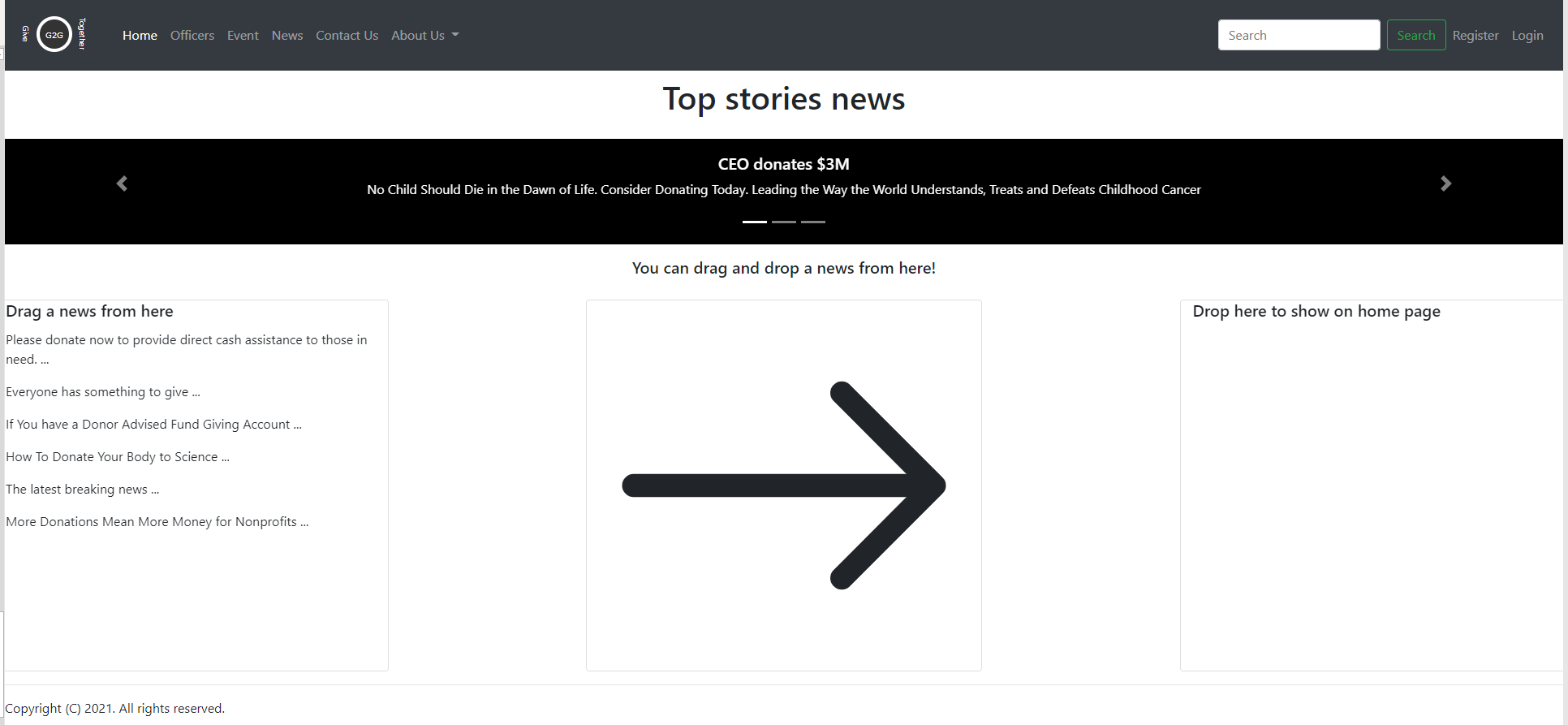
**Step 5** Click event menu item, currently it has all fake data



**Step6** Click news menu item



**Step7** Click Contact Us menu item



**Yuan’s Conclusion:**

I learned a lot from this project. I have never used Python and Django before. It’s different from many other programming languages. I like Python’s flexibility. For example, you don’t need to define the type for the variables, its type is determined by its value. It surprised me that the array can hold different types of items. It has a lot of functions supporting string manipulation, which makes string extraction and manipulation so easy. I think probably that is why a lot of machine learning and big data programing is using Python, because they have to process a lot of string and text. I also like the strict indentation rule that Python adheres to, because it makes the code very easy to read. In my first assignment, I mentioned that I hope there is an easy way to separate html files so me and my partner don’t have to step on each other. Now, with Django template inheritance, I can easily separate html pages.

What I don’t like is that it looks like we can’t directly use JQuery validation plugin with Django framework. From what I researched, Django Form validation seems to be the first choice to do the validation work. But will Django Form work with Bootstrap well? That is the improvement I am looking for.

**Edward’s Conclusion:**