**IFT458/598 – Project 1 Deliverable 4**

Yuan Li & Edward Halper

**Introduction**

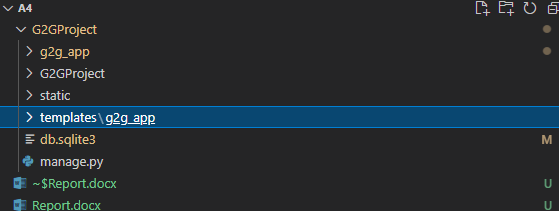
For this deliverable in the project we are tasked to use Django to take the Front End and back end deliverables to create a Django app. Using the front-end deliverables, we will separate the pages into separate html pages and put them into the templates folder of the Django app. Using different python scripts the app will be able to navigate between the html pages so that it will open and operate the same way that it did during our first deliverable in the class. We will also be taking this opportunity to fine tune any problem areas from the 1st deliverable.

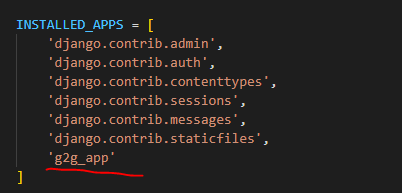
**Description of your work**

|  |  |  |
| --- | --- | --- |
| Name | Item | description |
| Yuan | Home template | 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  URL: <http://127.0.0.1:8000/> |
| Yuan | Login template | It extends the home page. Right now you can put any password and email address to pass the validation. After you click Login button you will be redirected to home page.  URL: <http://127.0.0.1:8000/login/> |
| Yuan | Officers template | 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. URL: <http://127.0.0.1:8000/officers/> |
| Yuan | Event template | 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.  URL: <http://127.0.0.1:8000/event/> |
| Yuan | Create new event template | It allows user to create a new event. It has html5 validation.  URL: <http://127.0.0.1:8000/createnewevent/> |
| Yuan | Contact Us template | It displays a static video, I had to configure static files in settings.py for the video to display. |
| Yuan | View.py | It has all the view method which is responding to each request, connecting template and fill out the template and return |
| Yuan | urls.py | It has all the url path. |
| Edward | SignUp/Register template | This has users create a username and password followed by collecting personal information to complete their profile for the site. Validation is still enabled on the page and will ask for specific inputs depending on the field. The page is tied into the login and home pages and eventually will populate the database of users. |
| Edwad | Associated validation css/js | Corrected minor discrepancies in different code to be able to use jquery and css with Django and validate the registration page. |

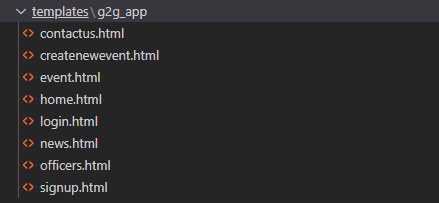
**Screenshots of code narrative from above**

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

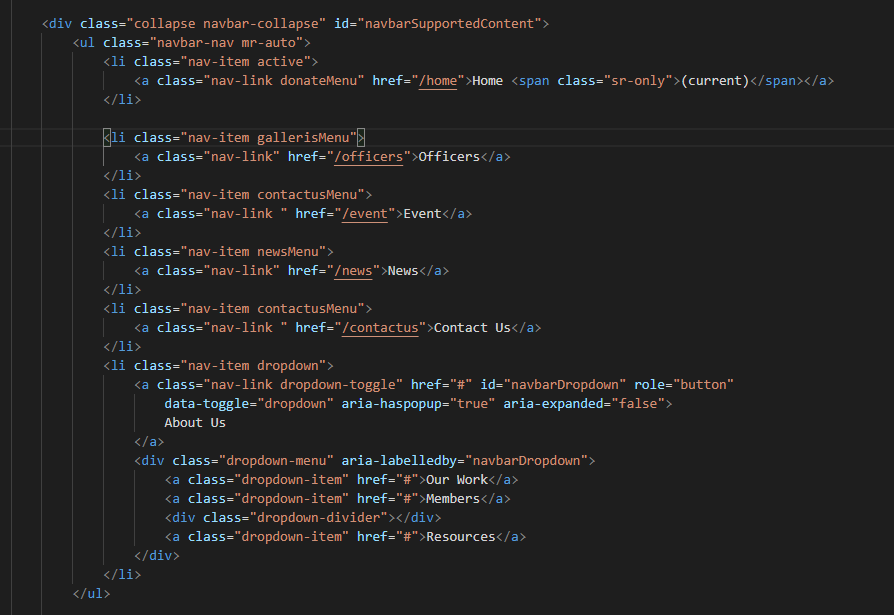




**2: Created templates, including all the html templates.**



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

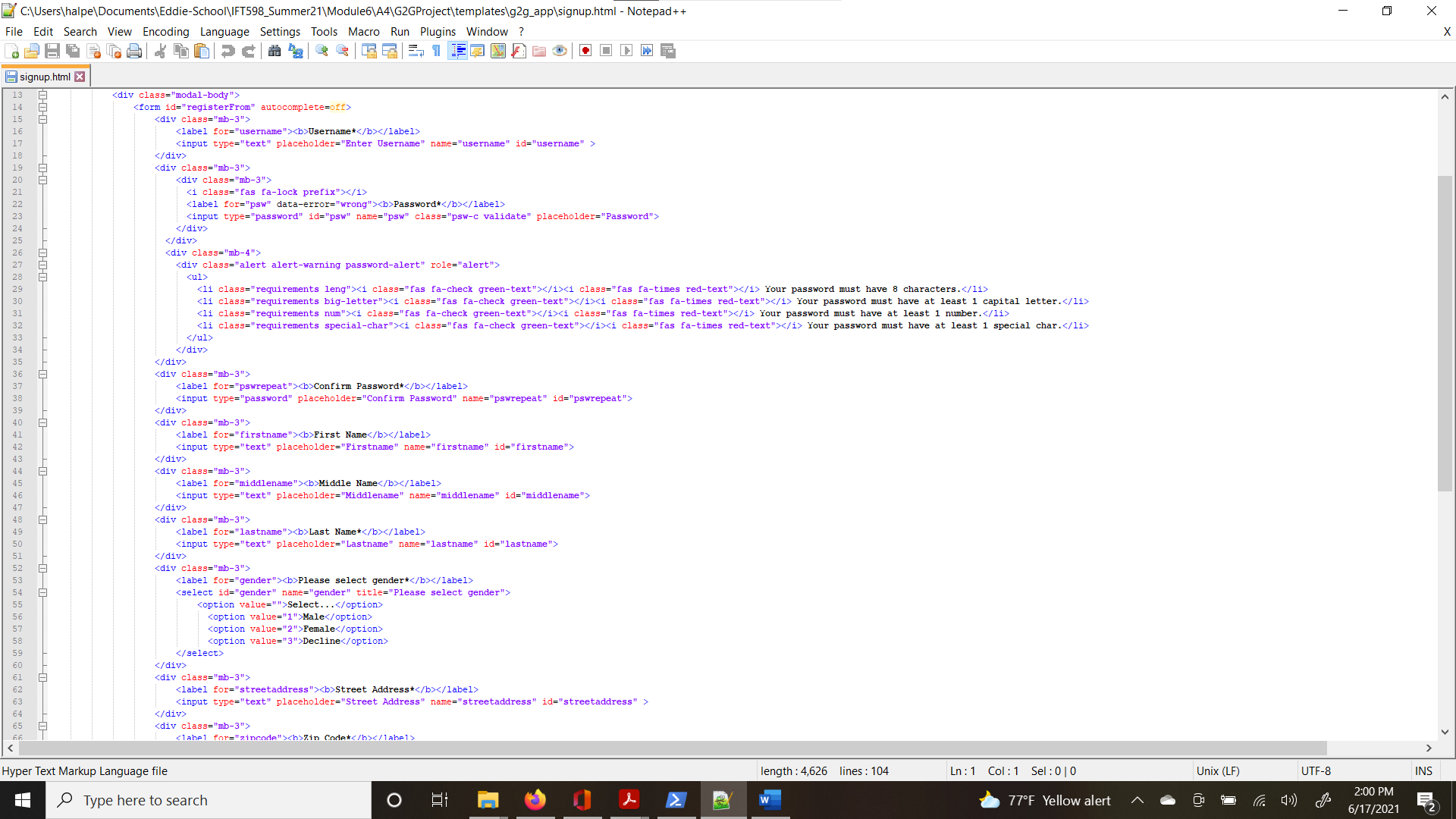




**4: Created login.html and signup.html (register)**

Text

Description automatically generated

****

**5. Created officer.html**



**6. Created event.html**

Text

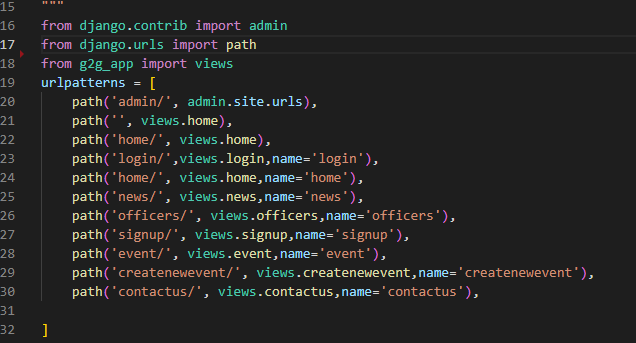
Description automatically generated

**7. Created contactus.html**

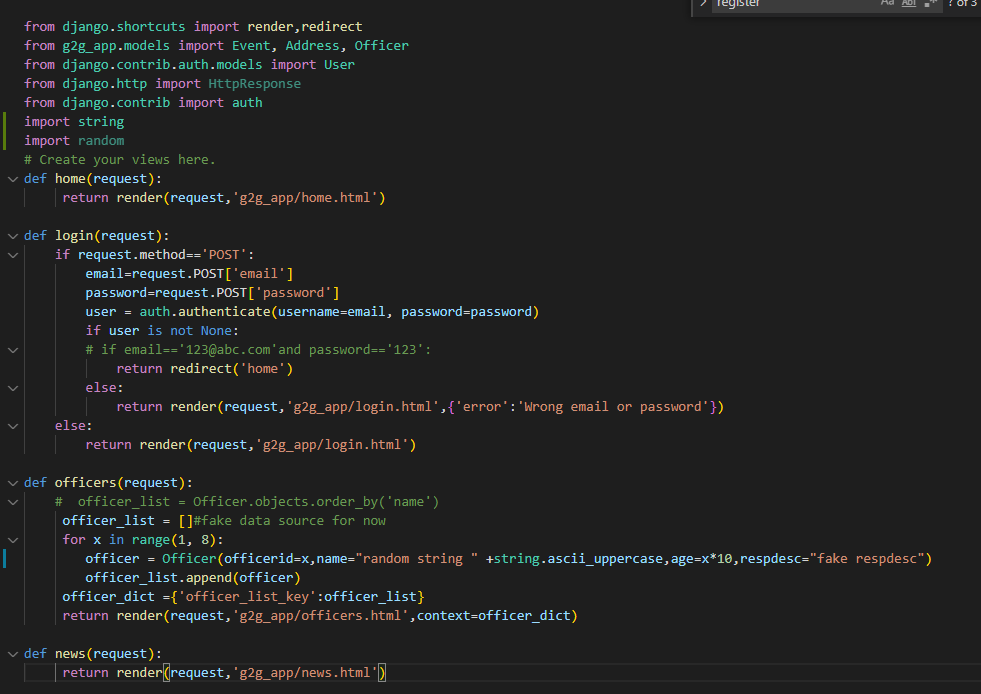
Text

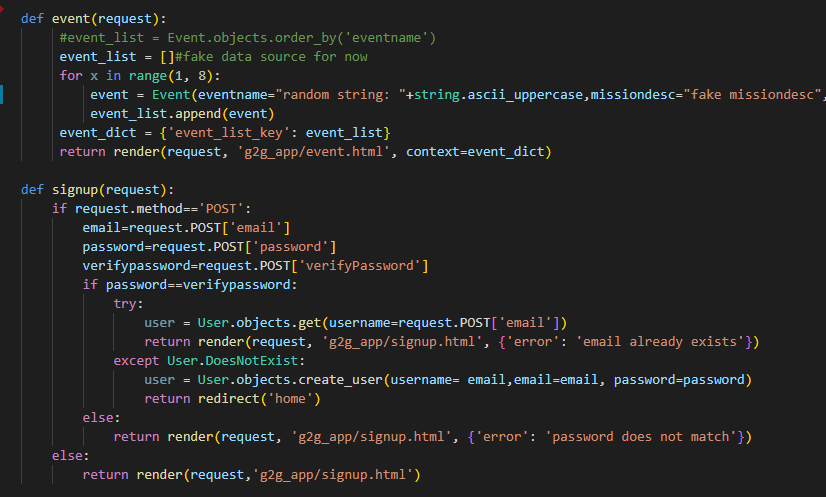
Description automatically generated

**8. config project urls.py**



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







**10. Created view method**

Text

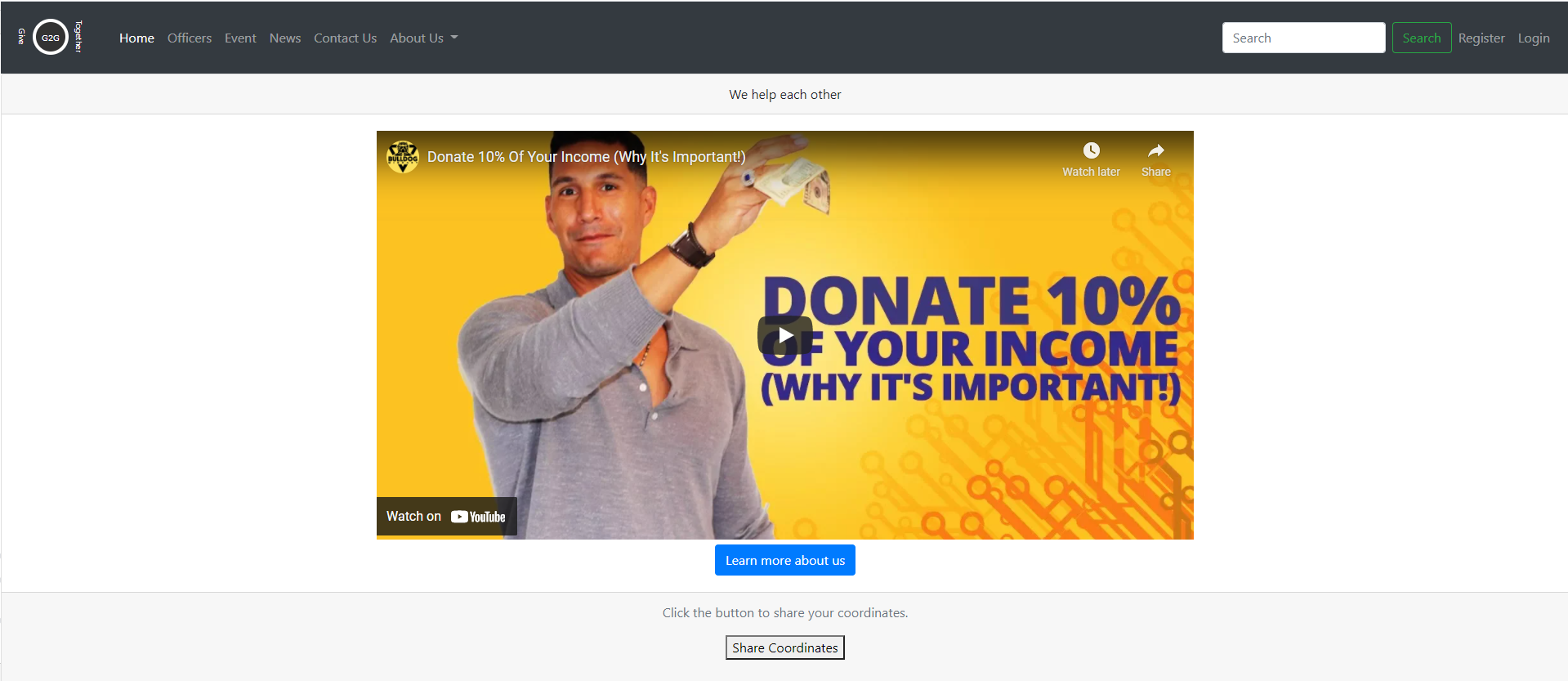
Description automatically generated

Text

Description automatically generated

**User manual**

**Step1: click Home nav item**

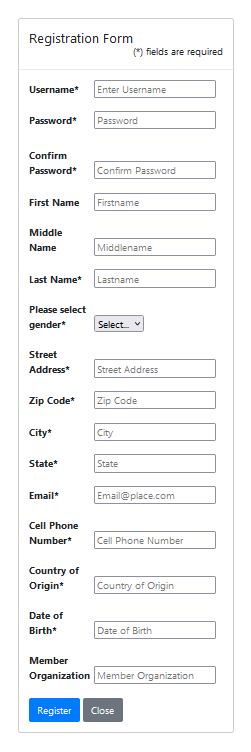


**Step2** Click Login , use [123@abc.com](mailto:123@abc.com) as default email address and any string e.g.123 as password to login

Graphical user interface, text, application, email

Description automatically generated

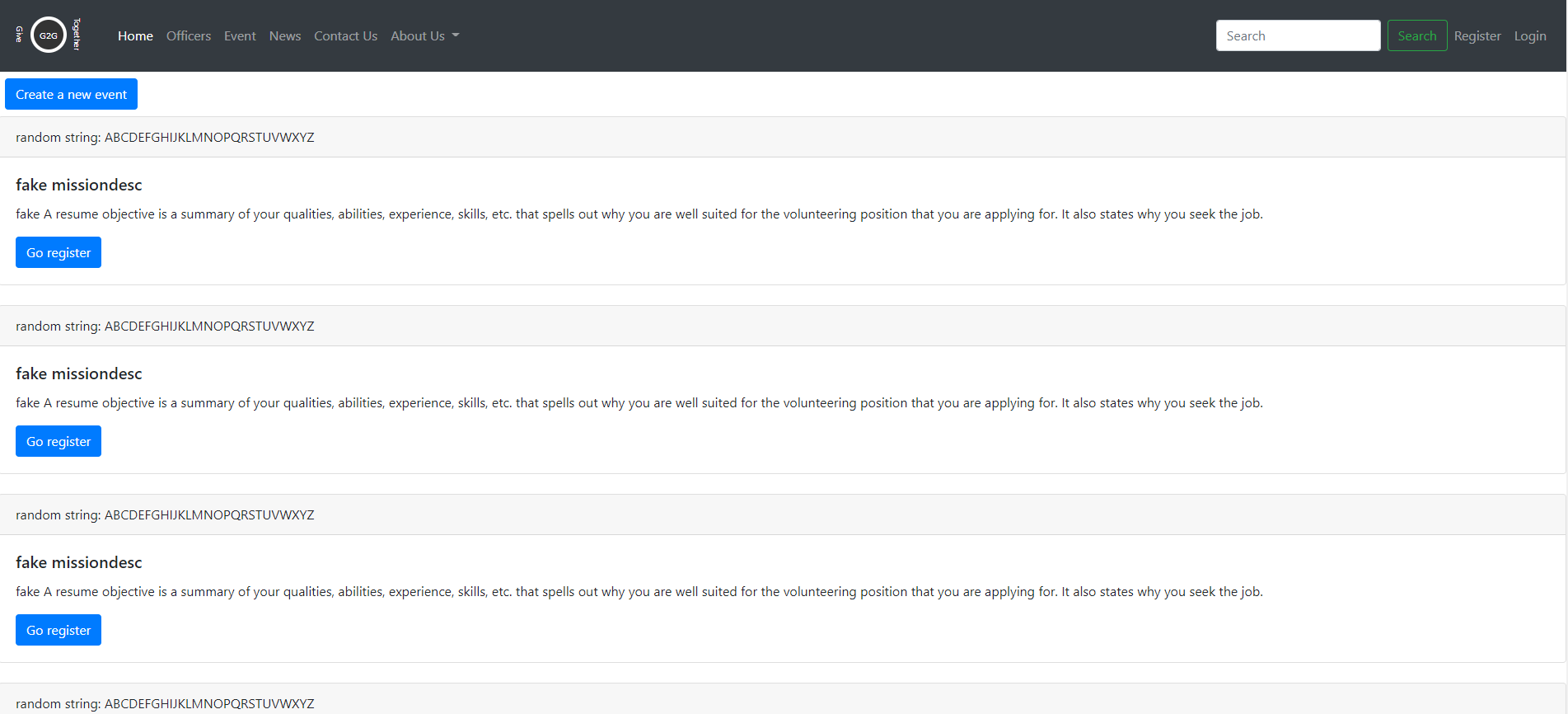
**Step3** Click Register menu item or click Register button from Login page.

****

**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

Graphical user interface

Description automatically generated

**Step7** Click Contact Us menu item

A bald eagle with its mouth open

Description automatically generated with medium confidence

**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:**

During this deliverable I was able to really do a deep dive of Django and found that it wasn’t all that bad to work with. It did take me some time to get used to it and run it on my windows system as the instructions in the class where for linux. But this allowed me to troubleshoot and really learn things that I otherwise would not have had to look up. For example after working through some exercises online I was able to take my partner’s portion of the deliverable and run it locally in order to test out the different pages and make sure everything was working properly. I was also able to incorporate the jquery and css files from our previous deliverable for the form validation which in the end I believe will really help out. For this exercise I had to run through all the same things my partner did in order to understand what had to happen to accomplish the Django app and be able to intelligently look at the deliverable and make inputs where needed to streamline our product. I believe the next deliverable will allow us to pull everything together to have a product that one could actually use.