



Script Programming/Python

CSCI-3351-01

Goon Squad Team

Bachelor's in Computer Science and Cybersecurity

Final Project Report

University of New Haven

TAGLIATELA COLLEGE OF ENGINEERING, West Haven, CT

FALL 2020

Submitted To: Prof. Reza Sadeghi

Contents

1	Phase 1	2
1.1	Team	2
1.2	Objective	2
1.3	Website Outline	2
2	Django Python and Docker Implementation	3
2.1	settings.py	3
2.2	admin.py	5
2.3	urls.py	6
2.4	asgi.py	7
2.5	views.py	7
2.6	DockerFile	8
2.7	docker-compose.yml	9
3	Phase 2	10
3.1	Web Applications	10
4	Phase 3	12
4.1	HTML Design	12
4.1.1	base.html	12
4.1.2	index.html	13
4.1.3	footer.html	14
4.1.4	navbar.html	15
4.1.5	about.html	15
4.1.6	contact.html	16
5	Phase 4	17
5.1	Postgres Database	17
5.2	Django Admin Portal	18
6	Phase 5	19
6.1	Pushing to Github	19

1 Phase 1

1.1 Team

Name of the Team: Goon Squad

Members:

1. Brandon Trundy (Team Head) - btrun1@unh.newhaven.edu
2. Steven Atilho (Team Member) - satil1@unh.newhaven.edu
3. Colin Dunn (Team Member) - cdunn5@unh.newhaven.edu
4. Andrew Mahr (Team Member) - amahr1@unh.newhaven.edu
5. Andrew Manfredi (Team Member) - amanf1@unh.newhaven.edu

1.2 Objective

The goal of our project is to create a website that can operate as a Resume/Biography platform for students. All group members have a section in the About web application to display various sections of information, such as research, professional experience, and pictures. The purpose of this website is to serve as a student's collection of personal information that can be shared to job recruiters or universities. Each aspect of the website will be split amongst several interactive web applications. Each web app or section of the web server will contain its own associated static HTML and/or CSS. A YouTube Series produced by Hacker Shack (Aaron and Davis) was used as a template for the production and development of the website.

1.3 Website Outline

1. Web-Admin Portal
 - (a) This web app will be a login portal for registered users of the website, such as admins or forum users. It will use the Django Auth app to create users and implement a login/logout page. There will also be portal views that are restricted by login status.
2. Postgres Database
 - (a) This database will contain data used throughout the web apps such as images, login usernames, password hashes, and other files. We will first create a Docker image to be able to run apps on any machine, and then we will set up a Postgres database connected to all services using a Docker Compose file.

3. Student About Page

- (a) This area will be a block of code which will present the photos and biographies of the team. The following information you will find in each user's biography includes: a brief description of each member, their experience/work experience, research they're involved in, their skillset, and courses they've taken.

4. Contact Page

- (a) This page will be a contact form through which users can contact the team for more information or submit comments about the website we have added our emails where we can be reached there.

5. Profile Page

- (a) This section will be a short admin/account portal for users of the website. It will contain an avatar picture, a short section for interests, and a short biography. In order to reach this page, the user will be prompted to log in with their username and password.

2 Django Python and Docker Implementation

2.1 settings.py

The Django settings file contains all the configuration of our Django project. The settings file is just a Python module with module-level variables. For example, `BASEDIR` points to top hierarchy of the project, whatever paths we define in the project are all relative to `BASEDIR`. To use `BASEDIR` we had to use the 'os' module provided by python. Furthermore, for the `INSTALLEDAPPS` variable section, we mention all apps that will be used in our Django project. In the `DATABASES` section, we specified the Postgres database that we integrated into our site. (dja [2020])

```

"""
Django settings for goonsquad_website project.

Generated by 'django-admin startproject' using Django 3.1.3.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.1/ref/settings/
"""

from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
PROJECT_DIR = os.path.join(BASE_DIR, "goonsquad_website")

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = "i-v%y82d-(w*1=hsjtbui8r&#m8b7n3gieixs=rtbzu0)v%a%5"

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "goonsquad_website.apps.accounts",
    "goonsquad_website.apps.contact",
]

MIDDLEWARE = [
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "django.contrib.auth.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
]

ROOT_URLCONF = "goonsquad_website.urls"

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [os.path.join(PROJECT_DIR, "templates")],
        "APP_DIRS": True,
        "OPTIONS": {
            "context_processors": [
                "django.template.context_processors.debug",
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
            ],
        },
    },
]

WSGI_APPLICATION = "goonsquad_website.wsgi.application"

# Database
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

POSTGRES_HOST = os.environ.get("POSTGRES_HOST", default="")
POSTGRES_DB = os.environ.get("POSTGRES_DB", default="postgres")
POSTGRES_USER = os.environ.get("POSTGRES_USER", default="")
POSTGRES_PASSWORD = os.environ.get("POSTGRES_PASSWORD", default="")

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql",
        "NAME": POSTGRES_DB,
        "USER": POSTGRES_USER,
        "PASSWORD": POSTGRES_PASSWORD,
        "HOST": POSTGRES_HOST,
        "PORT": 5432,
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator",
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
    },
    {
        "NAME": "django.contrib.auth.password_validation.CommonPasswordValidator",
    },
    {
        "NAME": "django.contrib.auth.password_validation.NumericPasswordValidator",
    },
]

# Internationalization
# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE_CODE = "en-us"

```

```
# Internationalization
# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE_CODE = "en-us"

TIME_ZONE = "UTC"

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC_URL = "/static/"
STATICFILES_DIRS = [os.path.join(BASE_DIR, "static")]


# Django Auth Settings
LOGIN_URL = "accounts:login"
LOGIN_REDIRECT_URL = "accounts:profile"
LOGOUT_REDIRECT_URL = "public:index"

EMAIL_BACKEND = "django.core.mail.backends.console.EmailBackend"
DEFAULT_FROM_EMAIL = "goonsquadwebsite@gmail.com"
```

2.2 admin.py

The admin.py file reads metadata from the project's models and provides an interface wherein "superusers" can manage content on the website. The capabilities of this interface are limited, and it should not be used to build the site's entire front-end. After specifying the URL path of the admin page, it can be accessed through the website. It can then be decided which of the app's models can be edited through the admin interface.

```
from django.contrib import admin

from .models import UserProfile, UserPersona, UserInterest

admin.site.register(UserProfile)
admin.site.register(UserPersona)
admin.site.register(UserInterest)
```

2.3 urls.py

URLS.py tells Django to search for URL patterns in the file books/urls.py. It also converts Python functions into a response that the HTTP can handle, like a 404. Every person who views the website needs to be handled by the Python module called URLConf which handles the way the site views your URL connections.

```

"""goonsquad_website URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

# from django.contrib.auth import views as auth_views

# from django.views.generic.base import TemplateView

from . import views

urlpatterns = [
    path("admin/", admin.site.urls),
    path("", include("goonsquad_website.apps.public.urls")),
    # path('accounts/profile', views.ProfileView.as_view(), name="profile"),
    path("accounts/", include("goonsquad_website.apps.accounts.urls")),
    path("contact/", include("goonsquad_website.apps.contact.urls"))
    # This is Django Auth stuff
]

# Differnt Django Views
# accounts/login/ [name='login']
# accounts/logout/ [name='logout']
# accounts/password_change/ [name='password_change']
# accounts/password_change/done [name='password_change_done']
# accounts/password_reset [name='password_reset']
# accounts/password_reset/done [name='password_reset_done']
# accounts/reset/<uidb64>/<token>/ [name='password_reset_confirm']
# accounts/reset/done [name='password_reset_complete']

```

2.4 asgi.py

The ASGI python file has us supply an application call which the application server uses to communicate with your code. It's commonly provided as an object named application in a Python module accessible to the server. When the ASGI server loads our application, Django needs to import the settings module, where our entire application is defined.

```
ASGI config for goonsquad_website project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "goonsquad_website.settings")

application = get_asgi_application()
```

2.5 views.py

Views.py is a function that gets an HTTP request and returns an HTTP response. This HTTP response is what shows up on a web page. An example of what the HTTP response could return HTML code for the web page.

```
from django.shortcuts import render

# Create your views here.

from django.http import HttpResponse, HttpRequest

# from django.template import loader

def index(request):
    print(request.user)
    return render(request, "index.html")

def about(request):
    return render(request, "about.html")

# def contact(request):
#     return render(request, "contact.html")
```


2.6 DockerFile

The Dockerfile is a text doc containing all the shell commands we could execute on the CLI to assemble an image of our containerized website. Using the docker build command we can create an automatic build that executes several CLI instructions in order. The docker build command builds an image from a Dockerfile and a context. The build's context is the set of files at a specified location PATH or URL. The PATH is a directory on your local file system. The URL is a Git repository location. The Docker daemon runs the instructions in the Dockerfile one-by-one, committing the result of each instruction to a new image if necessary, before finally outputting the ID of your new image. doc [2020]

```
FROM python:3.9.0-slim as production

ENV PYTHONUNBUFFERED=1
WORKDIR /app/

RUN apt-get update && \
    apt-get install -y \
    bash \
    build-essential \
    gcc \
    libffi-dev \
    musl-dev \
    openssl \
    postgresql \
    libpq-dev

COPY requirements/prod.txt ./requirements/prod.txt
RUN pip3 install -r ./requirements/prod.txt

COPY manage.py ./manage.py

COPY setup.config ./setup.config
COPY goonsquad_website ./goonsquad_website

EXPOSE 8000

FROM production as development

COPY requirements/dev.txt ./requirements/dev.txt
RUN pip3 install -r ./requirements/dev.txt
COPY . .
```

2.7 docker-compose.yml

The docker-compose.yml file describes the services that make our website. In our project, those services are a Django web server, and a Postgres database. The compose file also describes which Docker images these services use, how they link together, and any volumes they might need to be mounted inside the containers. Finally, this file describes which ports these services expose.

```
version: "3.7"

x-service-volumes: &service-volumes
  - ./:/app/:rw,cached

x-database-variables: &database-variables
  POSTGRES_DB: postgres
  POSTGRES_USER: postgres
  POSTGRES_PASSWORD: postgres

x-app-variables: &app-variables
  <<: *database-variables
  POSTGRES_HOST: postgres

services:
  website:
    image: goonsquad_website:latest
    command: python3 manage.py runserver 0.0.0.0:8000
    volumes: *service-volumes
    environment: *app-variables
    depends_on:
      - db_migrate
    ports:
      - "8000:8000"

  db_migrate:
    image: goonsquad_website:latest
    command: python3 manage.py migrate
    volumes: *service-volumes
    environment: *app-variables
    depends_on:
      - postgres

  postgres:
    image: postgres
    ports:
      - "5432:5432"
    environment: *database-variables
    volumes:
      - db-data:/var/lib/postgresql/data

volumes:
  db-data:
```

3 Phase 2

3.1 Web Applications

1. Web-Admin Portal

GOON SQUAD WEBSITE
HOME ABOUT CONTACT LOGIN

Login

➡

Username:

Password:

Login

Reach Us @
goonsquadwebsite@gmail.com


Quick Links
[Home](#) [About](#) [Contact](#)

Last Update - December 2020

2. Student About Page

GOON SQUAD WEBSITE
HOME ABOUT CONTACT LOGIN

Steven Atilho



Bio

University of New Haven hacking team member with skills in ethical hacking, system administration and networking. Junior undergraduate pursuing a bachelor's degree in Computer Science at UNH, with research interests in the areas of cybersecurity and AI. I am a self-motivated and passionate leader seeking to build the foundation for a career in the fields of DevSecOps and ITSec.

Experience

UNH Hacking Team:

- Compete in penetration testing and defense competitions at the local and national level (CPTC, CCDC, ISTS).
- Improve essential cybersecurity skills such as scripting, networking, ethical hacking, and system administration.
- Host weekly hacking club meetings offering educational presentations, and hands-on CTFs or training sessions.

TechLot Services LLC:

- Design, build, host and maintain web/email/database servers for clients.
- Diagnose and repair software/hardware issues on Windows devices.
- Perform certified Apple iPhone 5 - X/XR/XS smartphone repair.

Research

ONR Cyber Operative Research Scholars Program (CORES):

3. Contact Page

GOON SQUAD WEBSITE

HOMEABOUTCONTACTLOGIN

CONTACT

Welcome to our website! If you have questions or need more information, please email us here!

Andrew Mahr: amahr1@unh.newhaven.edu

Steven Atilho: satil1@unh.newhaven.edu

Brandon Trundy: btrun1@unh.newhaven.edu

Andrew Manfredi: amanf1@unh.newhaven.edu

Colin Dunn: cdunn5@unh.newhaven.edu

Name

Email

Message

Send

Reach Us @
goonsquadwebsite@gmail.com

Quick Links
[Home](#) [About](#) [Contact](#)

<mailto:satil1@unh.newhaven.edu>

4. Profile Page

GOON SQUAD WEBSITE

HOMEABOUTCONTACTPROFILELOGOUT

Andrew Manfredi
(@andrewmanfredi)

<https://www.linkedin.com/in/andrew-manfredi>

Interests: Guitar, Music, Philosophy, Psychology, Working Out

Andrew Manfredi is a 20-year-old University of New Haven Cybersecurity major also pursuing a minor in Spanish.

Reach Us @
goonsquadwebsite@gmail.com

Quick Links
[Home](#) [About](#) [Contact](#)

Last Update - December 2020

4 Phase 3

4.1 HTML Design

In order to build the website interface and create the real design of the website, the team used templates and styles from Bootstrap and its Freelancer template (boo [2020]). The team made modifications and changes to style and colors to fit the specifications desired for the website. In order to use the templates within the project, the team referenced the HTML and CSS code found within the template.

4.1.1 base.html

This html file deals with the base for every page of the website. This is where the header and footer are declared, and all the necessary styling and css information written in HTML.

```

Loading file tree... > goonsquad_website > templates > base.html
1  {% load static %}
2
3  <!-- templates/base.html -->
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <meta charset="utf-8">
8      <title>{% block title %}Goon Squad Website{% endblock %}</title>
9
10     <meta charset="utf-8" />
11     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
12     <meta name="description" content="" />
13     <meta name="author" content="" />
14
15     <!-- Font Awesome icons (free version)-->
16     <script src="https://use.fontawesome.com/releases/v5.13.0/js/all.js" crossorigin="anonymous"></script>
17     <!-- Google fonts-->
18     <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css" />
19     <link href="https://fonts.googleapis.com/css?family=Lato:400,700,400italic,700italic" rel="stylesheet" type="text/css" />
20     <!-- Core theme CSS (includes Bootstrap)-->
21     <link href="{% static 'theme/css/styles.css' %}" rel="stylesheet" />
22
23     <!-- Internal CSS -->
24     <link href="{% static 'css/main.css' %}" rel="stylesheet" />
25
26     {% block head %}
27     {% endblock %}
28
29 </head>
30 <body>
31     {% include 'navbar.html' %}
32     <main id="main">
33         {% block content %}
34         {% endblock %}
35     </main>
36
37     <!-- Bootstrap core JS-->
38     <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
39     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"></script>
40     <!-- Third party plugin JS-->
41     <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.4.1/jquery.easing.min.js"></script>
42     <!-- Core theme JS-->
43     <script src="{% static 'theme/js/scripts.js' %}"></script>
44
45 </body>
46 {% include 'footer.html' %}
47 </html>

```

4.1.2 index.html

This html file deals with the content displayed on the homepage. It has a brief overview of the website and the project itself.

```
{% extends 'base.html' %}
{% load static %}
<link href="{% static 'css/profile.css' %}" rel="stylesheet" />

{% block title %}Gandalf&#8217s Goon Squad Fan Page{% endblock %}

{% block content %}
{% comment %}  {% endcomment %}

<header class="masthead text-white text-center">
  <div class="container d-flex align-items-center flex-column">
    <!-- Masthead Avatar Image-->
    {% comment %}  {% endcomment %}
    <!-- Masthead Heading-->
    <h1 class="masthead-heading text-uppercase mb-0">Final Project CSCI-3351-01</h1>
  </div>
</header>

<br/>

<h1><center> Home </center></h1>

<div class="divider-custom">
<div class="divider-custom-line"></div>
<div class="divider-custom-icon"><i class="fas fa-home"></i></div>
<div class="divider-custom-line"></div>
</div>

<section class="page-section">
  <div class="container">
    <div class="text-center text-secondary mb-0">
      <p>
        This website was created for a project in a Python/Script Programming course at the University of New Haven.
        There are several web applications throughout this website, to meet the objective of a personal student website
        for the members of the group. You can navigate throughout the web apps by utilizing the navigation bar at the top
        of the webpage. The about webapp will contain a resume style biography containing various information about each
        student in the group, such as research and work experience. The contact page will send an email to the group, that
        may be used by job recruiters or other individuals/groups interested in contacting us. The profile page will show
        personalized information such as interests, about each student in the group. There is also a login/logout web app
        that allows for the users of the website to change their profile information, or to access the admin panel.
      </p>
    </div>
  </div>
</section>
```

4.1.3 footer.html

This html file deals with the content displayed at the bottom of every page. It includes links to the important pages of the website and displays the team email created for the project.

```
{% url 'public:index' as index_url %}
{% url 'public:about' as about_url %}
{% url 'contact:contact' as contact_url %}
{% url 'accounts:profile' as profile_url %}

<!-- Footer-->
<footer class="footer text-center">
  <div class="container">
    <div class="row">
      <!-- Footer Location-->
      <div class="col-lg-4 mb-5 mb-lg-0">
        <p class="lead mb-0">
          Reach Us @
          <br/>
          <a href="mailto:goonsquadwebsite@gmail.com">goonsquadwebsite@gmail.com</a>
        </p>
      </div>
      <!-- Footer Social Icons-->

      <div class="col-lg-4 mb-5 mb-lg-0">
        <!-- Social icons and buttons can go here-->
      </div>

      <!-- Footer About Text-->
      <div class="col-lg-4">
        <p class="lead mb-0 w-100">
          Quick Links
          <br/>
          <a class="px-2 {% if request.path == index_url %}active{% endif %}" href="{% index_url %}">Home</a>
          <a class="px-2 {% if request.path == about_url %}active{% endif %}" href="{% about_url %}">About</a>
          <a class="px-2 {% if request.path == contact_url %}active{% endif %}" href="{% contact_url %}">Contact</a>
        </p>
      </div>
    </div>
  </div>
</footer>

<!-- Copyright Section-->
<div class="copyright py-4 text-center text-white">
  <div class="container"><small>Last Update - December 2020</small></div>
</div>

<!-- Scroll to Top Button (Only visible on small and extra-small screen sizes)-->
<div class="scroll-to-top d-lg-none position-fixed">
  <a class="js-scroll-trigger d-block text-center text-white rounded" href="#page-top"><i class="fa fa-chevron-up"></i></a>
</div>
```

4.1.4 navbar.html

This html file deals with the creation of the navbar at the top of every page. The navbar contains the team name, and buttons with links to the important pages, login, and other information.

```
{% url 'contact:contact' as contact_url %}
{% url 'public:index' as index_url %}
{% url 'public:about' as about_url %}
{% url 'accounts:profile' as profile_url %}
{% url 'accounts:login' as login_url %}
{% url 'accounts:logout' as logout_url %}

<nav class="navbar navbar-expand-lg bg-secondary text-uppercase fixed-top" id="mainNav">
  <div class="container">
    <a class="navbar-brand js-scroll-trigger" href="{{ index_url }}">Goon Squad Website</a>
    <button class="navbar-toggler navbar-toggler-right text-uppercase font-weight-bold gs-bg-primary text-white rounded" type="button" data-toggle="collapse" data-target="#navbarResponsive"
      aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
      Menu
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded js-scroll-trigger" {% if request.path == index_url %}active{% endif %}" href="{{ index_url }}">Home</a></li>
        <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded js-scroll-trigger" {% if request.path == about_url %}active{% endif %}" href="{{ about_url }}">About</a></li>
        <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded js-scroll-trigger" {% if request.path == contact_url %}active{% endif %}" href="{{ contact_url }}">Contact</a></li>
        {% if user.is_authenticated %}
        <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 js-scroll-trigger" {% if request.path == profile_url %}active{% endif %}" href="{{ profile_url }}">Profile</a></li>
        <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 js-scroll-trigger" {% if request.path == logout_url %}active{% endif %}" href="{{ logout_url }}">Logout</a></li>
        {% else %}
        <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 js-scroll-trigger" {% if request.path == login_url %}active{% endif %}" href="{{ login_url }}">Login</a></li>
        {% endif %}
      </ul>
    </div>
  </div>
</nav>
```

4.1.5 about.html

This html page contains all of the groups information. This information includes pictures, biographies, skillsets, experience and anything more that team members wished to include.

```
{% extends 'base.html' %}
{% load static %}
<link href="{% static 'css/profile.css' %}" rel="stylesheet" />

{% block title %}Gandalf&#8217;s Goon Squad Fan Page{% endblock %}

{% block content %}
   {% endcomment %}

  <header class="masthead text-white text-center">
    <div class="container d-flex align-items-center flex-column">
      <!-- Masthead Avatar Image -->
       {% endcomment %}
      <!-- Masthead Heading -->
      <h1 class="masthead-heading text-uppercase mb-0">Final Project CSCI-3351-01</h1>
    </div>
  </header>

  <br>

  <h1><center> Home </center></h1>

  <div class="divider-custom">
    <div class="divider-custom-line"></div>
    <div class="divider-custom-icon"><i class="fas fa-home"></i></div>
    <div class="divider-custom-line"></div>
  </div>

  <section class="page-section">
    <div class="container">
      <div class="text-center text-secondary mb-0">
        <p>
          This website was created for a project in a Python/Script Programming course at the University of New Haven. There are several web applications throughout this website, to meet the objective of a personal student website for the members of the group. You can navigate throughout the web apps by utilizing the navigation bar at the top of the webpage. The about webapp will contain a resume style biography containing various information about each student in the group, such as research and work experience. The contact page will send an email to the group, that may be used by job recruiters or other individuals/groups interested in contacting us. The profile page will show personalized information such as interests, about each student in the group. There is also a login/logout web app that allows for the users of the website to change their profile information, or to access the admin panel.
        </p>
      </div>
    </div>
  </section>
</div>
```


4.1.6 contact.html

This html page contains a contact form from which users could contact the team for access to the website or to notify the team of any issues with the website.

```
{% extends 'base.html' %}

{% block title %}Contact{% endblock %}

{% block content %}
<section class="page-section">
  <div class="container">
    <!-- Contact Section Heading-->
    <h2 class="page-section-heading text-center text-uppercase text-secondary mb-0">Contact</h2>
    <!-- Icon Divider-->
    <div class="divider-custom">
      <div class="divider-custom-line"></div>
      <div class="divider-custom-icon"><i class="fas fa-envelope"></i></div>
      <div class="divider-custom-line"></div>
    </div>

    <div class="text-center text-secondary mb-0">
      <p>
        Welcome to our website! If you have questions or need more information, please email us here!
      </p>
      <p>
        Andrew Mahr: <a href="mailto:amahr1@unh.newhaven.edu">amahr1@unh.newhaven.edu</a>
      </p>
      <p>
        Steven Atilho: <a href="mailto:satil1@unh.newhaven.edu">satil1@unh.newhaven.edu</a>
      </p>
      <p>
        Brandon Trundy: <a href="mailto:btrun1@unh.newhaven.edu">btrun1@unh.newhaven.edu</a>
      </p>
      <p>
        Andrew Manfredi: <a href="mailto:amanf1@unh.newhaven.edu">amanf1@unh.newhaven.edu</a>
      </p>
      <p>
        Colin Dunn: <a href="mailto:cdunn5@unh.newhaven.edu">cdunn5@unh.newhaven.edu</a>
      </p>
    </div>

    <!-- Contact Section Form-->
    <div class="row">
      <div class="col-lg-8 mx-auto">

        {% if success %}
        <div class="alert alert-success" role="alert">
          Your message was sent! Check your inbox for a response.
        </div>
        {% endif %}

      </div>
    </div>
  </div>
</section>
{% endblock %}
```

```
{% endif %}

<!-- To configure the contact form email address, go to mail/contact_me.php and update the email address in the PHP file on line 19.-->
<form method="post">
  {% csrf_token %}

  {% if form.non_field_errors %}
  <div class="alert alert-danger" role="alert">
    {{ form.non_field_errors }}
  </div>
  {% endif %}

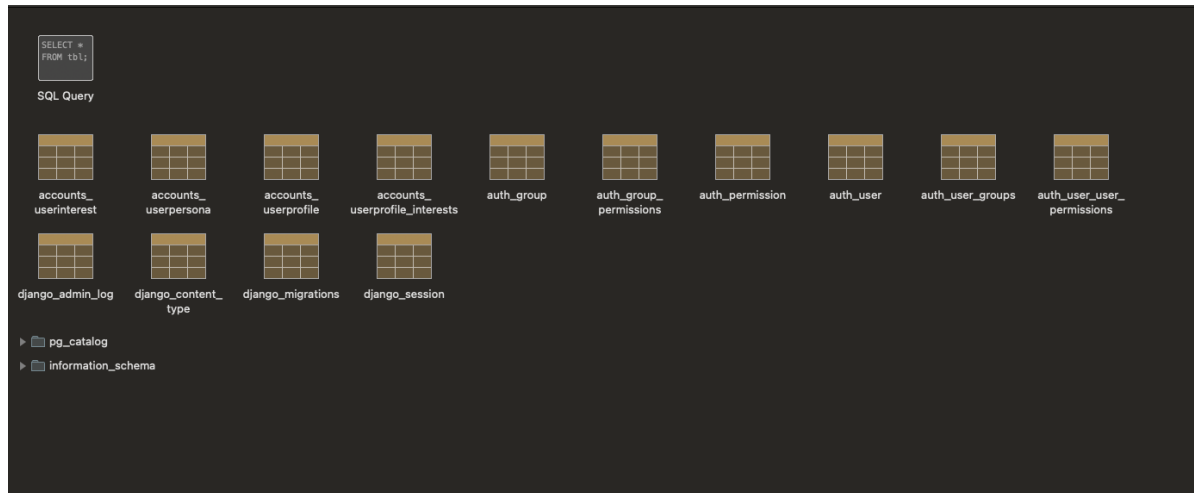
  {% for field in form %}
  <div class="control-group">
    <div class="form-group floating-label-form-group controls mb-0 pb-2">
      {{ field.label_tag }}
      {{ field }}
      {% for error in field.errors %}
      <p class="help-block text-danger">{{ error }}</p>
      {% endfor %}
    </div>
  </div>
  {% endfor %}

  <br />
  <div class="form-group"><button class="btn gs-bg-primary btn-xl text-white" type="submit">Send</button></div>
</form>
</div>
</div>
</section>
{% endblock %}
```

5 Phase 4

5.1 Postgres Database

The Postgres database contains the information about the users, passwords, and other data used within the website. Below is a picture of the different tables within the Postgres database. (pos [2020])



5.2 Django Admin Portal

This is the Django admin portal where administrators can add or remove users, profiles, interests and other database information.

Django administration WELCOME, ANDREW VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

ACCOUNTS

- User interests [+ Add](#) [Change](#)
- User personas [+ Add](#) [Change](#)
- User profiles [+ Add](#) [Change](#)

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#) [Change](#)
- Users [+ Add](#) [Change](#)

Recent actions

My actions

- UserProfile object (2) User profile
- + UserProfile object (5) User profile
- UserProfile object (1) User profile
- UserProfile object (3) User profile
- UserProfile object (3) User profile
- + Working Out User interest
- + Psychology User interest
- + Philosophy User interest
- + Music User interest
- + Guitar User interest

Django administration WELCOME, ANDREW VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Accounts > User interests

ACCOUNTS

- User interests [+ Add](#)
- User personas [+ Add](#)
- User profiles [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

Select user interest to change [ADD USER INTEREST +](#)

Action: 0 of 16 selected

- ☐ USER INTEREST
- ☐ Working Out
- ☐ Psychology
- ☐ Philosophy
- ☐ Music
- ☐ Guitar
- ☐ Hockey
- ☐ Running
- ☐ Hunting
- ☐ Gaming
- ☐ Coding
- ☐ Fishing
- ☐ Hiking
- ☐ Hacking Club
- ☐ Go-Karting
- ☐ Frisbee
- ☐ Golf

16 user interests

6 Phase 5

6.1 Pushing to Github

Once we completed the source code for our website, we pushed the entire folder onto a Github repository that was shared among all the group members, and the professor for collaboration purposes. This allows anyone to clone the repository onto their device and commit any necessary changes. (git [2020])

The screenshot shows a GitHub repository page for 'smatilho / CSCI3351_websiteProject'. The repository is private and has 1 star, 0 forks, and 0 watchers. The main branch is 'main'. The repository contains two files: 'code' (This is the code for our website, 20 minutes ago) and 'README.md' (Update README.md, 1 minute ago). The README.md file is displayed, showing the title 'CSCI3351_websiteProject' and a description: 'This repository contains all the source code for our Django python project website. It contains various CSS, HTML, python and javascript files necessary for the website to run and be designed.' The right sidebar shows the 'About' section with no description, 'Releases' section with no releases published, 'Packages' section with no packages published, 'Contributors' section with 2 contributors (smatilho Steven Atilho and acompsec1), and 'Languages' section showing a bar chart of file types: CSS 59.2%, HTML 22.6%, JavaScript 13.1%, Python 4.8%, and Other 0.3%.

smatilho / CSCI3351_websiteProject Private
Unwatch 1 Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

main CSCI3351_websiteProject / code / Go to file Add file

acompsc1 This is the code for our website.
 6415411 30 minutes ago [History](#)

..		
.vscode	This is the code for our website.	30 minutes ago
goonsquad_website	This is the code for our website.	30 minutes ago
requirements	This is the code for our website.	30 minutes ago
static	This is the code for our website.	30 minutes ago
.DS_Store	This is the code for our website.	30 minutes ago
.python-version	This is the code for our website.	30 minutes ago
Dockerfile	This is the code for our website.	30 minutes ago
Makefile	This is the code for our website.	30 minutes ago
docker-compose.yml	This is the code for our website.	30 minutes ago
dump	This is the code for our website.	30 minutes ago
manage.py	This is the code for our website.	30 minutes ago
setup.config	This is the code for our website.	30 minutes ago

Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Actions
Secrets

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

[Manage](#)

DIRECT ACCESS

5 have access to this repository. 4 collaborators. 1 invitation.

Manage access

[Invite a collaborator](#)

☐ Select all Type

Find a collaborator...

<input type="checkbox"/>	acompsc1 Collaborator	
<input type="checkbox"/>	amanf1 Collaborator	
<input type="checkbox"/>	ColinDunn15 Collaborator	
<input type="checkbox"/>	Reza Sadeghi Awaiting RezaSadeghiWSU's response	Pending Invite
<input type="checkbox"/>	trunds Collaborator	

References

Aaron and Davis. Hacker shack: How to make a website with python and django e01-e07. URL <https://www.youtube.com/channel/UCecNXmr7DYq1XxpWHSxaN0w>.

Django is a high-level python web framework that encourages rapid development and clean, pragmatic design. <https://www.djangoproject.com/>, 12 2020. Accessed on 2020-11-25.

Get started with docker, we help developers and development teams build and ship apps. <https://www.docker.com/>, 12 2020. Accessed on 2020-11-25.

Free bootstrap themes & templates. <https://startbootstrap.com/themes>, 12 2020. Accessed on 2020-11-25.

Postgresql: The world's most advanced open source relational database. <https://www.postgresql.org/>, 12 2020. Accessed on 2020-11-25.

Github is a development platform inspired by the way you work. <https://github.com/>, 12 2020. Accessed on 2020-11-25.