

---

# WEBSITE PROJECT PRESENTATION

BY: STEVEN ATILHO, ANDREW  
MAHR, COLIN DUNN, ANDREW  
MANFREDI, BRANDON TRUNDY





# WEBSITE DESIGN OVERVIEW

- Home Page -
- About Page -
- Contact Page -
- Login/Logout Portal -
- Profile Page -
- Design and Code Implementation

# FINAL PROJECT CSCI-3351-01

## Home



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.

## 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):

# ABOUT PAGE

## CONTACT



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

Andrew Mahr: [amahr1@unh.newhaven.edu](mailto:amahr1@unh.newhaven.edu)

Steven Atilho: [satil1@unh.newhaven.edu](mailto:satil1@unh.newhaven.edu)

Brandon Trundy: [btrun1@unh.newhaven.edu](mailto:btrun1@unh.newhaven.edu)

Andrew Manfredi: [amanf1@unh.newhaven.edu](mailto:amanf1@unh.newhaven.edu)

Colin Dunn: [cdunn5@unh.newhaven.edu](mailto:cdunn5@unh.newhaven.edu)

Name

Email

Message

Send

Reach Us @  
[goonsquadwebsite@gmail.com](mailto:goonsquadwebsite@gmail.com)

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

<mailto:satil1@unh.newhaven.edu>

# CONTACT PAGE

### Login



Username:

Password:

Login

Reach Us @  
[goonsquadwebsite@gmail.com](mailto:goonsquadwebsite@gmail.com)

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

Last Update - December 2020

# LOGIN/LOGOUT PORTAL





**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](mailto:goonsquadwebsite@gmail.com)

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

Last Update - December 2020

# PROFILE PAGE

# DJANGO ADMIN PORTAL

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

Django administration

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

Action:

Go

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





# DESIGN AND CODE IMPLEMENTATION

- Postgres Database – used to store account data such as logins, permissions, user interests and personas, etc...
- Docker – used to run website in a container from anywhere
  - Docker commands helped automate the setting up of commands and taking care of ensuring everything is started and removed when finished
- Django – Used as the shell and the framework for the website.
  - Authentication features and login portals
- Visual Studio Code – used to collaborate and create the files necessary for the website

# PYTHON CODE/SETTINGS

```
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.1/ref/settings/
11 """
12
13 from pathlib import Path
14 import os
15
16 # Build paths inside the project like this: BASE_DIR / 'subdir'.
17 BASE_DIR = Path(__file__).resolve().parent.parent
18 PROJECT_DIR = os.path.join(BASE_DIR, "goonsquad_website")
19
20
21 # Quick-start development settings - unsuitable for production
22 # See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/
23
24 # SECURITY WARNING: keep the secret key used in production secret!
25 SECRET_KEY = "i-v%y82d-(w*k!hsjtbui8r&#m8b7n3gieixs=rtbzu0)v%a%5"
26
27 # SECURITY WARNING: don't run with debug turned on in production!
28 DEBUG = True
29
30 ALLOWED_HOSTS = []
31
32
33 # Application definition
34
35 INSTALLED_APPS = [
36     "django.contrib.admin",
37     "django.contrib.auth",
38     "django.contrib.contenttypes",
39     "django.contrib.sessions",
40     "django.contrib.messages",
41     "django.contrib.staticfiles",
42     "goonsquad_website.apps.accounts",
43     "goonsquad_website.apps.contact",
44 ]
45
46 MIDDLEWARE = [
47     "django.middleware.security.SecurityMiddleware",
48     "django.contrib.sessions.middleware.SessionMiddleware",
49     "django.middleware.common.CommonMiddleware",
50     "django.middleware.csrf.CsrfViewMiddleware",
51     "django.contrib.auth.middleware.AuthenticationMiddleware",
52     "django.contrib.messages.middleware.MessageMiddleware",
53     "django.middleware.clickjacking.XFrameOptionsMiddleware",
54 ]
```

# PYTHON CODE/SETTINGS

```
55
56 ROOT_URLCONF = "goonsquad_website.urls"
57
58 TEMPLATES = [
59     {
60         "BACKEND": "django.template.backends.django.DjangoTemplates",
61         "DIRS": [os.path.join(PROJECT_DIR, "templates")],
62         "APP_DIRS": True,
63         "OPTIONS": {
64             "context_processors": [
65                 "django.template.context_processors.debug",
66                 "django.template.context_processors.request",
67                 "django.contrib.auth.context_processors.auth",
68                 "django.contrib.messages.context_processors.messages",
69             ],
70         },
71     },
72 ]
73
74 WSGI_APPLICATION = "goonsquad_website.wsgi.application"
75
76
77 # Database
78 # https://docs.djangoproject.com/en/3.1/ref/settings/#databases
79
80 POSTGRES_HOST = os.environ.get("POSTGRES_HOST", default="")
81 POSTGRES_DB = os.environ.get("POSTGRES_DB", default="postgres")
82 POSTGRES_USER = os.environ.get("POSTGRES_USER", default="")
83 POSTGRES_PASSWORD = os.environ.get("POSTGRES_PASSWORD", default="")
84
85 DATABASES = {
86     "default": {
87         "ENGINE": "django.db.backends.postgresql",
88         "NAME": POSTGRES_DB,
89         "USER": POSTGRES_USER,
90         "PASSWORD": POSTGRES_PASSWORD,
91         "HOST": POSTGRES_HOST,
92         "PORT": 5432,
93     }
94 }
95
96
97 # Password validation
98 # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators
99
100 AUTH_PASSWORD_VALIDATORS = [
101     {
102         "NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator"
```

# PYTHON CODE/SETTINGS

```
97 # Password validation
98 # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators
99
100 AUTH_PASSWORD_VALIDATORS = [
101     {
102         "NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator",
103     },
104     {
105         "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
106     },
107     {
108         "NAME": "django.contrib.auth.password_validation.CommonPasswordValidator",
109     },
110     {
111         "NAME": "django.contrib.auth.password_validation.NumericPasswordValidator",
112     },
113 ]
114
115
116 # Internationalization
117 # https://docs.djangoproject.com/en/3.1/topics/i18n/
118
119 LANGUAGE_CODE = "en-us"
120
121 TIME_ZONE = "UTC"
122
123 USE_I18N = True
124
125 USE_L10N = True
126
127 USE_TZ = True
128
129
130 # Static files (CSS, JavaScript, Images)
131 # https://docs.djangoproject.com/en/3.1/howto/static-files/
132
133 STATIC_URL = "/static/"
134 STATICFILES_DIRS = [os.path.join(BASE_DIR, "static")]
135
136 # Django Auth Settings
137 LOGIN_URL = "accounts:login"
138 LOGIN_REDIRECT_URL = "accounts:profile"
139 LOGOUT_REDIRECT_URL = "public:index"
140
141 EMAIL_BACKEND = "django.core.mail.backends.console.EmailBackend"
142 DEFAULT_FROM_EMAIL = "goonsquadwebsite@gmail.com"
143
```

# PYTHON CODE/VIEWS

```
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")
```

# PYTHON CODE/ADMIN

```
from django.contrib import admin

from .models import UserProfile, UserPersona, UserInterest

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

# PYTHON CODE/URLS

```
"""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
]

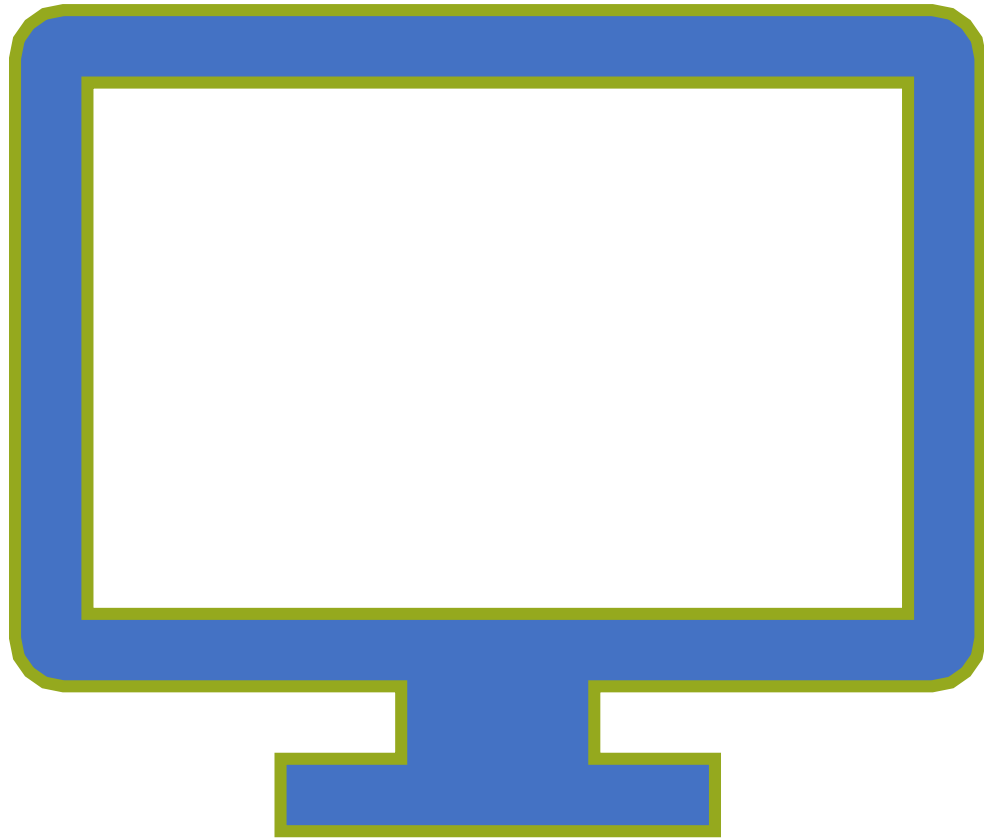
# Differnt Django Views
# accounts/login/ [name='login']
# accounts/logout/ [name='logout']
# accounts/password_change/ [name='password_change']
# accounts/password_change/done [name='password_change_done']
# accounts/password_reset [name='password_reset']
# accounts/password_reset/done [name='password_reset_done']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

# DOCKER

```
1 FROM python:3.9.0-slim as production
2
3
4 ENV PYTHONUNBUFFERED=1
5 WORKDIR /app/
6
7 RUN apt-get update && \
8     apt-get install -y \
9     bash \
10    build-essential \
11    gcc \
12    libffi-dev \
13    musl-dev \
14    openssl \
15    postgresql \
16    libpq-dev
17
18 COPY requirements/prod.txt ./requirements/prod.txt
19 RUN pip3 install -r ./requirements/prod.txt
20
21 COPY manage.py ./manage.py
22
23 COPY setup.config ./setup.config
24 COPY goonsquad_website ./goonsquad_website
25
26 EXPOSE 8000
27
28 FROM production as development
29
30 COPY requirements/dev.txt ./requirements/dev.txt
31 RUN pip3 install -r ./requirements/dev.txt
32 COPY . .
```

# DOCKER-COMPOSE

```
1 version: "3.7"
2
3 x-service-volumes: &service-volumes
4 - ./app:/rw,cached
5
6 x-database-variables: &database-variables
7 POSTGRES_DB: postgres
8 POSTGRES_USER: postgres
9 POSTGRES_PASSWORD: postgres
10
11 x-app-variables: &app-variables
12 <<: *database-variables
13 POSTGRES_HOST: postgres
14
15 services:
16   website:
17     image: goonsquad_website:latest
18     command: python3 manage.py runserver 0.0.0.0:8000
19     volumes: *service-volumes
20     environment: *app-variables
21     depends_on:
22       - db_migrate
23     ports:
24       - "8000:8000"
25
26   db_migrate:
27     image: goonsquad_website:latest
28     command: python3 manage.py migrate
29     volumes: *service-volumes
30     environment: *app-variables
31     depends_on:
32       - postgres
33
34   postgres:
35     image: postgres
36     ports:
37       - "5432:5432"
38     environment: *database-variables
39     volumes:
40       - db-data:/var/lib/postgresql/data
41
42 volumes:
43   db-data:
```



---

DEMO





Any Questions?