Web Dev Technologies Project 2017-2018

Νέμανια Νέντις
A.M. 1115201400124
Θανάσης Φιλιππίδης
A.M. 1115201400215

Contents

About

Professional Networking Site or PNSite.

A social networking website targeting career and business professionals of every field, its purpose is to enable them to network professionally.

Designed as a university project for the Technologies of Web Applications class at the National Kapodistrian University of Athens.

Functions supported

• Https (SSL/TLS) (self-signed certificate)

• Login/Sign up

• Admin/Administration page gives an overview about the users, and their personal details can be exported as XML.

• Navbar with Home page, Network, Ads (Job Advertisments), Messages, Notifications, Profile (Personal details), Settings, Search field, Logout.

• Home page tab shows the user's details on the left, above a panel with a quick preview of the user's network. There is a field on the right side where the user can create a new post and a timeline with the user's posts, the user's friends' posts, as well as the user's friends' friends posts that the user's friends have liked.
Clustering Collaborative Filtering is used to sort the order in which posts appear on one's timeline. Each post also has an amount of likes (interests) and comments.

• Network tab shows a panel with the user's friends, their details, their profile photos, and links to their profiles.

• Ads tab is divided in half. On the left side, there is a panel/list of notifications about other users that have applied to one or more of the user's job advertisements; if there are any.
On the right side, there is a field where the user can create a new ad and include some "hashtags" (skills) that are related to that advertisement. There is also a timeline displaying job advertisements from other users, ordered using a combination of Content Based and Clustering Collaborative Filtering. The user can apply on those advertisments.

• Messages tab is divided into two sections, the left one is a list of the people the user has had conversations with in the past. After clicking on one of those conversations, the whole conversation with that user appears on the right side, and there is a field where new messages can be written and sent to that person.

• Notifications tab is divided in two sections. The upper one shows the connection (friends) requests that this user has received from other users. The user can either accept or decline them. The lower section shows the notifications of other people liking or commenting on one of the user's posts.

• Profile tab provides a list of fields in which the user can add his personal details or update them. The user can also add/change his/her profile picture, and decide which details would be visible publicly, or only to the user's

friends.

• Settings tabs provides fields in order to change the user's email and password.

• Search field is a field in which a user can type a text string and any name/surname matching that string will appear in the search results. Once the user clicks on one of the results, he/she is transferred to the username's profile. Also, there is the option of instantly messaging them.

• Logout tab does, ehm the obvious.

Technologies used

• The python Django framework has been used to implement those functions, as well as sqlite for the back-end of the app. Note that for a production grade website sqlite is one of the worst options, as it for example will not enforce the model restrictions when creating one. We suggest PostgreSQL or MySQL. Anyhow we decided to include these just to make things clearer and to make migration to mysql easier if we choose to.

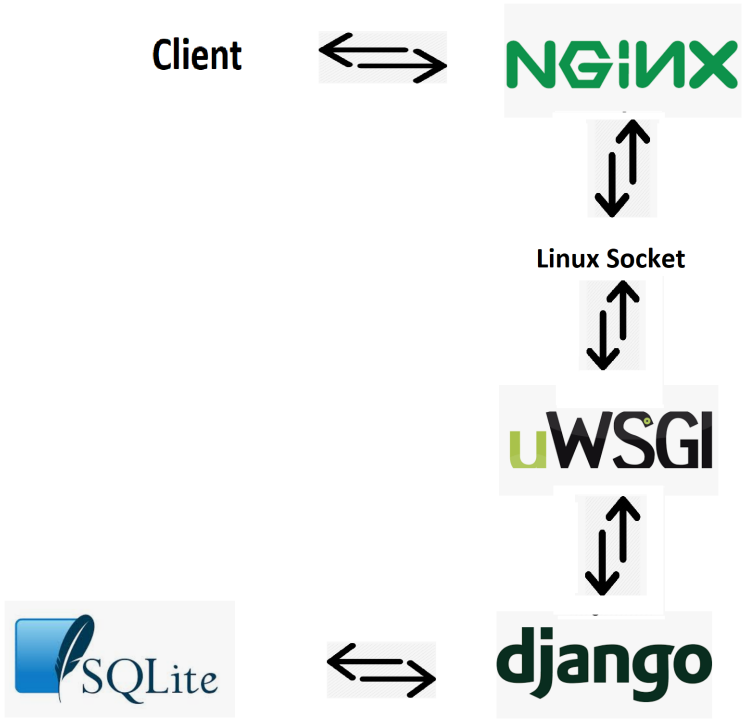
• For the front-end HTML, CSS, Bootstrap and Javascript was used.

• The Django framework provides a development server but it is very limited. So we choose Nginx as a server. It is connected to Django through uWSGI. Django provides a wsgi module and it makes it very easy to connect the stack.

• In order to avoid reloading out pages and to make the site more user friendly jquery was used along with ajax. Almost all of the related code can be found in PNapp/static/PNapp/js/rango-juqery.js file

• A python virtual environment to isolate the components.

The Stack



Setup

Keep in mind this was developed on ubuntu 16.04 and python3.

1. Setup existing Django Projexitct Tutorial (click me)

   If you wish to run the development server on a different (ex.8001) port just run: python ./manage.py runserver 8001.

   You can now visit the site at localhost:8000. And the admin page at localhost:8000/admin.

   Whenever you edit your model fields (adding a new one, changing an existing one or altering any of the arguments it takes) then you should always run migrations:

   python manage.py makemigrations

   python manage.py migrate

2. Setup Nginx HTTP Server Self-Signed SSL/TLS Certificates on Ubuntu 16.04 LTS Servers (clickme)

3. Setup Nginx HTTP Server Self-Signed SSL/TLS Certificates on Ubuntu 16.04 LTS Servers (clickme)