Coursework 2 Report

Rafal Ozog
40217610@napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

Keywords – api, web-development, python, jinja2, html, html5, css, css3, flask, levinux, startUpCircle

1 'StartUp Circle'

1.1 What was the aim?

This coursework is the second and last project in this module. The main goal of this work was to sum up knowledge we should gain and to express it in a form of working, interactive web application. A number of different AI technologies has been covered to give us a proper skillset for this task. We have learnt and practised Python, jinja2, HTML, CSS, Linux using. We also have understood a lot of theoretical, abstract issues, needed in web-development - like API's and URL hierarchies. In this report I would like to summarize all this knowledge I have gained and present my own project, which is actually its output.

1.2 What is the 'Circle'?

The idea of my web application was to create something what could in make someone's life easier in any way. I have realized that there are many people trying to create an appropriate team and collect enough money to start their own business. Very often troubles on this stage make them discouraged and finally stop their plans. I wanted to change that. I started to think about web-page, which would allow people to exchange their business ideas, form start-up teams and gain appropriate funds. That is what 'StartUp Circle' is created for.
Each 'circle' is a one business idea with its leader, explanation and investment (needed) value. User can find circles which seem to be interesting for him (and join them or inform that is interested) or create his own circles. Everyone can become circle team member if the leader will accept him. The member's input into the circle can be both his skillset (work) or money (investment).
In next sections I would like to explain a functionality and its implementation in details.

2 Front-end Design - user's view

2.1 Navigation Menu

It is very important to allow user have an access to each main path from every single location within our web-page. To meet this requirement I have decided to implement a comfortable, horizontal menu on a top of each page. The navigation menu allow user to login (or register if he doesn't have an account), go to circle creation page, go to search engine page or come back to main (index) page (after he clicks on logo). If the user has already logged in, the 'log in' option is replaced by 'log out'.

2.2 User Menu

On left-hand side has been located the second part of user interface - the user menu. This section contains information about logged in user (first name, surname, e-mail, town and country), his personal avatar and two useful buttons, which allow the user to visit his account page or to log out. If the user is not logged in - then the buttons redirect him to the 'login' or 'register' page. In this case user data is replaced by simple name 'Anonymous user'. This menu is also displayed and available from each single location within the url structure.

2.3 Main page

The very first thing seen by user is the main page. The content displayed on this page consists of 9 newest circles, which are displayed in shortened form. The user can access each circle details by simple 'See more' button click (however, he has to be logged in to see the details - if he is not, then the page will redirect him to the 'login' page).

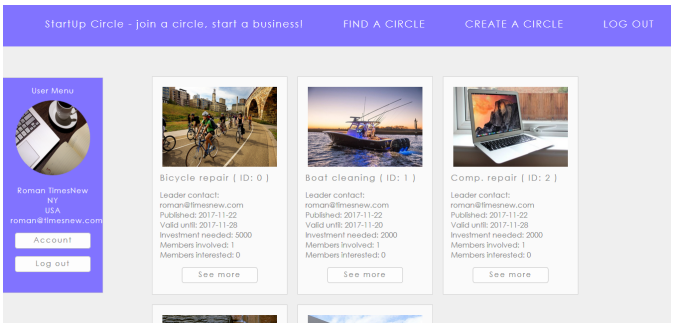

Figure 1: **Main page**

2.4 Find a circle - internal search engine

The application contains an internat mechanism, which allows user to filter only those circles, which seem to be the most interesting for him. The seach engine takes a phrase, which has to be found within title or description,

minimal and maximal investment value and returns (displays) only circles, which meet all these conditions.
The search engine is available both for logged in and anonymous users, however (just like with the main page) - only the first group is allowed to see the most important parts, like description and contact details.

2.5 Login page - do you have an account?

Login page allows user to undertake two actions - to log in using his already created account or to create a new one. This page is displayed every time, when someone tries to access sections restricted for anonymous users.

2.6 Let's create an account!

To create a new account, user has to provide his new credentials (e-mail, which will be his new login and password). He can also optionally provide his personal data, including full name, address, town, country - and contact details, which will make him more accesible and therefore - more valuable for potential circle team. What is more, it is possible to add a short personal/professional description (e.g. skillset, education, experience) and to add a personal photo or avatar.
After the 'Register' button click, user is automatically logged in and redirected to the main page.

2.7 Create a new circle - make your business idea alive!

We have created our own account - so now we can finally access full potential of the 'StartUp Circle'! In this section (restricted for anonymous users), we can create our own circle. Friendly-looking interface will assist us. We need to provide some basic information - title, description, estimated investment value and an advertisement expiry date. We also can add some photo or illustration optionally.

2.8 Account page - manage your profile

On the account page user can check all his personal and contact details, his profile photo and personal description. This section is accessible (only for logged users) through the 'account' button, located on the left-hand side 'user menu'.

2.9 Log out and keep the account safe

When user decides to finish his visit, he can easily log out of his account by simple click onto one of two available 'log out' buttons. One of them is located within the navigation menu (upper right corner) and the second within the user menu (lower left corner). After the logout the page will redirect him to the main page.

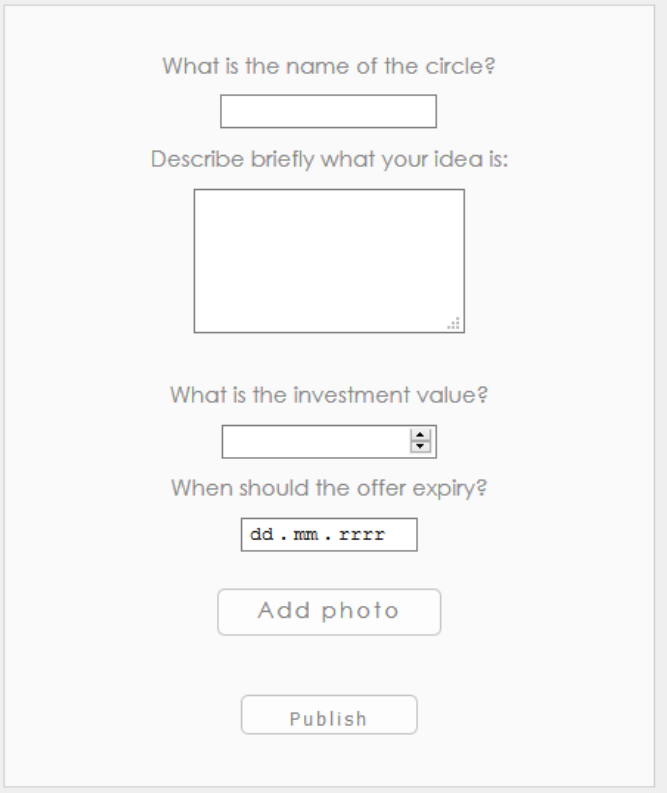

Figure 2: **Main page**

3 Back-end Design - what is hidden?

3.1 Database - heart of the project

The most internal layer of the application, which from I started the whole work is data organization. I have decided to create internal SQL database, implemented by python library sqlite3.
The data is split into two separated files - 'users' and 'data'. The first one contains information about accounts and the second - about created circles. This solution allows administrator to export or copy these two data parts separately. Both parts are related to appropriate 'schema' files.
Databases can be accessed and operated through special interface (which consists of Python functions) located in files 'data.py' and 'users.py'.

3.2 Python code organization

Whole code has been organized into few functional parts - 'data.py', 'users.py', 'find.py' (internal search engine) and 'index.py'. This division makes the code readable and open for modification. What is also important, different functionality sections can be exported separately to different projects in the future.

3.3 URL hierarchy - API

The application is based on semantically-organized URL hierarchy. Paths related to similar funtions have been grouped together, to provide readable and simple API.

This division corresponds to the Python code division (mentioned in a previous paragraph). Paths related to user (account) functionality are grouped within common 'users' path and those related to circles - into 'circles' path.

3.4 Login mechanism

As the page needs users to be fully-functional I have implemented appropriate mechanisms, which allow user to create an account, log in and access sections restricted for anonymous visitors.
Registration is based on database interface (hidden within 'users.py'), which allows developer to create and enter a new record into the 'users' base.
Login function is located within the main program file. It takes credentials provided by html template ('login.html') and compares them (login, which is actually a user e-mail - and password) with those stored within database. If they are correct - then the user 'logged in' status and his login will be stored within **a browser session**.
An important piece of this mechanism is a function, checking if the user is logged in (if a particular path requires it). The ('check if logged') function **decorates** an other one (related to restricted path), checks the user status (within the browser session) and accordingly to its value return 'true' or redirects user to the 'login' page.

3.5 Password encryption

To provide the highest security standards user passwords are encrypted. This process is split into two activities. When the user creates an account, his password is firstly **salted** and then **hashed**. Only the hash is stored within database (never clear password). Both processes are provided by Python **'bcrypt'** library.
When the user wants to log in, the 'login' function compares a hash of provided credential with the stored hash.

3.6 Internal search engine

The website allow user to filter displayed circles. A searching function is stored within separate python file ('find.py'). When the appropriate html template passes searching credentials (searched phrase, minimal and maximal investment values) it connects to the database and send an SQL query. Returned table is then packed (row by row) to a list and passed forward (to calling function).

3.7 My own layout - CSS

I consider website layout as important factor of a user experience. If a page doesn't look pretty, people will not want to visit it and get to know what is inside. Therefore, in my personal opinion, sometimes layout may be more important than back-end logic.
I decided to create my own layout using my personal CSS and HTML knowledge. I tried to do my best to make it readable, clean and friendly-looking. I took into account factors like proper font, letter-spacing, different div's proportions, button layout, color palette and many other. During this work I was following the rule of structure and layout separation. The whole CSS code is located within single file and can be replaced by other one.
Has the goal been reached? I will leave that question to be answered by users.

3.8 Browser session using

I have decided to use browers session in this project. That structure is responsible for user status and login storage, which in turn is used by different sub-pages to find out if the user is logged (and therefore which interface version should be displayed and which pages restricted) and who the user actually is.

3.9 URL argument passing

The website uses url addresses to pass arguments back to the code. This solution has been used for example to open each circle's individual page. There is only one HTML template for this purpose, however depending on argument passed within URL (a circle ID), displayed content will be specified.

4 Enhancements - what could be improved?

Nothing is perfect. Even the best piece of art could be improved in some way. Web-development seems to fit perfectly to this statement. I was trying to prepare the best possible web application I could, however there is still a number of features I would like to implement or improve, having a little bit more time.

4.1 Internal communication

In this moment user can log in, read a circle details and decide to contact a leader using e-mail or phone. I would like to implement internal way of communication, which would be self-sufficient. There are already prepared buttons, which could be clicked by user to indicate his interesting within the project or to send an alert to a leader informing that someone has just requested team membership. It doesn't work yet.
I would like to implement also some internal message exchange mechanism, which would allow users to contact without using external communication ways.

4.2 CSS refactorization

If I would have more time, I would definitely like to proceed a refactorization of CSS code. During last days I was working in a hurry, so as a result a little mess has appeared there. Some pieces of code could be integrated into common sections. Especially style of divs indicated by different IDs could be easily grouped into few classes (some of them use classes, but not all), what would make the code more readable.

5 Critical Evaluation - how do I see my own project?

5.1 What am I satisfied of?

Generally I am happy with a final result. In my personal feeling website's functionality is working well and the layout does not look badly. I can say that in my opinion the project is built on well-organized database and its proper Python interface and the code has been split into functional and readable sections, what makes it possible to be changed by another developer. What is more, planned and semantically-organized URL hierarchy forms transparent API, which seems to be simple in using and changing.
I am also satisfied of (a little bit) more advanced features used within the project - password encryption, login mechanism, browser session using.
The last thing I would like to mention about in this section is a CSS layout, which has been designed, planned and implemented entirely by me.

5.2 What could be better?

The first thing I am not really satisfied of is that I have not implemented the whole functionality that I intended to. As I already mentioned in section 4.1 I wanted to allow users indicate if they are interested within particular circle and to send circle leaders alerts about membership requests (which in turn could be accepted or rejected by them).
Another thing that I would like to fix is generally a number of bugs, which of some are hidden and some probably undiscovered. Basically, I have to say that I should provide a good testing strategy.

6 Personal Evaluation

I think that the most challenging part for me was to connect sql database to the website. I always had a problem with joining this to different projects. I am glad that I could practice that during this project.
This project allowed me to express gained knowledge. It was also a good opportunity to train and improve my skills.
What is important I can say I had a good fun during this work and definitely satisfaction at the end.