Honour Project
CSI4900 Technical Report
Winter 2018

Author:
Junhan Liu 7228243

Supervisor: Dr. Abdulmotaleb El Saddik
Co-supervisor: Fedwa Lamarti

Github repository:
[https://github.com/jimjimliu/Bicy-Eco-System.git]

Demo:
https://youtu.be/qwFPymhrlKU

## Introduction

> ## Project Goal

The project is to develop a web based application with interface that helps users to monitor and track their physical exercise progress (i.e. exercise like riding a bike or running) over time. Users receive motivation prompts(elaboration follows) while they perform their daily exercises.

The application stores users' every-time exercises information in the database; every time after users login in, the application displays users' profile including users' personal health information and their previous physical activities' information. The application is able to generate various charts for users to gain a better and intuitive understanding of users' activity progress: whether they are making any progress during a chosen period of time, their total exercise time, overall energy burned, and etc.

In terms of motivation prompts: the application gives users' a opportunity to donate the energy (When ones bike or run in a gym using the equipment, equipments would generate energy. We assume this energy generated by users could be recycled.) generated by their activities. Users could select, in the interface, a country to donate their green energy. Of course, all the data would be collected and then displayed for users in the interface.

This is the overall goal of the project and some main functions of it. More detailed functionality description will be illuminated in functionality section.

> ## How to run

> - ### To run the application:

Strongly suggest running the system in chrome browser, not IE.

1, install WAMP server for php+postgresql extension configuration. Download WAMP server of 64 bits, you can find it here: http://www.wampserver.com/en/download-wampserver-64bits/ follow the wizard to install it.

2, After installation check on the quick launch bar(right bottom corner in desktop) to see if WAMP server is running. If it is running, the icon changes colour to green.

3, Click on WAMP icon from the quick launch bar to open context menu, select `PHP -> PHP extensions -> php_pgsql.`

4, Put the project root folder named "project" under c:\WAMP\www\ folder.

5, Download postgresql from: https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

6, Follow the guide to set up password and etc. Download early versions before version 10. PostgreSQL 10 might not prompt you to set up users and password. Early versions are more stable.

7, To modify database, go to postgresql folder, run the postrgesql shell to access the database, normally located in

```
C:\program files\postgresql\9.6 (your version is)\scripts
```

OR

Use pgadmin UI to munipulate database, located in

```
C:\program files\postgresql\9.6\data\pgadmin.exe
```

8, Download python 3.6 or later version, and install. Add python 3.6 to variable path.

9, Using 'database.txt' to populate the database first. you can also copy the script code into postgresql shell to populate.

10, Open browser and type http://localhost/project/HomePage.html to start.

### *> - ### Before you test the system*

You need to change the configurations in just one .php file.
Go to connection.php, you will see something like this:

```
<?php
    //connect the databse
    $dbname ="xxxxx";  //dabase name
    $user_name = "postgres";
    $pass="xxxxx"; //password
    $host="localhost";
    $port="5432"; //database port
```

Change the configuration to your own database information.

Also, in folder `python simulation`, you will see this code:

```
#return a connection object
def dbConnection():
    hostname = 'localhost'
    database = 'xxxx'
    port = '5432'
    username = 'postgres'
    password = 'xxxx'
```

Change configuration to you own database information.

Development Details
### *> ## Type of application*

Web based application, interacts with postgreSQL database.

> ## Target users

The application is aimed to help individuals that would like to perform daily physical activities to better monitor their progress; at the same time, those who want to keep track of their activities' data over time.

> ## Functionality description

- `User sign up:`

Signing up an account. Asking for users' personal health information(i.e. height, weight, age, gender, name, email address, password).

- `User preferences:`

When users have signed up for the first time, they will be asked to select a country to donate their green energy. Users could change their choice of country every a period of time.

- `User login:`

Allows users to login and allows users to recover their password.

- `personal account:`

Providing a panel which users could choose which category of data they would like to view. Users could view their history physical activities related data including their total exercise time, overall energy generated and donated, how many calories they have burned, their bmi index, and etc. Providing interface that displays whether the user is making any progress over the selected period of time by some charts. Users could select to view the overall energy donation information: a pie/bar chart displays the overall energy donated for each country.

- `Display screens:`

The user can view his profile in a screen, but cannot change information other than email address, and password. When the user is pedaling, we display the information related to his current exercise activity so he can view it as he bikes, such as the date and time of exercising, the duration and the average speed of biking. We then store this information in the database.

- `Messages displayed:`

During the exercising period, messages are displayed for user encouragement, such as: You burned "number" of calories! Great job!". A sound can be played for encouragement). We also display motivation messages, such as: If you exercise 5 more minutes you will generate "number" of watts. At the end, we display a message like: Thank you, you have donated "number" of watts today.

- `log out:`

If the system has been idle for more than 40 minutes, the system automatically log out the user.

- `Email recover:`

By clicking recover email, users enter their registered email. An email recover message will be sent to users registered email address containing users requested personal information.

## *> ## Technology*

### *>- ### Language*

`HTML5` `CSS` `javascript` `SQL` `PHP` `jQuery`

### *> - ### Tools*

 `postgresql 10`, `pgAdmin 4` `WAMP server`

### *> - ### Framework*

`Chartist.js` `fontAwsome` `jQuery`

Design

User Interface Design

The user interface design is based on evaluation of user and tasks analysis. The website interface including several parts as following:

- Homepage
- Log in page
- Sign up page
- Personal sign up information page
- Biking Activity page
- Personal account page

When designing the user interface, I decided to design with the simplicity, user experience, efficiency, and aesthetics of the site as the core through analysis of users. And through heuristic evaluation to modify our interface design to enhance the user experience.

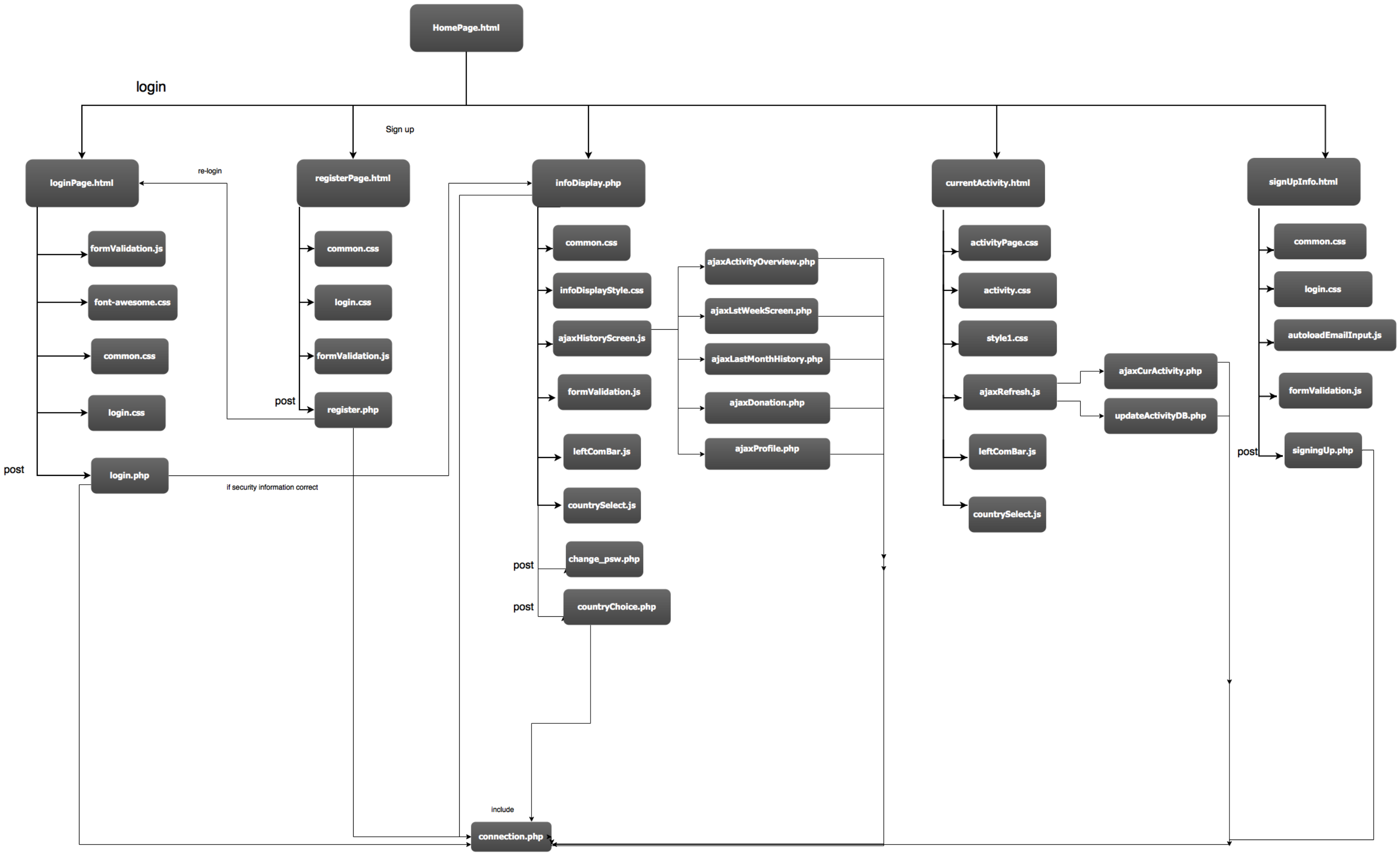
For login, sign up and home page, users use them to get into this web application . So I try to make them succinct to avoid poor user experiences. The appearance, layout, and functionality buttons are basically identical. The interface appearance follows the principles that interface should be simple but robust, the interface must convey the theme of the application, and the interface should be designed with good affordance.

For Personal sign up information page, it shows the user's information and account details. The goal is to provide users a good mental design that users could do more things and view more things using less operations. To avoid displaying too much informations in a single screen. I added a left sidebar to allow users to select different category in order to display different information.

For activity page, the main functionality of it is to provide users a screen that displays real time average speed, cumulative calories burned, and cumulative watts generated. So I focus on the visibility of system activity. I designed the font size into consideration here. The font size of particular information should be made larger than other elements to let users view their real time information better. At the beginning, the stop key in the activity page is not darkened after the end of the activity, although it will not update the database again, but it will make the user think that you can continue to press. So I made the stop key be darkened after user press “stop and save” button.

In the part of error handling, the user interface can prompt error notification messages to inform users the error state and to guide user to their goals. The notification is easy to understand by users.

System Flowchart



## General

- Complete System Flow Chart can be found in folder `flowchat`
- Source code documentation can be found in foler `documentation`