SINGAPORE UNIVERSITY OF SOCIAL SCIENCES

ICT239

End-of-Course Assessment – January Semester 2021

Web Application Development

INSTRUCTIONS TO STUDENTS:

1. This End-of-Course Assessment paper contains **THREE (3)** questions and comprises **TEN (10)** pages (including the cover page).

2. You are to include the following particulars in your submission: Course Code, Title of the ECA, SUSS PI No., Your Name, and Submission Date.

3. Late submission will be subjected to the marks deduction scheme. Please refer to the Student Handbook for details.

IMPORTANT NOTE

ECA Submission Deadline: Sunday, 23 May 2021, 12 noon

Please Read This Information before You Start Working on your ECA

This ECA carries 70% of the course marks and is a compulsory component. It is to be done individually and not collaboratively with other students.

Submission

You are to submit the ECA assignment in exactly the same manner as your tutor-marked assignments (TMA), i.e. using Canvas. Submission in any other manner like hardcopy or any other means will not be accepted.

Electronic transmission is not immediate. It is possible that the network traffic may be particularly heavy on the cut-off date and connections to the system cannot be guaranteed. Hence, you are advised to submit your assignment the day before the cut-off date in order to make sure that the submission is accepted and in good time.

Once you have submitted your ECA assignment, the status is displayed on the computer screen. You will only receive a successful assignment submission message if you had applied for the e-mail notification option.

ECA Marks Deduction Scheme

Please note the following:

*a) Submission Cut-off Time – Unless otherwise advised, the cut-off time for ECA submission will be at **12:00 noon** on the day of the deadline. All submission timings will be based on the time recorded by Canvas.*

*b) Start Time for Deduction – Students are given a grace period of 12hours. Hence calculation of late submissions of ECAs will begin at **00:00 hrs** the following day (this applies even if it is a holiday or weekend) after the deadline.*

*c) How the Scheme Works – From 00:00 hrs the following day after the deadline, **10 marks** will be deducted for each **24-hour block**. Submissions that are subject to more than 50 marks deduction will be assigned **zero mark**. For examples on how the scheme works, please refer to Section 5.2 Para 1.7.3 of the Student Handbook.*

Any extra files, missing appendices or corrections received after the cut-off date will also not be considered in the grading of your ECA assignment.

Plagiarism and Collusion

Plagiarism and collusion are forms of cheating and are not acceptable in any form of a student's work, including this ECA assignment. You can avoid plagiarism by giving appropriate references when you use some other people's ideas, words or pictures (including diagrams). Refer to the American Psychological Association (APA) Manual if you need reminding about quoting and referencing. You can avoid collusion by ensuring that your submission is based on your own individual effort.

The electronic submission of your ECA assignment will be screened through a plagiarism detecting software. For more information about plagiarism and cheating, you should refer to the Student Handbook. SUSS takes a tough stance against plagiarism and collusion. Serious cases will normally result in the student being referred to SUSS's Student Disciplinary Group. For other cases, significant marking penalties or expulsion from the course will be imposed.

Answer all questions. (Total 100 marks)

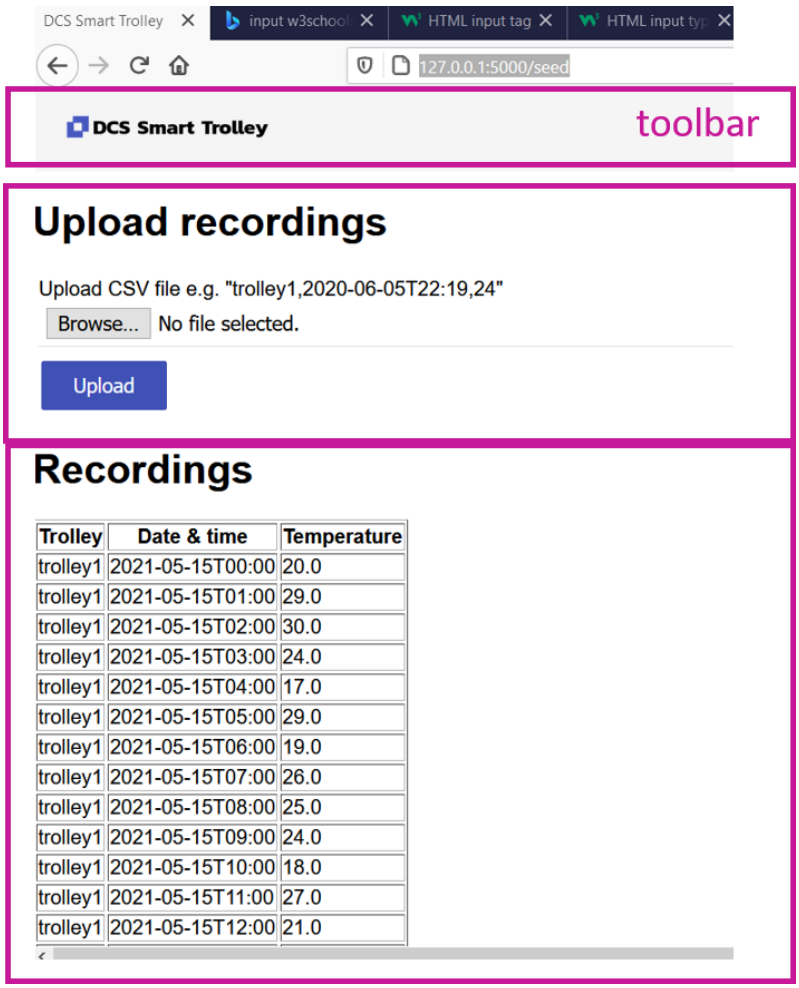
This Resit ECA is based on the TMA for July 2020 semester, which you have passed. You may reuse the source code of your TMA as a base to revise and extend further if it is applicable. Similarly, source code from the review materials posted on the Resit group on Canvas is made available for your use for this ECA. More specifically, the zipped source code file can be retrieved from the following URL under the Modules subfolder.

https://canvas.suss.edu.sg/courses/34268/files/5916730?module_item_id=429050

In TMA Q3, you are asked to develop a function to upload data in the CSV format. One input data file, dated 2021-05-15, consisting of 3 fields, Trolley, Data & Time, and Temperature, can be retrieved from the following link:

https://github.com/paulhjwu/ICT239-Resit/blob/master/seed_demo_ECA.csv

Through an URL http://127.0.0.1:5000/seed, the data can be uploaded, via the section "Upload recordings", and stored in the backend database that can be displayed in the section "Recordings" as shown in the following Diagram Q1(b).



Trolley	Date & time	Temperature
trolley1	2021-05-15T00:00	20.0
trolley1	2021-05-15T01:00	29.0
trolley1	2021-05-15T02:00	30.0
trolley1	2021-05-15T03:00	24.0
trolley1	2021-05-15T04:00	17.0
trolley1	2021-05-15T05:00	29.0
trolley1	2021-05-15T06:00	19.0
trolley1	2021-05-15T07:00	26.0
trolley1	2021-05-15T08:00	25.0
trolley1	2021-05-15T09:00	24.0
trolley1	2021-05-15T10:00	18.0
trolley1	2021-05-15T11:00	27.0
trolley1	2021-05-15T12:00	21.0

Diagram Q1(a): "Upload recordings" and "Recordings" Sections of the view of http://127.0.0.1/seed

Question 1

This question concerning the implementation of Diagram Q1(a), which is to be done according to the MVC design principles.

(a) Regarding the "Upload recordings" section, read the following flask source code and partial HTML code that are used to process the request for http://127.0.0.1:5000/seed, and explain what this code means by filling in the blank below.

```
@app.route("/seed", methods=['GET','POST'])
def seed():
    if request.method == 'GET':
        recordings = repo.get_recordings()
        return render_template("seed.html", recordings=recordings)

    elif request.method == 'POST':
        print(request.form.get('type'))
        file = request.files.get('file')
        repo.upload_recording(file)
        return redirect(url_for('seed'))
```

Backend Flask Controller Code

```
<h1>Upload recordings</h1>
<form action="/seed" method="post" enctype="multipart/form-data">
    <div>
        <label for='upload'>Upload CSV file e.g. "trolley1,2020-06-05T22:19,24"</label>
        <input id='upload' name='file' type='file' accept='.csv' required>
    </div>
    <div>
        <input type="submit" name="type" value="Upload"/>
    </div>
</form>
```

Frotend HTML Code in seed.html – Section "Upload recordings"

Read and comprehend the above two source codes and fill in the blank to explain the functioning of the source codes.

1. When http://127.0.0.1/seed is accessed via the http GET request a webpage seed.html will be displayed
2. Before that, a list of the current recordings will be retrieved by a method repo.get_recordings() and stored in the variable recordings.
3. The recordings are passed into the template system through render_template() method to be displayed together with the webpage in the Recordings section.
4. When the Browse button is pressed in Diagram Q1(b), users can select a file of the CSV type to be uploaded.
5. When the Upload button is pressed, a http POST request, the variable file is assigned with the parameter of the request named file.
6. The method repo.upload_recording() is used to upload the content of the file and update the backend database.

7. After the upload action is completed, the view will display seed.html webpage again due to the redirect(url_for('seed')) statement.
8. However, this time, the content displayed in the Recordings section will include the recordings that were uploaded.

(30 marks)

(b) Regarding the "Recordings" section, a part of the code has been completed, please fill in the remaining of the codes that will allow the display of the content of the recordings. Bear in mind, the render_template method is called at the first place with the content of the recordings passed in to the template system.

```
<h1>Recordings</h1>
<table border=1>
    <tr>
        <th>Trolley</th>
        <th>Date & time</th>
        <th>Temperature</th>
    </tr>

    // Fill in the remaining codes here

</table>
```

Frotend HTML Code in seed.html – Section "Recordings"

(10 marks)

```
    {% for recording in recordings %}
    <tr>
        <td>{{ recording.trolley }}</td>
        <td>{{ recording.datetime }}</td>
        <td>{{ recording.temperature }}</td>
    </tr>
    {% endfor %}
```

(c) Given

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>DCS Smart Trolley</title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" type="text/css" media="screen" href="{{ url_for('static', filename=
</head>
<body>
    <div class="toolbar">
        <span class="logo"><img id="logo-img" src="{{ url_for('static', filename='logo.png') }}
        <span class="logout"><a href="#">Logout</a></span>
    </div>
    {% block body %}{% endblock %}
</body>
</html>
```

```
├── seed_demo_ECA.csv
├── templates/
│   ├── base.html
│   └── seed.html
├── static/
│   ├── logo.png
│   ├── user.png
│   └── index.css
├── requirements.txt
└── venv
```

Namely, please include **Backend Flask Controller Code** in app.py, and **Frotend HTML Code** in seed.html.

Now develop the repo.py module that include the two methods that were used in app.py to handel the /seed route:

- upload_recording
- get_recordings

Question 2

In TMA Q1, a Webpage is required to be developed as shown in Diagram Q1(a) below.

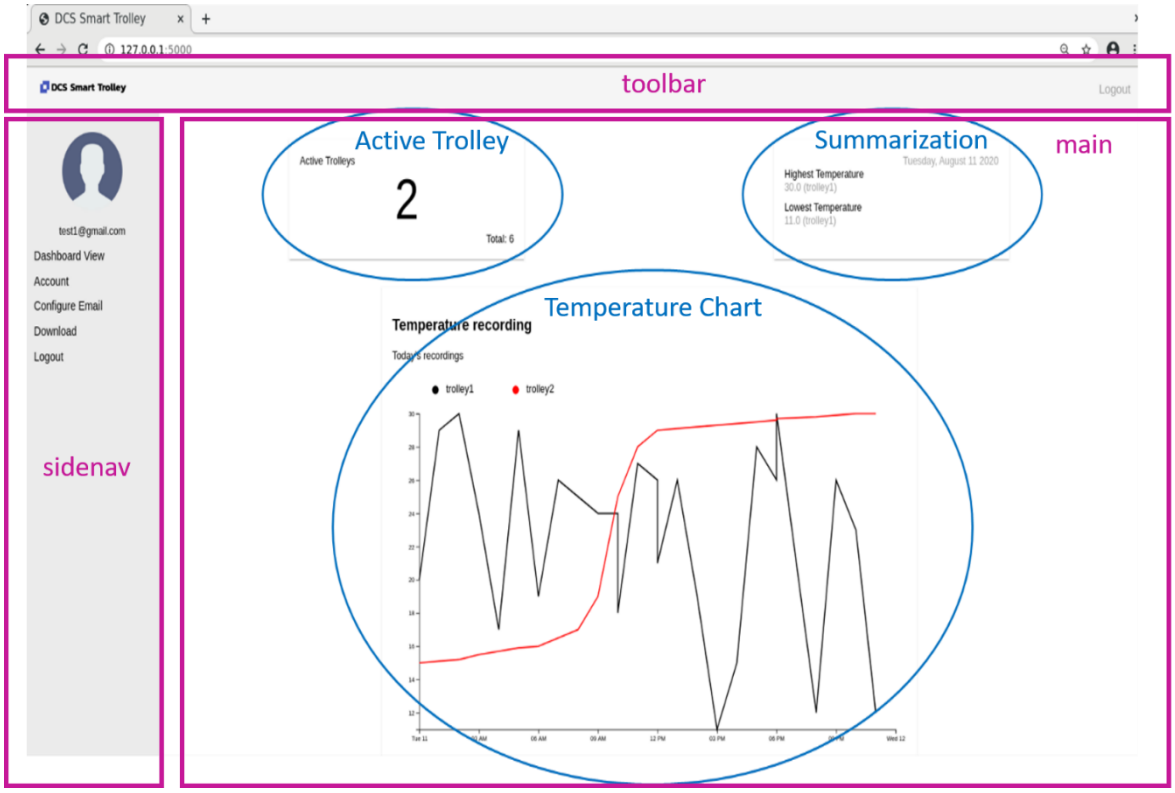


Diagram Q1(a)

The webpage is consisted of 3 main sections:

1. The toolbar section that consists of the company logo and the Logout link
2. The side navigation, or sidenav, section that consists of icon of users, user id, and links to the Dashboard, Account, Configure Email, Download views, and Logout interface and functions
3. The main section that contains the following three views:
    a. Active Trolleys view: show the number of the current active trolleys and total trolleys for a specified date;
    b. Summarization view: show the trolley with the highest and lowest temperatures for a specified date;
    c. Temperature Chart view: show a chart of temperature recording for the current date (say, 'Today'). The x-axis refers to the hours along a day; y-axis refers to the temperature value; and legend refers to trolley id.

Note that upload_recording will take the input file and

Let's exam Q1(a) in further details.

You are asked to develop two new webpages as follows as shown below:

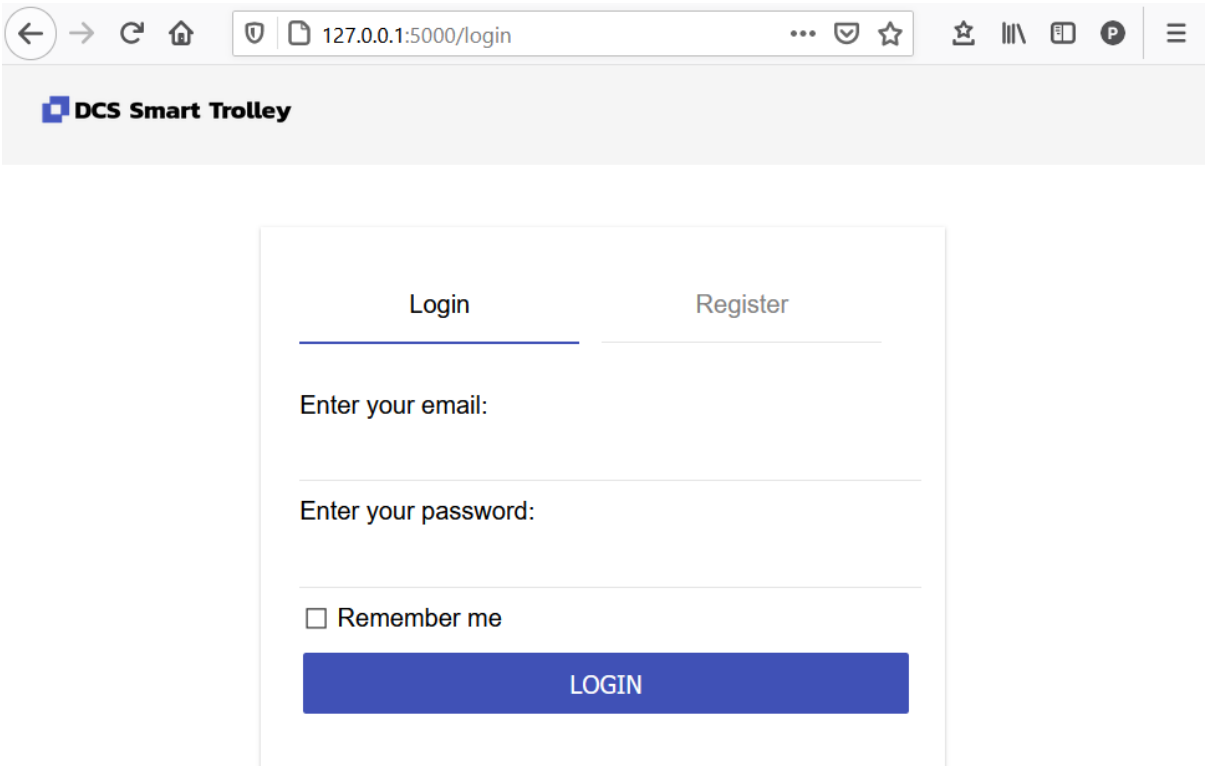


Figure Q1(a): The display of login.html

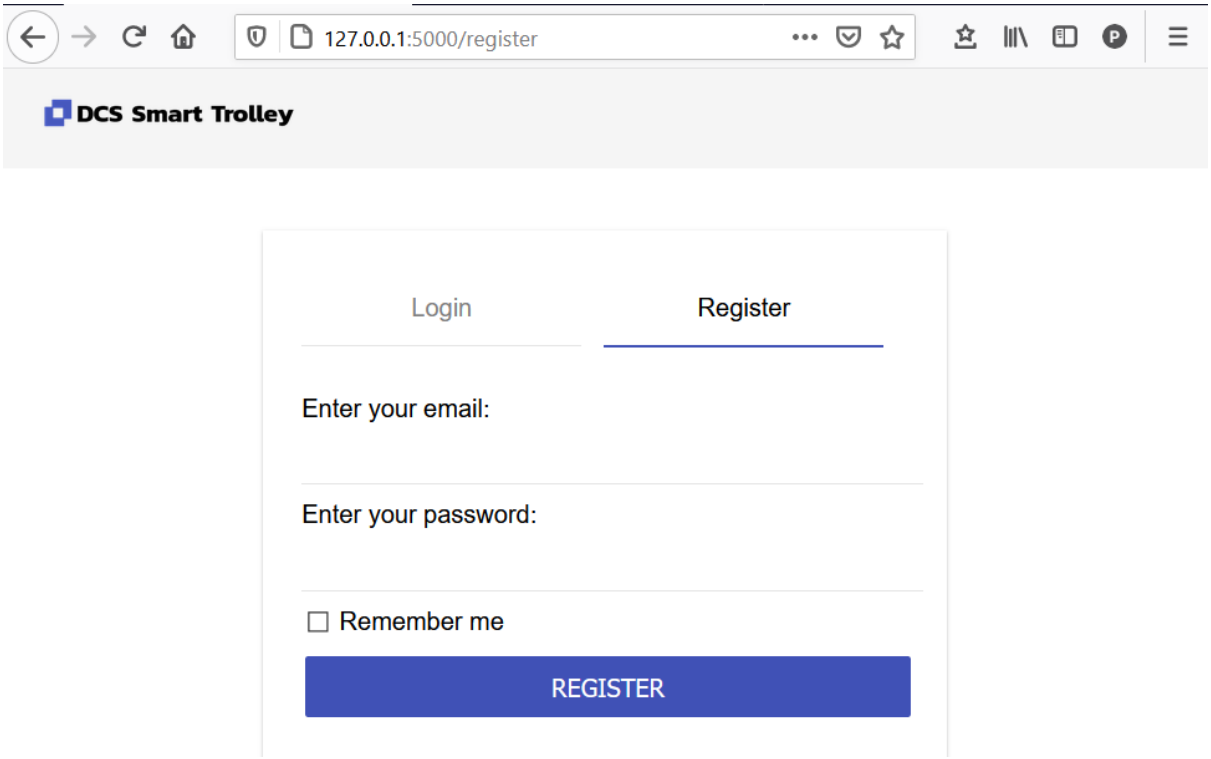


Figure Q1(b): The display of register.html

Q3

User login management is an essential component of Website development, Seminar 5 in the review materials you are introduced to a module called loginAuth.

The recording of Seminar 5 can be found below, the explanation of loginAuth is from 15:00 onwards.
https://suss.zoom.us/rec/play/7MgF6XmtioS4MZrPW-xEYxIq2jMqcpsuqf9mDdxPNy5LFLmn2dqf1iJaAWN4W-S_7BZWW_KbNAHVu0T3.4Nr-JPL9PeZ5Of8r

The source codes under discussion can be found under the subfolder loginAuth of the zipped source code file.

Employ Flask Framework to organize files for 'data model, **models.py**, 'routes/controllers', app.py, and 'view templates', and graphics and JavaScript following the following directory structur, including all the necessary libraries specified in the file "requirements.txt" to setup the python virtual environment under the "venv" directory accordingly.

```
/home/user/Projects/ECA
└─ Q1/
    ├── models.py
    ├── app.py
```

```
├── seed1.csv
├── templates/
│   ├── index.html
│   ├── log.html
│   ├── register.html
│   ├── upload.html
│   └── dashboard.html
├── static/
│   ├── logo.png
│   ├── style.css
│   ├── log.js
│   └── dashboard.js
├── requirements.txt
└── venv
```

(a) Described the content of the requirements.txt and the process as how the python virtual environment can be set up.

(3 marks)

(b) In order to include the BMR calculation, the register page view, namely, register.html, needs to be modified to include the height and date of birth information of a registered user as shown in Figure Q1(b) below:

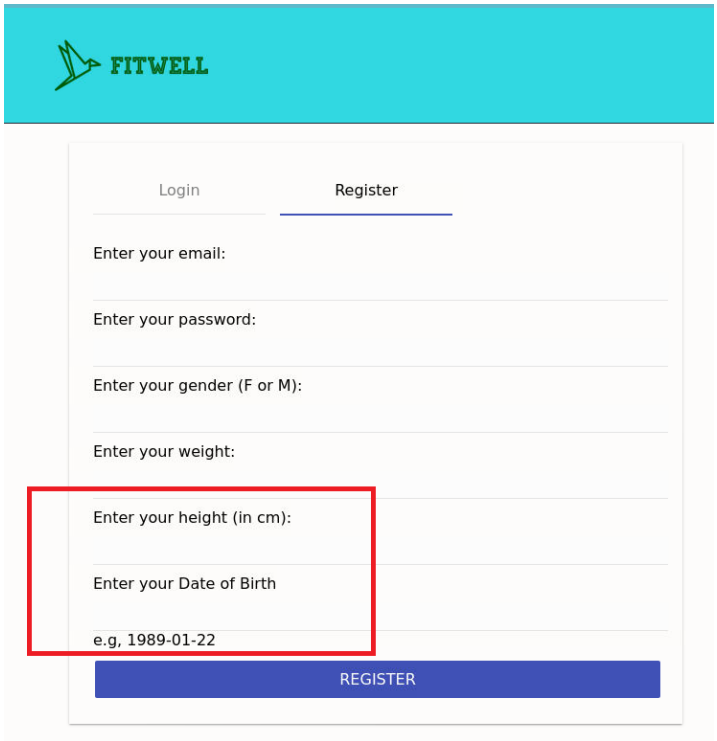


Figure Q1(b)

(2 marks)

(c) At the backend server side, the logic in "app.py" and data model "models.py" also needs to be re-designed to cater for this new formula for computing the total daily calorie consumption.

Let's assume the basic schema for activity log data remain the same as those in TMA as follows:

```
email,datetime,weight,walking,running,swimming,bicycling
```

Describe the data model that has been implemented in your TMA following the MongoDB terminology:

(i) What are the names of the collections?

(3 marks)

(ii) What are the fields of the documents in each of the collections?

(3 marks)

(iii) How the above data model needs to be redesigned to cater for new information and calculation needed?

(4 marks)

(iv) Justify the re-design of the data model described in Q1(c)(iii) above.

(5 marks)

(v) Implement the data model as redesigned in the point Q1(c)(iv) above in the python programs models.py based on MongoDB.

(10 marks)

(vi) Implement the logic to keep the relevant data updated given the upload and log functions that would update the data in the data model, and display the correct information when dashboard function is performed.

(10 marks)

Question 2

Improve on the solution to Question 1, construct a website by introducing new and revising existing programs based on the following directory structure

```
/home/user/Projects/ECA
├── Q1/
│   ...
└── Q2/
    ├── models.py
    ├── auth.py
    ├── app.py
    ├── app.py
    ├── seed1.csv
    ├── templates/
    │   ├── base.html
    │   ├── index.html
    │   ├── log.html
    │   ├── register.html
```

```
│   ├── upload.html
│   └── dashboard.html
├── static/
│   ├── logo.png
│   ├── style.css
│   ├── log.js
│   └── dashboard.js
├── requirements.txt
└── venv
```

(a) Incorporate flask_login module, in **auth.py**, into the App to provide access control and contextualize the information presentation and visualization for each user.

(7 marks)

(b) Refactor the html's files by introducing the **base.html** template that would be inherited by the other html files for the various functional views.

(7 marks)

(c) When a user visits the website, the landing page will only present two web page views, namely, the login and register views. If other page views are attempted by a user who is yet to logon, an error message will be prompted to ask the visitor to register and login to the App.

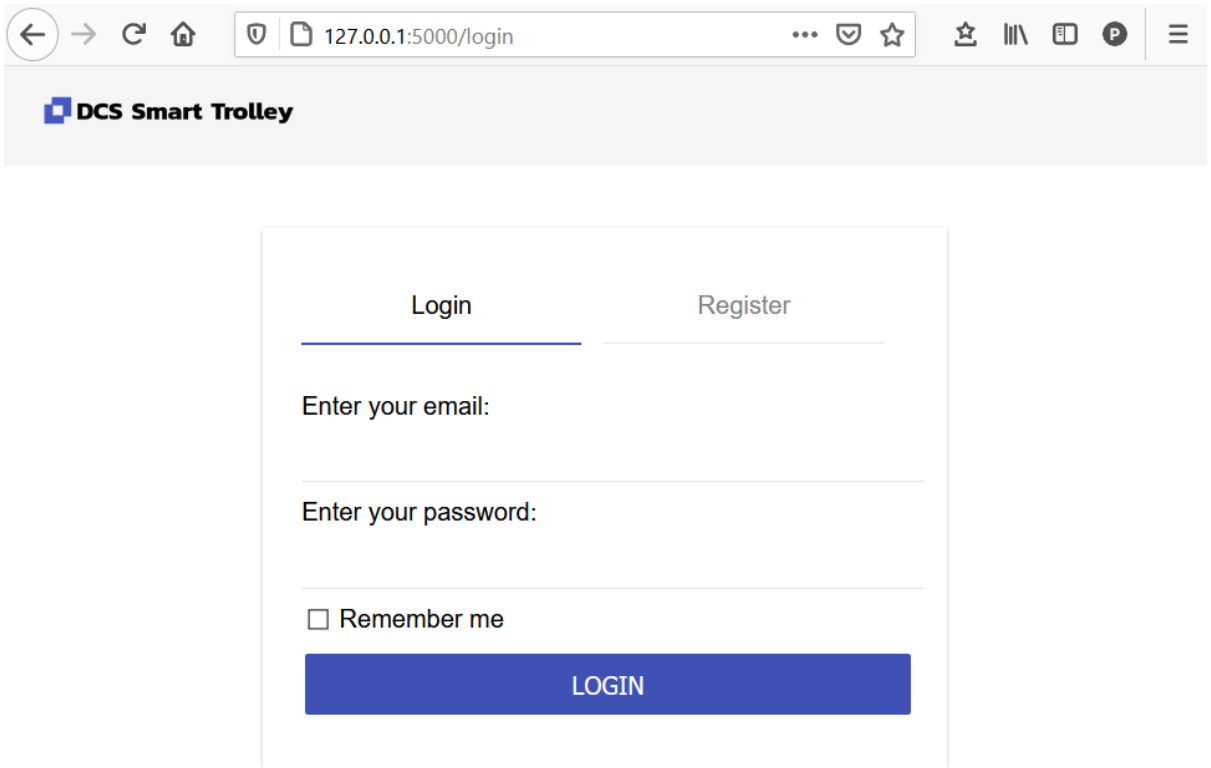


Figure Q2(c)(i): Only Two Active Links for a Visitor

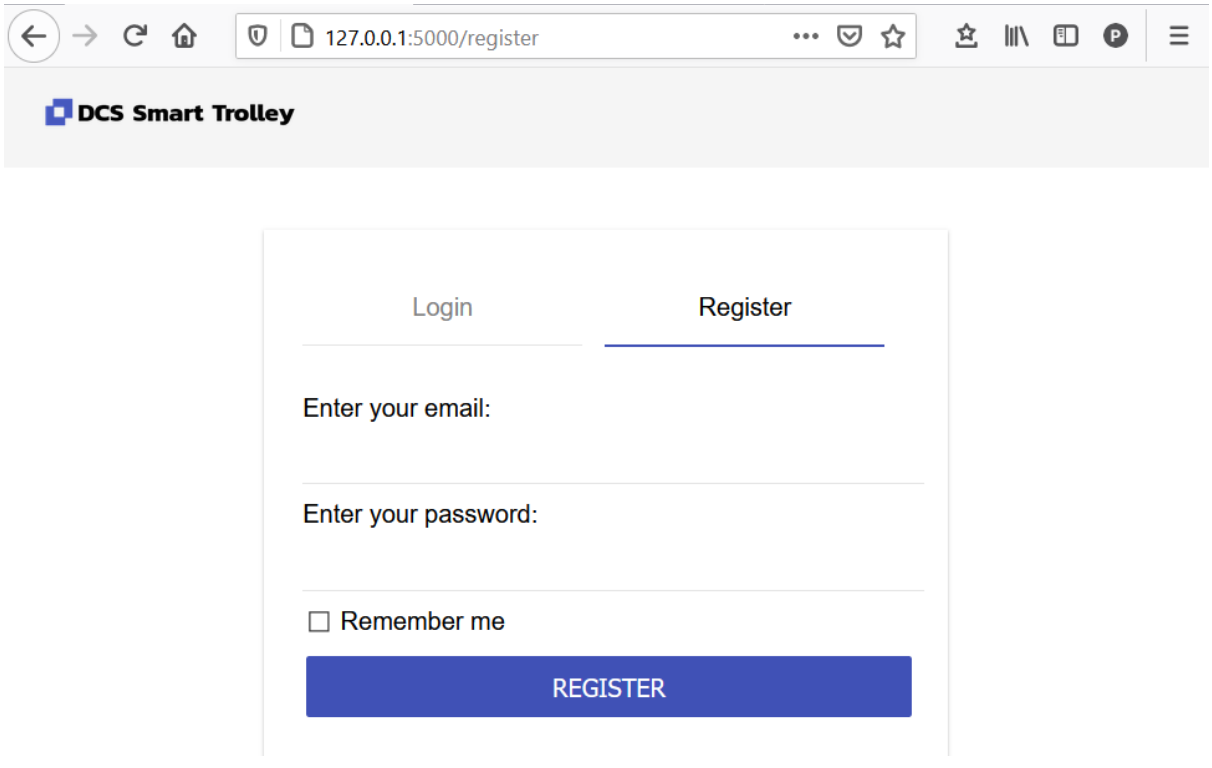


Figure Q2(c)(ii): Visitor is asked to log in to access restricted pages

(6 marks)

(d) Once the registration is done, a visitor can login the Apps as a user given that the password information is being authenticated. If the authentication fails, the visitor would be presented with an error message to re-attempt.

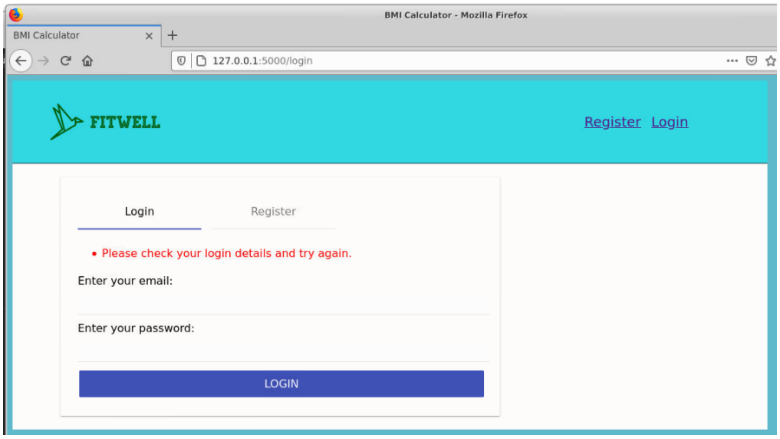


Figure Q2(d): Error Message is presented for failed authentication

(5 marks)

(e) If the user has been successfully authenticated, the log page view will be presented, and the sidebar will show corresponding personal information of the login user. The corresponding links that are accessible to the user will also be shown accordingly. Note that only the user admin@fitwell.com has the rights to upload calorie consumption data in batch mode through a file. The log function behaves the same as that was required of the TMA solution.

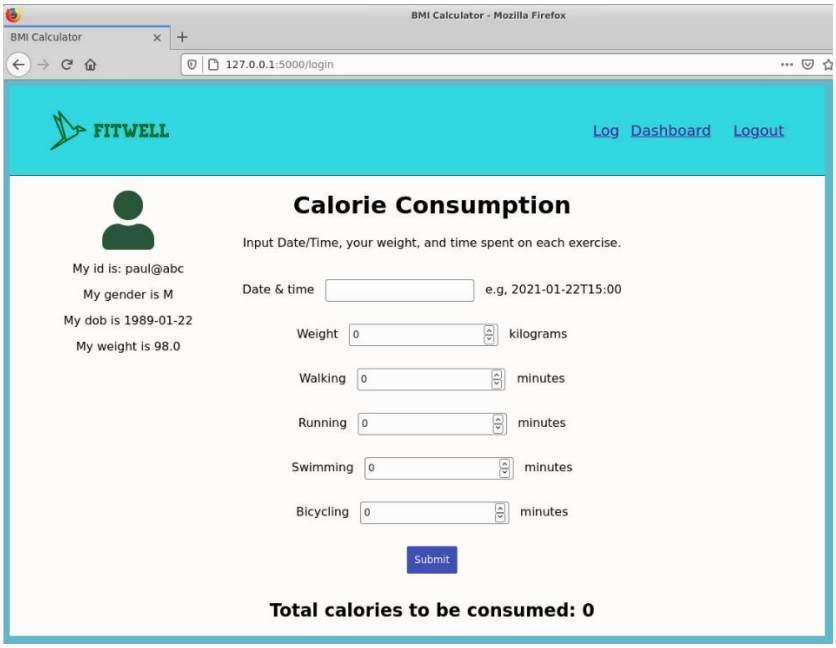


Figure Q2(e)(i): General users can access Log, Dashboard, and Logout

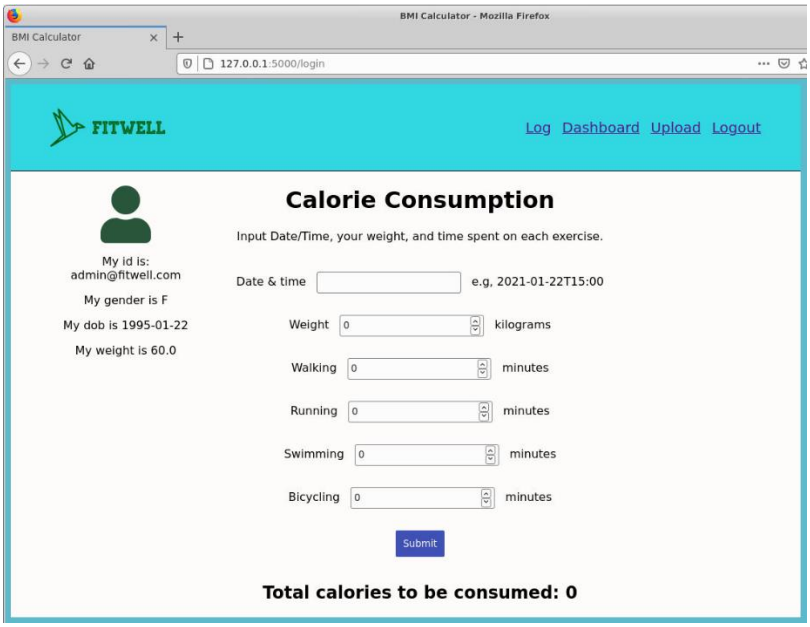


Figure Q2(e)(ii): Admin user can access Log, Dashboard, Upload and Logout

(5 marks)

(f) When the Dashboard page view is visited, the calorie consumption pertaining to the particular user will be displayed. However, if the login user is admin@fitwell.com all the users' consumption data will be displayed at the same time.

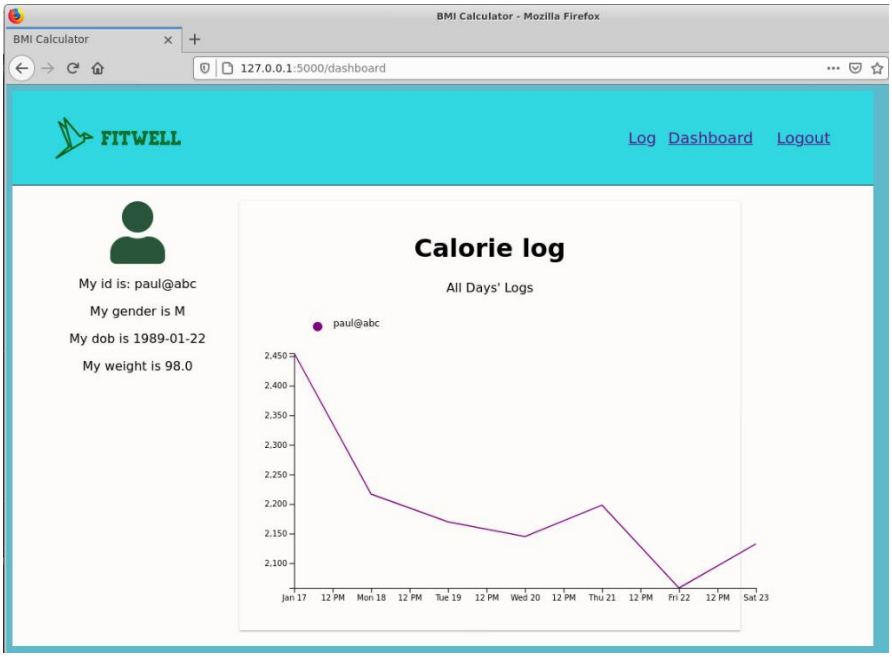


Figure Q2(f)(i): General users' dashboard shows the user's own consumption

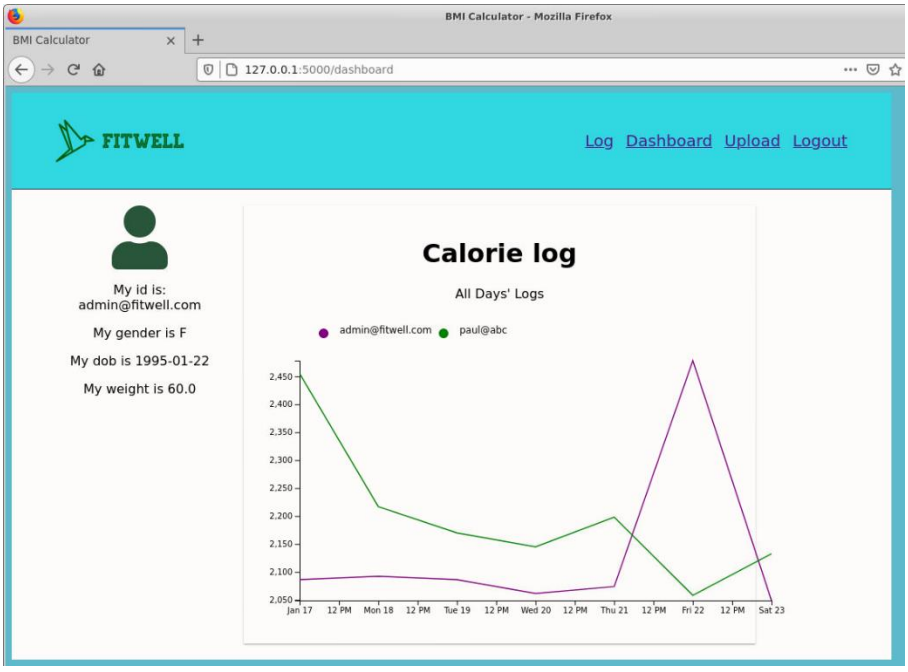


Figure Q2(f)(ii): Admin user's dashboard shows all user's consumption

(10 marks)

Question 3

The dashboard views have been quite focused on the trends of calorie consumption. You are asked to design new visualization functions that may help users track their health even more effectively.

(a) Propose **FIVE (5)** potential new ways to capture or/and visualize data that would help the users to managing their personal health more effectively. Justify your proposals.

(10 marks)

(b) Implement and experiment with **ONE (1)** of the proposal to demonstrate how the App can be improved.

(10 marks)

----- END OF ECA PAPER -----