

## ICT239 Web Programming

---

### Tutor-Marked Assignment July 2020

---

#### **STRICTLY CONFIDENTIAL**

Under no circumstances should students be allowed to see the contents of this booklet. Should this booklet, or any part of it, fall into the hands of a student, SUSS must be informed immediately.

## TUTOR-MARKED ASSIGNMENT (TMA)

This assignment is worth **18 %** of the final mark for **ICT239, Web Programming**

The cut-off date for this assignment is **Monday, 19 October 2020, 2355hrs.**

### Note to Students:

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

---

*Answer all questions. (Total 100 marks)*

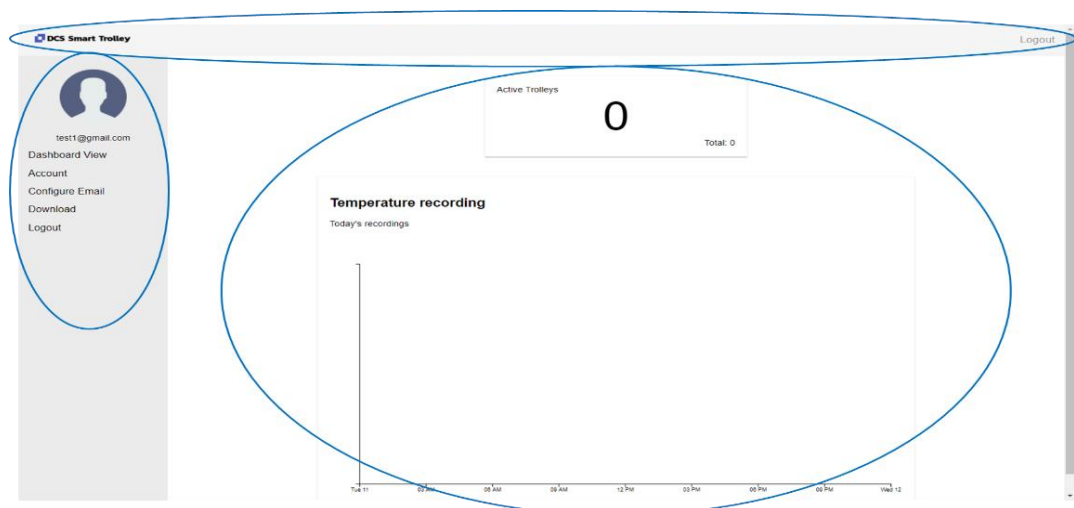
### **Question 1**

The Dietetics and Catering Services of a hospital is responsible for meeting the nutritional and dietary requirements of its patients. Food services play a crucial role in the facilitation of a faster recovery for patients and aiding in their treatment. The quality objectives of hospital food services are to provide the right food to the right patient at the right time while avoiding food wastage. Not only must the food provided be safe for consumption, it must also be palatable.

To meet these objectives, heated food trolleys are used in hospitals to keep the portioned food trays at a desired temperature before they are delivered to the wards. As part of audit requirements, hospitals are required to maintain a record of food trolley temperatures before the food is served to patients every day.

Please design a Web application to help the hospital staff for recording the temperature of food trolleys. This application can reduce the time and labour needed to record temperatures of food trolleys, as well as improve the accuracy of the data collected.

According to the following User Interface (UI) in Diagram Q1 below suggested by the project team, please setup the 'Dashboard View' page accordingly. Please create a folder named 'q1' and save all the relevant resources in the subfolders like 'q1/template', 'q1/static', etc.



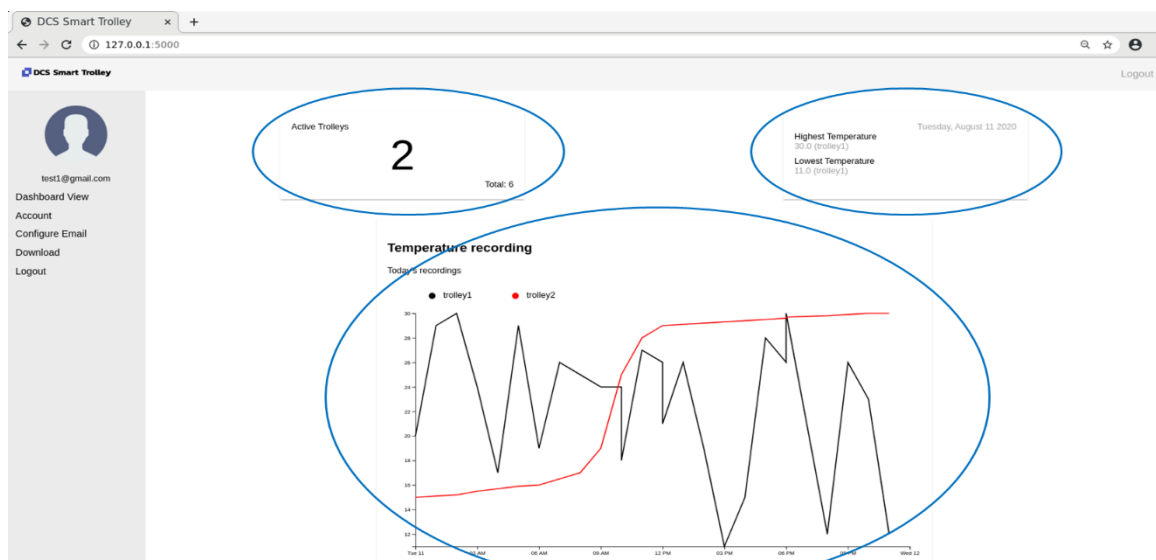
**Diagram Q1**

- (a) Apply HTML and CSS to create a corresponding .html file and a .css file. Do note that the 'Dashboard View' page contains **THREE (3)** major sections which are circled in the above layout.
- 'Description title bar': including project logo and name;
  - 'Side navigation bar': including user profile, dashboard view, account, configure email, download, logout;
  - 'Content area': including active trolley tab and temperature recording chart.
- Please create the **THREE (3)** sections with the exact proportions shown in the diagram. Create your own image for icons if required.

(12 marks)

- (b) Update the .html and the .css file in Q1(a) to create a demo 'content area' shown exactly as Diagram Q1(b) below. In the content area, there are **THREE (3)** circled visual elements.
- Active Trolleys tab: show the number of the current active trolleys and total trolleys;
  - Summarization tab: show the trolley with the highest and lowest temperatures for a specified date;
  - Chart tab: 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.
- Can display fake data but the values should relate among three tabs. No limitation on the selection of charting libraries.

(12 marks)



**Diagram Q1(b)**

## Question 2

The system aims to analyse the trolley temperature information for different dates. Hence, the web application should store the trolley temperature data with proper attributes.

For example:

Trolleys – store trolley identity as well as the latest status;

Stats – store the temperature summarization information for each date;

Recordings – store the history temperature recordings for each trolley.

- (a) Employ Flask Framework to design the necessary MVC models, controllers and views. Please illustrate your plan on how to implement MVC in your flask application. The names and attributes of the planned ‘data model’, ‘routes/controllers’, and ‘view templates’ should be listed out.

(9 marks)

- (b) Establish a new folder named ‘q2’ and create the Flask MVC application by re-using the code in Q1. Please download the demo dataset from the following URL link and update the datetime column with your current date.

[https://raw.githubusercontent.com/LiuFang00/ICT239/master/seed\\_demo\\_TMA.csv](https://raw.githubusercontent.com/LiuFang00/ICT239/master/seed_demo_TMA.csv)

Organize the demo dataset with regards to your data model plan in Q2(a). Store the organized datasets in the backend. Please use MongoDB to store your data, both local and online versions are allowed.

(10 marks)

- (c) Let the Flask application implement all UI elements required in Q1(b). Here, all the displayed data values should be retrieved from the backend data models created in Q2(b). Here, we suppose ‘today’s date’ is 11 Aug. 2020.

(8 marks)

- (d) Create the chart based on your understanding of Scalable Vector Graphic (SVG). Make sure the chart properly demonstrating the value retrieved from the backend data models. Here, we suppose ‘today’s date’ is 11 Aug. 2020.

(8 marks)

### Question 3

The system would also like to support two ways of data upload functions. One way is to collect records from user input; the other way is to load records from external files. Please refer to the following Diagram Q3 and answer the questions.

**Create a recording**

Trolley

Date & time  
mm/dd/yyyy, --:--:--

Temperature

Record

**Upload recordings**

Upload CSV file e.g. "trolley1,2020-06-05T22:19,24"

Choose File No file chosen

Upload

**Recordings**

Trolley	Date & time	Temperature
trolley1	2020-06-30T00:00	20.0
trolley1	2020-06-30T01:00	29.0
trolley1	2020-06-30T02:00	30.0

**Diagram Q3**

- (a) Build on your Flask application designed in Q2, design a route to render a new .html template which impalement the UI elements for trolley temperature collection. **THREE (3)** major sections should be involved.
- Create a recording: allow a user to key in a trolley temperature with selected datetime;
  - Upload recordings: allow a user to upload trolley temperatures from an external .csv file;
  - Recordings: show all uploaded recordings in a table.

One button is required to choose file and two buttons are required to confirm Record/Upload records.

(15 marks)

- (b) For the 'Create a recording' section, please process user input data properly with 'GET' or 'POST' method. The 'date & time' tab should support both user type in and selection from a calendar shown in Diagram Q3(b) below. After click the 'Record' button, the data should be stored in the backend side, and displayed in the 'Recording' section accordingly.

(13 marks)

DCS Smart Trolley

## Create a recording

Trolley

Date & time

dd/mm/yyyy --:--:--

August 2020

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Today

11	49	AM
12	50	PM
01	51	
02	52	
03	53	
04	54	
05	55	

Record

Cancel

**Diagram Q3(b)**

- (c) For the 'Upload recordings' section, please allow a user to select local files from your computer by clicking 'Choose File' button. The selected file name should be displayed as shown in the circle below. Once the user clicks the 'Upload' button, the file should be saved in the backend and the corresponding 'Recording' section should be updated with new records.

(13 marks)

## Upload recordings

Upload CSV file e.g. "trolley1 2020-06-05T22:19,24"

Choose File seed\_demo\_TMA.csv

Upload

**Diagram Q3(c)**

---- END OF ASSIGNMENT ----