CS 396 Term Project

Personalized Information Organizer

**Overview**

In this project, we aim to build a web-based system to help a user keep his/her personalized inform such as dairy, calendar, contacts, etc. The project has three phases: web server, database, and web interface.

**Phase 1 (individual work): Building Website Using Django (Due on Sep. 26)**

The goal of this project is to build a website which can provide services of forum, documentation, searching, linking, etc. Please use Django to finish this project and you can find some tutorials here <https://docs.djangoproject.com/en/2.2/intro/tutorial01/>

Your project should be able to do the following tasks:

1. Create an administrator who can delete other users, delete/edit all posts, create new discussion topics
2. Provide login function, and sign-up function for new users.
3. Any user can create a new post, and reply other users’ posts.
4. Any user can upload files (could be multiple files).
5. Show the numbers of posts and visits, and keep these numbers updated.
6. You can store some files on your website
7. Include a link to google calendar in a drop list.
8. Show a warning a message/sign to remind users that there is an update or something interesting posted in the forum (words/content/…etc).
9. Your Website interface should be friendly.

Please submit a zip file, including 1. ReadMe 2. Project report 3. Source codes.

ReadMe file includes major instruction steps for repeating your project results. Anyone should be able to follow the instructions and run your source codes correctly. Please do not forget system setting up.

In project report, please include:

1. Project description

2 Functions that you have implemented

3. Technical details: Database server, Software, language/scripts

4. Features you like to highlight.

5. Discussion: what techniques/languages/scripts are better for a particular task? Comparison between different techniques/languages/scripts.

**Phase 2 (individual work): Database Management System (Due on Oct. 24)**

You will build a database system to maintain the following information.

**Core information**

Activities: types of activities, time (start and end), location, description of activity.

Shopping: item, quantity, price, store, related activity

Meeting: Meeting name, ID, number of participant, name of participant, meeting materials (presentation, documents)

Health: doctor name, phone number, email, address, medicine, dose, time to take the medicine.

**Elective information (you choose at least one)**

Details of items and/or medicine and/or others. For an example of item, you can include different stores, prices, advantage and disadvantages of the item. For medicine, you can include name of the medicine, function, side-effective, price, brand, pharmacy,

Your database needs at least five tables and should be in the third normal form.

You database should be supported by the Django server that you have setup in phase 1.

**Phase 3 (individual or team work): Web interface**

The goal of this project is to provide users visualized results of queries to the database that you built in Phase 1 &2.

The system includes:

1. A Django server supporting all requesting services
2. A browser-based interface allows the user to query and display the corresponding results (The system keeps the functions you have finished in Phase2.)
3. Data visualization (the detailed requirements are given in the following.)

Particularly, your system will

1. Show a calendar which you can input your activities.
2. Upload a file (e.g. csv, txt, xlsx, etc) and automatically save the data into the database.
3. You can click on the activity on the calendar to find the details. For example, if you click on meeting activity, you will see details of the meeting.
4. Show a bar chart of spending type vs amount (food, entertainment, travel, sport, education, etc, at least five types).
5. Send a message/email to remind the user it is the time to do the activity on the calendar (for example, it is the time to take the medicine).

<div class="mb-3">

<label for="formFileMultiple" class="form-label">Multiple files input example</label>

<input class="form-control" type="file" id="formFileMultiple" multiple>

</div>