Faculty of Information and Communication Technology

ITCS223 Introduction to Web Development

Project Phase II (62 pt. + 3 extra pt.)

Assigned Date: January 29, 2025

In this team project, you are about to develop a web application for the selected business domain, such as online shopping, music and entertainment, food services, etc.

Project Description

Phase II allows you to implement the front-end and back-end parts of your web application designed in Phase I. You will implement the web application as follows:

- Front-end pages use HTML, CSS, and JavaScript; all pages are hosted on the front-end server.
- The back-end of your web application, including the database and web services, is hosted on the back-end server, which uses Node.js.
- In addition, you will interact with your own web services and a *public* web service using JavaScript.

The simple architecture of this system is shown below. With a group of 4 - 5 students, there are four tasks to complete:
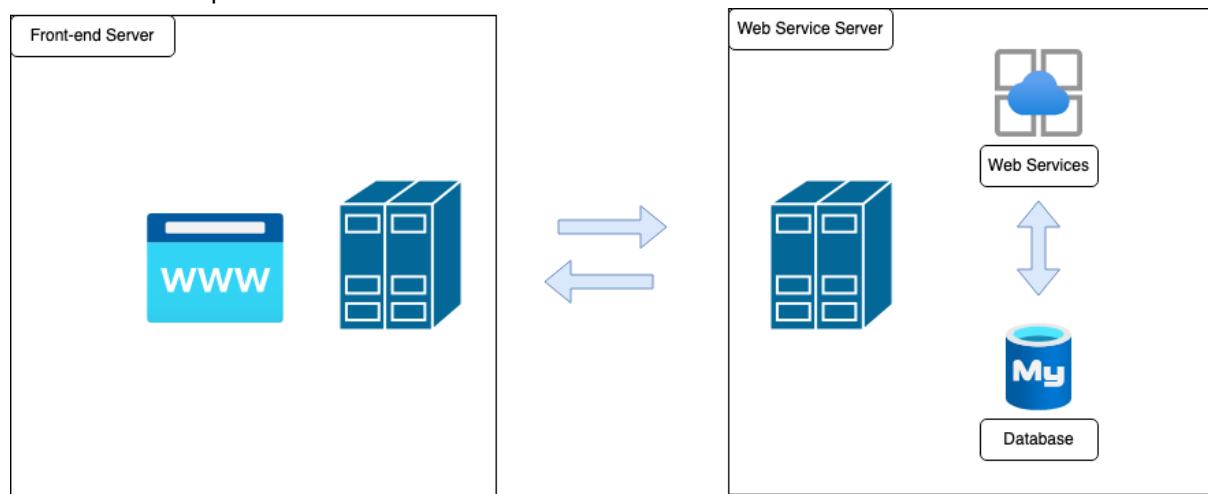
Task 1: Front-end Implementation

You will implement all the web pages designed in Phase I. Minor changes in design are acceptable. Each page should have the following details:

1. **Home Page:**
   - The page **must** have <u>a navigation bar</u> indicating the sitemap to other pages.
   - The page **must** have a link or button for accessing the <u>Login</u> Page
   - Include some content (e.g., texts or images) that is meaningful and related to your business domain.
2. **"Login" Page for Administrators (1 page)**

- The page should have a form consisting of input boxes and buttons for authentication.
- The page must connect to the homepage/navigation bar.

3. **"Search" Page for searching products/services (1 page)**
    - The page should have a form consisting of a search box and criteria for searching (e.g., search by product name, product type). The search results will be shown on the same page.
    - There must be at least **three** criteria used for searching (exclude No criteria search), for example, search by name, search by rating, search by product/service types. **Note**: searching and filtering are different.
    - The search result will be shown as a creative table or list on the same page.
    - Each product or service of the output should go to its "Detail" page (4)

4. **"Detail" page for the product/service result**
    - The detail page shows the information regarding the selected product/service.
    - The detail page should link to the previous search result from the "Search" page (3)
    - This page should be a template for the details to show the information fetched from the web services.

5. **"Product/Service Management" page for administrators**
    - The page provides a form to add a new product/service.
    - The page should allow the administrators to modify the existing products/services.
    - The page should allow the administrator to delete specific products/services.

6. **"Team" page (1-5 pages)**
    - Have the information in detail about your team members.
    - (*Optional*) Each member can have his/her own individual page.

Additional requirements for your web application
- Every page must have access to the navigation bar (i.e., link back to your home page and other pages)
- Each page's code must be well-structured (indented) and have a reasonable number of comments.
- Each page must use semantic elements (e.g., <header> <footer> <article>)
- Each page must have either **external or internal CSS**. You are welcome to create your own CSS, apply any CSS frameworks, or the ready-made CSS style. To make sure you understand the code.
- The front-end part needs to have its own web server.

Task 2: Database Implementation
The database needs to be implemented in DBMS, preferably MySQL. Your database must store:
1. Administrator information (e.g., first name, last name, address, age, and email),
2. Administrator login information (e.g., username, password, role, and log-in log), and
3. Your business domain data (e.g., product info, services).

Note that you will use the database you designed in Phase I. In addition, you must use **meaningful data** (at least 10 data for each table) in your database.

Task 3: **Web Services**

You are required to create web services to handle the following functions:

1. Authentication web service for administrators. This web service will be called when the administrator's log in to the web application.
2. Product/Service Search and Details: This web service will be called when users search and view products or services in the system. The search function supports the following functions:
   - **No criteria search**: the web service will return all results.
   - **Criteria search**: The web service must support <u>at least three criteria</u>. Therefore, when the users make a search request, they can enter all criteria or some criteria.
3. Product/Service management web service for administrators. The web services allow the administrators to:
   - Insert products or services in the system
   - Update products or services in the system
   - Delete products or services in the system

In addition, you need to provide at least 2 test cases for each service in the comment above the services. Your service will be tested by **Postman [ref]**. For example,

```
// Testing Insert a new Student
// method: post
// URL: http://localhost:3000/student
// body: raw JSON
// {
//    "student": {
//        "STU_ID": 99998,
//        "STU_FNAME": "Wudhichart",
//        "STU_LNAME": "Sawangphol",
//        "STU_AGE": 35
//    }
// }
```

Task 4: **Web Services Interaction**

This task aims to connect your front-end implementation to the web service in Task 3 using JavaScript, e.g., **Fetch** API. You are required to create connections from the interface to the web services for all functions and pages. In addition, your web application needs to connect to **one** useful public web service. For example, Google Maps can be used to show the location of stores.

<u>Note that the front-end (UI) and back-end (Web service) need to be implemented on different web servers.</u>

Task 5: Report

This task requires students to write a report of their work in project phases I and II. The report needs to include:

1. Cover page that contains Faculty name, Unit name, and the information of members
2. Project Overview
3. Navigation Diagram of your web application
4. Details of your web application and code
5. Details of your web services and code
6. The testing results of web services using Postman or any program for testing web service

Additional requirements for your web application and web services
- All web services **must** be implemented using **Node.js or other JavaScript frameworks.**
- All web services **must** be connected to the database server.
- The front-end (UI) and back-end (Web service) need to be implemented on <u>different web server</u>s.
- You are required to ensure that your submission can be deployed in the machine of lecturers or LAs.
- The project must have a **README.md** explaining how to run/start your web application and web service. Please clearly state in the readme file if there are extra remarks, bugs, or exceptions.
- The project must be no larger than 50MB. If you have many images, please store them in external cloud storage and use the URL for references.

<mark>*** Please read the marking criteria for the project below ***</mark>

Extra Task (3 pt.)

If your web application and web services are deployed on the provided server, you will receive an extra point (3 pt.).

To get this extra score, you **must first finish all requirements above**. In addition, the additional task must be expressed clearly in the report and README.md.

Instruction for Submission

1. For the first time, when you access Project Phase II on GitHub Classroom, the student who has the smallest ID needs to create a team. The team's name needs to be in the format of **SectionX_GroupY**, where X is your section, and Y is the team number. For other team members, you need to select the correct team on GitHub Classroom.
2. The students must submit the code and report of tasks 1, 2, 3, and 4 on GitHub Classroom and README.md to describe the installation and how to use your web application.
3. On GitHub Classroom, it will consist of:
   a. For tasks 1 and 4, you must submit all front-end source files, encompassing HTML, CSS, and supplementary files (e.g., images), **excluding the 'node_modules' directory**. Rename your source code directory to secX_grY_fe_src.
   b. For task 2, you need to export your database as an SQL file, rename your SQL file to **secX_grY_database.sql,** and submit it to GitHub Classroom.
   c. For task 3, you need to submit all web service source files **except the 'node_modules' directory**. Rename your source code directory to **secX_grY_ws_src.**
   d. For task 5, the students must submit the report in PDF format on the repository.

<mark>Sunday, May 4, 2024, at 11:59 PM BKK Time (UTC+7)</mark>
(For all sections)
<mark>Note: Late submissions will be accepted.</mark>

Academic Integrity

Please do your own group work. Your survival in the subsequent courses heavily depends on the programming skills that you harvest in this course. Though students are allowed and encouraged to discuss ideas with others and search for online sources, the actual solutions must be written by themselves without being dictated or looking at others' code. Collaboration in writing solutions with other groups is not allowed, as it would be unfair to other students. It is better to submit a broken program

that is a result of your own effort than take somebody else's work for your own credit! Students who know how to obtain the solutions are encouraged to help others by guiding them and teaching them the core material needed to complete the project rather than giving away the solutions.

Front-end Marking Criteria (24 pt.)

All functions in the description need to be preserved. Otherwise, the score will be reduced accordingly.

	Criteria	Level 2	Level 1	Level 0
1.	**Homepage**	The webpage complies with all requirements, including a navigation bar, login, and content or images to explain the business.	The webpage complies with some of the requirements, including a navigation bar, login, or content or images to explain the business.	**No** page is implemented
2.	**Login page**	The webpage includes a form with an input box and a button for logging in, and it connects to the main page of the website (Home Page).	The webpage is missing some requirements, such as a form with an input box or a button for logging in, or it does not connect to the main page of the website (Home Page).	**No** page is implemented
3.	**Search page**	The webpage meets all the requirements, including having a form with an input box and a button for searching, providing at least three search criteria, displaying simulated search results, and linking the results to a detailed product or service page.	The webpage is missing some requirements, such as having a form with an input box or a button for searching, providing at least three search criteria, or displaying simulated search results, or linking the results to a detailed product or service page.	**No** page is implemented
4.	**Detail page**	The webpage meets all the requirements, including having images and details of the products or services, providing references for the sources of information, and connecting to the search results for the products or services.	The webpage is missing some requirements, such as lacking images or details of the products or services, or not providing references for the sources of information, or not connecting to the search results for the products or services.	**No** page is implemented
5.	**Product/Service Management page**	A page has **all** required details	A page has **some** required details	**No** page is implemented
6.	**Team page**	The webpage meets all the requirements, including providing information and images of all group members, as well as links to at least one personal social media account for each member.	The webpage is missing some requirements, such as information and images of some group members or links to at least one personal social media account for each member.	**No** page is implemented
7.	**Semantic elements**	The semantic elements are applied to **all pages**	The semantic elements are applied to **some pages**	**No semantic element** is applied at any pages.

	Criteria	Level 2	Level 1	Level 0
8.	**Navigation bar**	Have a navigation bar for every page and all pages are linked	Have a navigation bar for every page or all pages are partially linked	**No a navigation bar** is implemented
9.	**CSS**	**All pages** are applied the external or internal CSS correctly	**Some** pages are applied the external or internal CSS correctly	**No page** is applied the external or internal CSS (i.e. only use HTML or in-line CSS)
10.	**CSS Selector**	There is the use of element selection, class selector, and ID selector, with at least one instance of each.	There is a lack of use of element selection, class selector, or ID selector, or they are not fully implemented.	**No CSS selector**
11.	**readme.txt**	The steps for running the project are well-written and the project can be run smoothly	Incomplete or missing steps in the readme file	**No readme file or No instruction**
12.	**Source code and comments**	**All pages** are appropriately **indented** and **commented**	**Some** pages are appropriately **indented** and **commented**	**No page** is **indented** and **commented**

Web Services and Interaction Marking Criteria (28 pt.)

All functions in the description need to be preserved. Otherwise, the score will be reduced accordingly.

	Criteria	Level 2	Level 1	Level 0
1.	**Authentication Service**	**Authentication** Service is implemented, and it **works properly**.	**Authentication** Service is implemented, and it is **syntactically correct**.	**No Authentication** Service is implemented.
2.	**Search Service**	**Search** Service is implemented, and it **works properly**.	**Search** Service is implemented, and it is **syntactically correct**.	**No Search** Service is implemented.
3.	**Insert Service**	**Insert** Service is implemented, and it **works properly**.	**Insert** Service is implemented, and it is **syntactically correct**.	**No Insert** Service is implemented.
4.	**Update Service**	**Update** Service is implemented, and it **works properly**.	**Update** Service is implemented, and it is **syntactically correct**.	**No Update** Service is implemented.
5.	**Delete Service**	**Delete** Service is implemented, and it **works properly**.	**Delete** Service is implemented, and it is **syntactically correct**.	**No Delete** Service is implemented.
6.	**Test cases**	**All** test cases are provided.	**At least 50 %** of test cases are provided.	**No** test cases or less than 50% of test cases are provided**.**
7.	**Authentication Service interaction**	**Authentication** Service interaction is implemented, and it **works properly**.	**Authentication** Service interaction is implemented, and it is **syntactically correct**.	**No Authentication** Service interaction is implemented
8.	**Search and View Service interaction**	**Search and View** Service interaction is implemented, and it **works properly**.	**Search and View** Service interaction is implemented, and it is **syntactically correct**.	**No Search and View** Service interaction is implemented
9.	**Insert Service interaction**	**Insert** Service interaction is implemented, and it **works properly**.	**Insert** Service interaction is implemented, and it is **syntactically correct**.	**No Insert** Service interaction is implemented

Criteria	Level 2	Level 1	Level 0
10. Update Service interaction	**Update** Service interaction is implemented, and it **works properly**.	**Update** Service interaction is implemented, and it is **syntactically correct**.	**No Update** Service interaction is implemented
11. Delete Service interaction	**Delete** Service interaction is implemented, and it **works properly**.	**Delete** Service interaction is implemented, and it is **syntactically correct**.	**No Delete** Service interaction is implemented
12. Public Web Service interaction	**Public Web** Service interaction is implemented, and it **works properly**.	**Public Web** Service interaction is implemented, and it is **syntactically correct**.	**No Public Web** Service interaction is implemented
13. Submission	**All** submitted files are in the **correct format and named**	**Some** of the submitted files is in the **correct format and named**	**No** submitted file is in the correct **format** and **named**
14. Source code	All necessary blocks of code have comments.	Some necessary blocks of code have comments.	There is no comment.

Note:
- "Work properly" means that when we test the web services, they should return the correct results.
- "Syntactically correct" means all the codes for each web service are written correctly, but they cannot be tested or returned a result.

Report Marking Criteria (10 pt.)

Criteria	Level 2	Level 1	Level 0
1. Title Page	All required information is well-written.	Most of the required information is well-written.	Missing information.
2. Organization	Proper formatting, sections clearly labeled, well-organized, professional style	Some formatting errors or missing sections	Multiple formatting errors
3. Topics	All the requested topics are covered in good detail.	Some requested topics absent or details lacking.	Multiple topics lacking or insufficient details.
4. Grammar, Usage, Mechanics, Spelling	No errors	Only one or two errors	More than two errors
5. Quality of information	Interesting and informative.	Some details vague	Details somewhat sketchy