**Project Phase Two** (System Development)

Keep all your work in the repo; do **not** submit anything on Canvas.

|  |
| --- |
| **Task 1 - Client Side** (Front End) |

•In this phase, the development of the client side (front end) should have major progress.

•Your phase two presentation should include the following parts of the client side:

1)A front-end **system architecture diagram** that shows the components (modules) of the front end of the system, as well as their relationship (how they interact with each other)

2)**Screenshots** of the Graphical User Interface (GUI) of the system (multiple screenshots to show the different pages/parts of the GUI **required**)

3)A list of the **APIs/technology** used to implement the front end of the system

|  |
| --- |
| **Task 2 - Server Side** (Back End) |

•In this phase, the development of the server side (back end) should have some progress.

I am **not** expecting to see a running program in this phase, but at least you should start the development of the "main engine" of the system.

•Your phase two presentation should include the following parts of the server side:

1)The **database system** used to handle the input data, output data, and external data in the system

2)A **data flow diagram** that shows where (in which components) the data are stored and how the data are transmitted from one component to another

3)The **queries** to the database in the system (design of the queries **required** in this phase, implementation of the designed queries **not** required in this phase)

|  |
| --- |
| **Task 3 - Maintenace of the GitHub Project Repository** |

•Source code should be stored and well maintained in the GitHub project repository.

•Every developer should directly use GitHub to accomplish his/her development work.

|  |  |
| --- | --- |
| **Good Example** | Every developer pulls the updates of the remote repo to his/her local repo, works on the development in the local repo, then commits and pushes his/her work to the remote repo. |
| **Bad Example** | A designated developer handles all the git issues.  Other developers work on the system development locally and send their code to the designated developer.  Then, the designated developer combines all other developers' code and commits and pushes the updated project to the remote repo one hour before the deadline. |

•Source code and documents should always be consistent in your GitHub project repository.  If during the development process, your code is no longer consistent with the documents (e.g., change of system architecture, adding/removing APIs, modified database queries), the developers need to let the project manager know and the team should update the documents in a timely manner.

Keep all your work in the repo; do **not** submit anything on Canvas.