Web Dev Project Requirements

Requirement Timelines:

1. Lab 9 (18/05-21/05) (No Assessment):

- Each team needs to be scheduled to meet me offline preferably (in the worst case online) for 10 minutes to discuss the idea and show me what you have got so far (your progress).
- This is no assessment for this so just be honest about your status!
- All reports, description, code must be in your github repository:
    - Your source code is in the github repo
    - Your required report, description, links, diagram,...(everything) must be in the format of the README file of the github repo.
    - In short, I only accept one single github link which contains everything of your project!
- Your README file should contains the draft information as following:
    - Describe your idea (Super Important):
        - Describe your topic to me.
        - Your motivation for why your team chose this topic.
        - Show me images/videos of similar topics.
        - Show me whatever you got so far!
    - What kind of features you want to have.
        - Primary features (must-have)
        - Secondary features (nice-to-have)
    - Mention the language , libraries, platform or framework to build your application/topics and why do you choose those?
    - Break down the application/topic into many sub tasks and assign the sub tasks among team members:
        - List team members.
        - Responsibilities for each team member..
        - Make sure all your team members are contributors of that github account.
        - Each team member at least commits and pushes something on the repo.
    - Diagrams:
        - Use cases design.
        - Database design.
        - Interface design.
    - References:
        - Provide links you base your code here (to avoid plagiarism)
        - Useful links for instructors to understand your project!
    - Here is the suggested structure of the report:

- **CHAPTER I: INTRODUCTION**
  - Motivation
  - Problems Statement
  - Scope
- **CHAPTER II: LITERATURE REVIEW**
  - Similar Applications/Systems
  - Platform and Tools Review
- **CHAPTER III: SYSTEM DESIGN**
  - System Requirement Specification
    - Functional Requirements
    - Requirements Analysis
    - Non-functional Requirements
  - System Design Specification
    - Use-cases Diagram (summary goal use case and at least 3 main use cases)
    - Sequence Diagram (at least 3 main sequences)
    - Activity Diagram (at least 3 main activities) (optional)
    - Class Diagram
- **CHAPTER IV: SYSTEM IMPLEMENTATION**
- **CHAPTER V: CONCLUSION AND DISCUSSION**
  - List of accomplished work
  - Strength and Weakness
  - Future Work
- References

Tips and tricks for Github:

- **Pro Tip**: for github/git, if you don't want to learn or deal with git commands, command lines and terminal then you can use this free beautiful Git GUI, **GitKaren:**
  - **GitKaren** allows you to click buttons, drag and drop to manage your github repo.
  - **Resources for GitKaren:**
    https://www.gitkraken.com/download
    https://www.youtube.com/watch?v=AMFoJoNh3V4&ab_channel=GitKraken

- **Github tutorials:**
  - Tutorials to create github repo:
    https://docs.github.com/en/get-started/quickstart/create-a-repo
    https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/creating-your-first-repository-using-github-desktop
  - Tutorials to add people to be contributors to your repo:
    https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-user-account/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository
  - Tutorials to commit and push files to repo:
    https://www.datacamp.com/community/tutorials/git-push-pull

- **Github Readme tutorials:**
  - Tutorials to write README on github:
    https://www.nobledesktop.com/learn/git/create-a-readme-file

https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax
https://bulldogjob.com/news/449-how-to-write-a-good-readme-for-your-github-project
https://www.freecodecamp.org/news/how-to-write-a-good-readme-file/

- Here are the list of awesome README files of other projects on github: https://github.com/matiassingers/awesome-readme

2. In the final lab 12 (Assessment):

- Your team will present your project officially for marking in the final lab session 12.
- In fact, your lab session 12 to present your project will be on the dates for Tom's labs:
    - Wednesday morning lab.
    - Friday afternoon lab.
    - Saturday morning lab.

- **Submit your github on the Excel:** https://docs.google.com/spreadsheets/d/1GJ7EjOcGq-cnOLy6oBzRF4DZyD-myj81xP20wE_ktkk/edit?usp=sharing
- **Double check your team information correct in the provided excel:**
    - ID
    - Fullname
    - Email
    - Team Name
    - Lab Day

- **The timeline for each presentation:**
    - Your 10-minute presentation in the lab (maybe 7 minutes go through above info and 3 minute of live demo of your project).
    - **Data [Important]**: You must show your application can CRUD (Create Read Update Delete) with the data with user interaction on the front end!
        - Demonstration (must be included real data manipulation such as insert, view, update, delete, select, search, etc)
    - The PowerPoint slides are recommended as you should deliver your ideas/points better with slides but if you can figure out a better way to present your points then go for it!
    - 3-5 minutes at the end for Q&A session with your lab instructor.

- **You are expected to achieve and present the following:**
    - Complete your application with completely implemented features.
    - The completed report (README in your github) should have:
        - Introduction
            - Your team

        - Your topic
        - Motivation or why this topic
        - Task allocation for each team member.
    - Discuss about your technology stack selection (language/library/framework)
    - Show all required diagrams and explain them.
    - Show your github project page:
        - README (the report)
        - Team Commits in github
    - Feature:
        - Which features are completed
        - Which features are incompleted
        - Future features if you want to developer further after the course
    - Discuss challenges in the project.
    - Result and Conclusion
        - Screenshots/Videos/GIF of your topic to showcase all features.
    - What have you learned?

- **Marking Criteria (According to Dr. Sinh):**
    - 20% report (README)
    - 30% presentation
    - 50% demo

Consider this project as a project at work you are gonna have in the future! So it would be a great opportunity to practice to work as a team and get the project done!

Good luck with your project!

Cheers,

Tom