Module: **Web Application Development**

Assignment No. = **Group Assignment/Project**

Total Marks: **40**

This is a group assignment where you are asked to work in a group of 6 or 7 members and build a web application collectively. Groups have been assigned on the Loop and details of each group will be added on the Loop page. Even though it is a group project, the grading of the project will be done on the overall project submission (50% marks) which will be equally given to all “active” team members and individual role marks (50% marks) which will be individually graded based on quality of tasks assigned to the individual team member. Please note that all group members should strictly adhere to DCU Academic Integrity policy and students are not allowed to discuss or share their group work with other groups.

Requirements:

In this assignment, you are required to

- Build a complete web application using the technologies learned during this module.
- Your application for the group assignment is a Factory Machinery Status & Repair Tracking System (details are added below)
- You have been already assessed for each of the individual underlying technology, your focus in this assignment/project should be on the overall web application performing a specific task rather than using all technologies within your web application.
- You are allowed to use any IDE and UI Frameworks for development; however, you have to stick to the core techniques covered in this module e.g. you are not allowed to use PHP instead of Python Django.

Minimum Functionalities:

There must be at least following functionalities in your web application.

- Minimum 5 or more static web pages connected through hyperlinks with reasonable contents in each page, with menu bar and navigation.
- Have some client level Javascript functionalities such as events handling, forms validation and dynamic contents creation at the client side.
- Minimum 5 or more dynamic pages/functionalities through server-side programming displaying contents based on the context or user input.
- Have a back-end database running and connected through server-side programming where data can be dynamically queried or updated.
- Have a login functionality showing multiple roles for different kinds of authenticated users.
- The source code should be a Django Python project containing a Dockerfile that can build and run the website. The connection to the SQL database must be configurable at runtime using Docker Environment variables, e.g. using “dj-database-url”). The SQL migrations must be automatically applied when the Docker container starts. Test data can be provided

as part of the “django migrate” SQL migrations or seperately using “django loaddata” fixtures run on startup. A Docker volume can be provided for persistent file storage. The website should become fully operational when started from the Docker container given a fresh empty Docker volume & blank database connection.
- Comprehensive project documentation, including introduction to the application domain, requirements gathering, technology selection decisions, code management decisions, roles assignments and responsibilities, minutes of group meetings, and installation instructions including README file.
- Project demonstration using screencast with audio explanation or at least screenshots of different pages/functionalities with text explanation should be submitted.

Assessment Purpose:

The main purpose of this assignment is to assess your ability to

- Understand the Web Application Development Life-Cycle.
- Understand the requirements for a web application, your potential audience and provide better end-user experience.
- Provide a right balance between web page design, navigation and back-end functionalities.
- Design Front-end of your application using HTML and CSS.
- Use Java Script to provide basic level browser based interaction for validation, visualisation, event handling, dynamic menu bars and animations etc.
- Design back-end logic for server-side programming.
- Design and create a database tier for your web application.
- Store, query and update data web browser into persistent data storage.
- Create and maintain user sessions with your web application and provide log in functionality.
- Demonstrate your ability to work in a team environment and contribute to build large web applications.
- Quality assurance and testing your web application.
- Deploy and maintain your web application.

Web Application (Factory Machinery Status & Repair Tracking System):

For this group assignment, all groups are asked to build the following web application:

ACME Manufacturing Corp. has contracted your software development company to build a website for them that they can use to track the operational status, repair requests and repair history for all factory machinery in their company (*what they manufacture is up to you)*. Fortunately for you, they are a professional engineering company and know what they are looking for in a system, so as part of the procurement contract they have provided the following ideal requirements:

- A piece of machinery can have the status "OK", "Warning", "Fault".
- If a piece of machinery breaks down, "Technicians" should be able to create a "Fault" case against it, indicating that its not working and needs to be repaired, along with notes and images about the fault. Each time a "Fault" is created, it should get its own unique case number where the history of the fault can be recorded and inspected later.
- When "Repair" personnel are working on the machine, they can add new notes and images to the "Fault" case, and if it has been fully repaired, they can mark the Fault as being resolved, returning the machine back to “OK” operational status. “Technicians” and “Managers” should be able to comment on these cases too.

- “Warnings” are free-form text string statuses that can be added to a machine. A machine has a status of “Warning” if any warnings are actively set. A machine can have multiple active “Warnings” (duplicate warning strings can be ignored). Warnings can be added by “Technicians” users (or via the API below) and removed by “Technicians” or “Repair”.
- "Managers" should see a dashboard summarizing all current machine statuses, but also the ability to drill down and view summaries for different collections of machinery.
- "Managers" should be able to generate and export file reports for groups of machinery or individual pieces of machinery.
- "Managers" can assign "Technicians" and "Repair" personnel to specific machines. When "Technicians" and "Repair" accounts view their main dashboard view, they should first see the statuses of the machines they are assigned to, but if they want they can still view the statuses of the other machines.
- More important machinery should appear first when being viewed in lists.
- "Managers" can add new pieces of machinery and delete old pieces of machinery.
- External monitoring systems that the customer already has should be able make a simple JSON-based HTTP POST API request to automatically record "Warnings" or "Faults" for any piece of machinery. (You should provide test forms for this piece of functionality).
- REST APIs should be provided to view current machine status (e.g. this API could be read by external systems to show system status on an operator console or using colored LEDs).
- REST APIs should exist that can list open cases, read the content of a specific case and and add a text entry to a specific case.
- A piece of machinery can be a member of multiple user-definable collections, e.g. *Main-Campus, Building-A, Floor-12, Room-15, Soldering-Machines, Model-53A*. Collections are “flat”, i.e. they only contain machines and no other nested collections. Collection names should match the regex [A-Za-z0-9\-].

However you are free to expand with additional functionality if you wish (e.g. different categories of fault, various fault reporting statistics, indoor floor maps, equipment time-series data, repair time & cost tracking, replacement part order tracking, additional APIs for machine & user management etc) but this additional functionality is not required for full marks.

If you do not have enough time to implement everything, be sure to implement the most important basic features first to provide a basic usable system for the customers required use case.

Data Visualisation/Summarisation/Dashboard: To present the current and historical machine statuses in a way that is easy to understand for the user at a glance, you should create summary visualizations. A few examples for visualisation can be taken from the Javascript Graphics Section on the W3Schools Website accessible at: https://www.w3schools.com/js/js_graphics.asp The most important JavaScript libraries for graphs and visualisation are Plotly.js, Chart.js, D3.js and Google Chart.

User Authentication and User Access Roles: You are expected to implement an authentication mechanism for your group application. The different possible roles of users are “Technician”, “Repair”, “Managers” and “View-only”. Only “Managers” should be able to create or delete accounts.

User Manual: A user guide should be included with the product, detailing what the website does and how it can be used. This will be read by each of the different kinds of users describe above, along with IT staff who will be maintaining the site.

List of Potential Roles for Group Project:

Below is the list of potential roles which members of the group can take on during assignment preparation and submission. These are only tentative roles; you are free to create new roles as well. There is also no restriction to have assigned members for each role or on the number of group members taking the same role, e.g., if you believe frontend development or server-side programming requires more efforts for your application multiple members can take on the same role. You can take your individual role depending on your confidence on one of the technologies or skills. A sample list of roles is as below.

- Project management, business requirements analysis and/or Web application structure design.
- Front-end development HTML and CSS
- Client-side interactive features design (e.g. Java Script)
- Server-side development
- REST API development
- Database design and connectivity
- Testing and quality assurance
- Cloud deployment, maintenance, and scalability
- Code repository management, and installation
- Project documentation including minutes of the meetings, and the final report (each group member should contribute to preparing project documentation explaining their role and approach for the implementation of their assigned tasks.)
- Project demonstration (screencast recording or documentation with screenshots)

Submission Instructions:

1. Please arrange an initial meeting with your group members and make decisions regarding the application structure, roles, and responsibilities. You should submit a short document containing information on the assigned roles of each member by March 19th, 2025. This can be submitted by one member of the group on the behalf of whole team, and it should be submitted in the Group Assignment – Overall Group Marks section of the assignment on module loop page.

2. The following files must be submitted on the loop on or before the submission deadline
   a. Minutes of the meeting arranged at least once in a week (Kick off meeting and one meeting per week during week 9, 10, 11, 12, 13)
   b. Project source code and if available link to the code repository
   c. A Readme file containing any additional installation instructions
   d. Project documentation
   e. A screencast with audio for project demonstration and/or screenshots of web pages with explanation of different navigation steps.

You can submit all files as one zip file or multiple zip files depending on the application size. One person can submit the project for the whole group. It must be submitted in the Group Assignment – Overall Group Marks section of the assignment.

3. Each student must submit the following document in the Group Assignment – Individual Marks section of the assignment.

a. A maximum of one-page description of your role and tasks which you have fulfilled in the group assignment. You can take this text from the project documentation, or you can only include pointers to group submission documentation indicating which part of the project was done by you to highlight your individual contributions. Please keep it to the point and don’t exceed it beyond the one-page limit.