Read the course assignment instructions carefully. If you should not understand anything, please contact one of the teachers. As per the BED program rules, students are not allowed to discuss the course assignment directly with other students. However, questions can be asked in the SP1 class chat, where teachers can comment and give advice as needed.

This is an individual project, not a group project.

The course assignment is graded A-F.

Customer Requirements:

You as a developer have been contacted by WeDeliverTECH™ company. The company specializes in selling technology components and delivering them directly to customers. They would like a web app that has a **Reception Management dashboard** to manage the following:

- Staff member out-of-office logging
- Deliveries tracking

In a meeting with the executives of the company, they presented their needs as follows.

1. They require the ability to **keep track of their staff members** who have clocked out of the office. If a staff member should leave the office for a meeting during the business day, the receptionist must "clock them out" on a dashboard. This dashboard will give the receptionist an overview of who is currently in the office and who is currently out of the office.
   If a staff member needs to leave the office (for example when taking a lunch break), the receptionist needs to "clock them out". She must select the staff member in the table and click the 'Out' button. She must then have a way to enter the length of the staff member's absence in minutes, and the system should calculate and show what the duration of their absence (In hours and minutes if equal or greater than 60minutes) will be in the table, as well as their expected time of return.
   This is how the receptionist knows when they should return to the office, and that they are out of the office. Their status should also change from In to Out.
   Likewise, when they return to the office, and they are "clocked in" their status must change from Out to In. If the staff member has been out-of-office longer than they indicated, a toast notification should appear on the screen with the staff picture, name, surname and amount of time they have been out-of-office for. The receptionist can then decide what the next course of action is to follow up on this. When the staff member returns from their absence, clicking the 'In' button will clear their Out Time, Duration and Expected Return Time cells, and update their status.

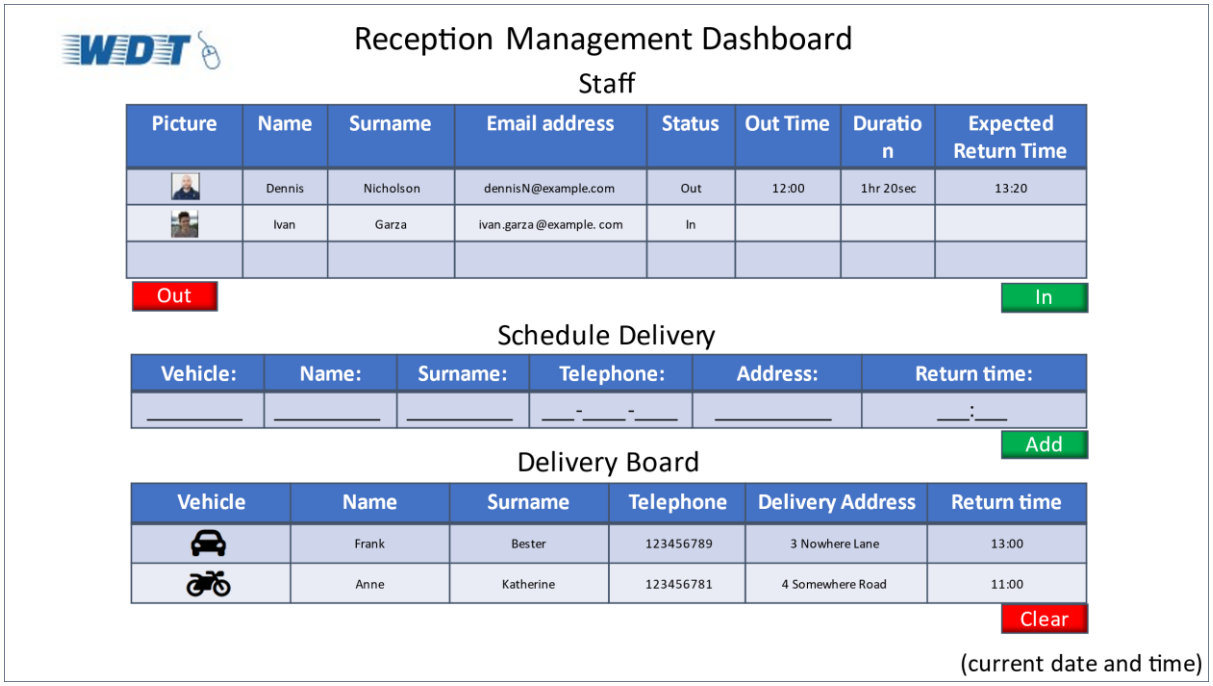
2. In the same web page, they require functionality to **keep track of current deliveries** of orders to customers. To provide the customers with the best service, the receptionist must have information regarding the current deliveries. Therefore, in the dashboard, there needs to be a Schedule Delivery area, where the receptionist manually inputs information about the delivery driver and their current delivery, when they arrive for work in the morning. This is necessary as delivery drivers are employed on a temporary basis, and some might not show up for work every day.

The driver's type of vehicle (Motorcycle or Car), name, surname, telephone number, delivery address and the time that each delivery driver will return to the office needs to be captured so that the receptionist knows who is not available for deliveries at any point in the business day.
Once the required information is captured, clicking the 'Add' button will add the delivery information to the Delivery Board table. The receptionist has asked that vehicle icons be used in the Delivery Board, as it is easier for her to find the correct delivery at a quick glance. WeDeliverTECH™ also requested that if the return time has passed but the delivery driver has not yet returned, a toast notification with the name, surname, telephone, estimated return time, and address of the current delivery must be displayed so that the receptionist can follow up with the delivery driver. When a delivery driver returns to the office, the receptionist can select their row in the delivery board and click the 'Clear' button. This should remove their row from the Delivery Board and delete the object for that delivery driver. There must be a confirmation popup so that the receptionist can't remove a delivery driver by mistake.

3. The Current Date and Time should be clearly displayed at the bottom of the web page, updated every second. This is essentially a digital clock for the Receptionist to refer to. The specific format required for this clock is "Day Month Year Hour:Minute:Second".

4. The executives have indicated that the company brand must be reflected in this system. This should be done by including the WeDeliverTech™ logo on the web page, and only using the specified fonts and colours included in the Company Branding Profile, for the web elements.

WeDeliverTECH™ has provided us with a basic mock-up of the layout they would like the reception management dashboard to follow:



Dashboard Layout

Your task as the developer is to ensure that the correct functionality is present, according to the customer requirements stipulated by the executives of the WeDeliverTECH™.

Project specifications

The WeDeliverTECH™ executives have also indicated the following:

Their internal IT department is going to maintain the web application once it has been delivered. The head of the IT department requested that **Bootstrap with JavaScript/jQuery** and OOP (Object Oriented Programming) concepts should be used for the web application.

As the IT department does not have any knowledge of NodeJS implementation, they have requested that **NodeJS should not be implemented** for the project.

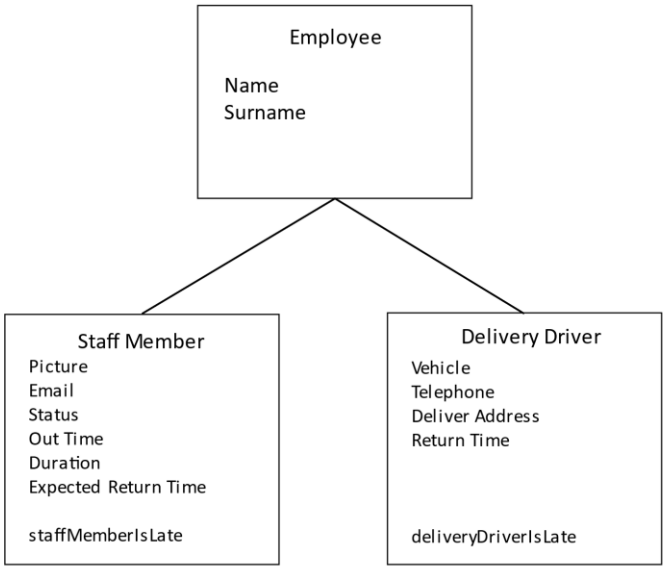
Any external libraries or plugins may be used for the project, but the minimum requirements (Bootstrap, JavaScript/jQuery and OOP concepts) must be met. Remember to include instructions in your read.me, regarding the use of any external libraries or plugins.

All JavaScript functionality described in this web application's requirements, need to be in a JS file in the correct folder. This file needs to be name **wdt_app.js**

You should pay attention to the user experience, therefore when buttons on the Dashboard are hovered over with a mouse cursor, this should be indicated with some animation on that button. You may add animations to the web application where you think it may enhance user experience.

To protect their staff members' identities, the company has requested that you use the API (Application Programming Interface) https://randomuser.me/ to get the demo information for 5 staff members.

This diagram depicting the required types of objects with methods or properties has been provided by the IT Department:



This Staff table should then be filled with the newly created objects (created according to the diagram) when the web page is initially loaded.
The 5 staff members in the table should be unique (i.e., 5 different staff members).

The Delivery Driver information will be manually filled in by the receptionist, and all user input data in this system should be validated to make sure it is in the correct format. The minimum validation should be that all fields are filled in, and that the time format is correct (hh:mm).

This input data should be used to create the relevant Object, and the Delivery Driver table should then be filled with this object's data.

There has also been a request to create and maintain a project plan using the Jira software. Sprints, Epics, and Issues must be present, along with a Board that is kept up to date in terms of task progress and task assignments (which team member is assigned to each task). The timeline for this project should match the amount of time students have been given to complete this Semester Project (I.e., total project time is 4 weeks for full-time students, and 8 weeks for part-time students).

Although this is just the first phase, the executives are planning on including more functionality in the coming phases of this project. Because of this, they have requested that the web application has a navigation bar at the top of the screen with the following menu options:

Dashboard	Inventory	Orders
	Search	Search
	Add	Add
	Remove	Remove

The *Inventory* and *Orders* sub-menu items, which should be hidden unless the navbar is interacted with, should not be operational, meaning nothing should happen when you click on them.

The ‘Dashboard’ navigation item should be active by default. Clicking on this navigation item will navigate to the main Reception Management Dashboard page with the tables for staff and delivery.

WeDeliverTech™ Company Branding Profile:

Logo:

Element	Style
Web page text elements (paragraphs, etc)	Font: Calibri Font colour: #212529
Navbar	Font: Consolas
Navbar Menu Item	Background colour: #0E8EA8 Font colour: #fff
Navbar Sub-menu Item	Background colour: #83D1E1 Font colour: #212529
Table	Border colour: #212529 Border corners: rounded Border thickness: thin lines
Table Header	Font: Consolas Background colour: #0E8EA8 Font colour: #fff
Table Cell	Font: Calibri Background colour: #83D1E1 Font colour: #212529
Positive button Confirm E.g., Confirm, Next, Add, Yes	Font: Consolas Background colour: #198754 Font colour: #fff

Negative button Cancel E.g., Cancel, Previous, Delete, Remove	Font: Consolas Background colour: #dc3545 Font colour: #fff

Jira Project Management

You must create a **"Reflection Report"**, in the form of a PDF document named **"Jira.pdf"**
This report must include the following:

- Screenshots of project Sprints, Epics, and Issues, elaborating on why you chose to create these issues.
- Screenshots of project Roadmap, Board and Backlog, and elaborations on the progress of the project
- A summary of the management of this project, explaining why some choices were made in the development of this web application. This section can discuss some of the more difficult tasks, and how the results were achieved. The **reflection report section must be between 500-1000 words** and can include additional screenshots of what is being addressed.

Git

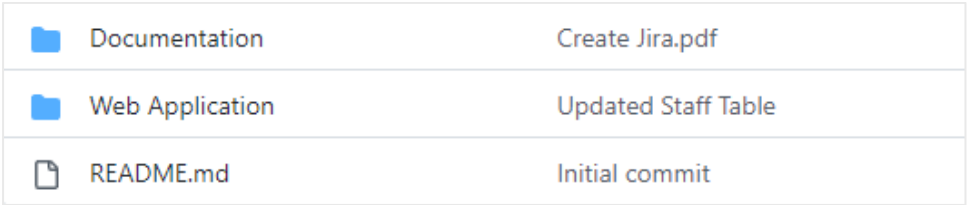
For the delivery of this project, you are to create a **new private** GIT repository.
The repository must be named "fname_lname_sp1".
(Replace 'fname' with your First Name, and 'lname' with your Last Name).

This repository must have **two folders** in it:

- A "Web Application" folder for all the separate project code folders.
- A "Documentation" folder for the project documentation. (This folder is where your "Jira.pdf" file will be placed)

There must also be a ***"readme.md"*** file in your repository (Outside of the previously mentioned folders) which documents the location of the code files and includes **complete instructions on how to install and run your web application,** including any external libraries or plugins.

Example of private repository folders



Make sure to give 'JoshNoroff' and 'FJ-teach' access to your repository in GitHub.
This access ensures that your code can be cloned to be marked. Failure to be given access to download the code/web app will result in a failed grade.

How to give access to your private repository:
1. On GitHub.com, navigate to the main page of your private repository.
2. Under your repository name, click the “Settings” tab.
3. In the "Access" section of the left sidebar menu, click “Collaborators”.
4. In the “Manage Access” section of the page, click “Add people”.
5. In the search field, type the account name you wish to give access to.
6. Select the name in the list of matches
7. Click “Add x to this repository”
8. Do this for both “JoshNoroff” and “FJ-teach”.

Project Delivery

Finally, the **link to your repository needs to be submitted on Moodle, in a .txt file** named the following: “FName_LName_SP_AUGFT22.txt”.

(Replace ‘FName’ with your First Name, and ‘LName’ with your Last Name).

NOTE: This .txt file is the ONLY submission for this Course Assignment (In this .txt file, you must add your repository link to which you have committed your web application code). The .txt file needs to be submitted before the closing date. ***Any GIT commits after the submission closing date will not be graded.***

Grading Criteria

Reflection Report PDF must include Jira screenshots of appropriate Sprints, Epics, and Issues, with explanations as to why these were created. Elaboration on the management of this project, progression tracking of issues and solutions to challenges in creating the web application. This should be accompanied by screenshots of the Roadmap, Board and Backlog. Word count is between 500-1000 words.

System styling adheres to WeDeliverTech™ Company Branding Profile
(All the styling should be in a single, separate CSS file).

Correct API call/s made on page load.
The API JSON String response should be converted into a JS object/s, which should then be used to create relevant class objects.
Inheritance is used in object creation, using the data from the API call.
Staff table populated with the objects of the 5 unique staff members.
(There must be a staffUserGet function that makes the API call(s) and processes the response(s), I.e., converts the API response(s) to the relevant JS class object(s)).

Clicking ‘Out’ prompts the user for data, updates the relevant staff member’s object, and then updates the Staff table.
(There must be a staffOut function)

Clicking ‘In’ updates the relevant staff member’s object and updates the Staff table.
(There must be a staffIn function)

Toast should be shown, with the correct information, when a staff member has not returned by the expected return time.
(There must be a staffMemberIsLate function)

Delivery Driver information is manually entered into input elements in the Delivery Driver table. The table is populated with the Delivery Driver object data.
(There must be an addDelivery function which adds the delivery driver’s information to the Delivery Board table)

Delivery Driver input is validated (Checked for correct format).
(There must be a validateDelivery function)

Inheritance is used in the Delivery Driver object creation.

Appropriate icons are used for Vehicle types, not images.

Toast should be shown, with the correct information, when a delivery driver has not returned by the estimated return time.
(There must be a deliveryDriverIsLate function)

The current Date and Time should be updated every second (basically a digital clock), in the specified format (Day, Month, Year, Hour:Minute:Second”). Example 5 June 2022 14:54
(There must be a digitalClock function)

At least the minimum hover animations are present, as requested.

Marks and grade allocation work as follows:

Your submission will make up your final grade.
Course Completion counts for 0% but it is a requirement to complete, to receive your grade.

%	Grade
90-100	A
80-89	B
60-79	C
50-59	D
40-49	E
0-39	F – Course Assignment Failed. Must do a Resit.