

# MCDA5550 – PWA Assignment

## Overview

Your task is to build a Progressive Web App (PWA) that combines task management with motivational content. The app should display a **random motivational quote**, and include a **to-do** list with deadlines, while working reliably offline properly.

## Submission Instructions

1. Create a GitHub repository for the project.
2. If the repository is private, grant access to [A00446838](#) and [simonachkar](#).
3. Submit the repository link on Brightspace.

## Application Features

### Motivational Quote




- Display a random motivational quote when the app loads
- Quotes may be:
  - Hardcoded
  - Loaded from a local JSON file
  - Fetched from an API (must work offline using caching)

### To-Do List with Deadlines

Users must be able to:

- Add a task with: Task description & Deadline date
- View all tasks
- Delete tasks (or mark them as completed)

Each task must visually reflect its deadline status:

-  Yellow: Deadline is approaching (in a week)
-  Orange: Deadline is very close (in less than 3 days)
-  Red: Deadline has passed

# Technical Requirements

## PWA Implementation

- Build a fully functioning PWA using either plain JavaScript, Vite, Create React App, or a framework of your choice.
- Include a **manifest** file with appropriate icons, a background color, and a start URL to ensure full PWA compatibility.
- Implement a **service worker** to enable **offline capabilities** and resource caching.

## Data Storage

- Use **IndexedDB** to store and manage tasks
- Tasks must persist across:
  - Page reloads
  - Browser restarts
  - Offline usage

## Offline Functionality

- The app should function offline using service workers and IndexedDB
- The app must **clearly indicate when it is offline**

## User Interface

- UI must be functional and responsive
- Clear visual distinction for:
  - Task deadlines
  - Offline status
  - Design does not need to be fancy, but it must be usable and responsive

## Documentation

- Include a README.md file with setup and usage instructions, documenting any assumptions or external resources used.

# Notes

- **Focus on Functionality!** While a basic level of UI design is required, the primary focus should be on the functionality and technical implementation of the PWA features.
- Feel free to use assets and code snippets provided in the lecture's **repository**. This can help accelerate your development process.
- To inspect and debug your app, right-click on the page, select 'Inspect', and navigate to the "Application" tab in **Chrome DevTools**. This is important for reviewing service workers, IndexedDB, and manifest implementation.
- During development and testing, regularly clear your browser's storage to ensure you're seeing the most recent updates. This can be done through the '**Clear Storage**' section in the 'Application' tab of Chrome DevTools.
- Make regular commits to your GitHub repository to track changes and facilitate version control. This also ensures your work is consistently backed up.

## Additional Code

To assist with the learning curve of IndexedDB and Dexie, here is an example function for removing a task from the database:

```
// Removes a task record from the database by id.
function removeTask(id) {
  return db.tasks
    .delete(id)
    .then(() => {
      console.log(`Task with id ${id} was removed`);
      return true;
    })
    .catch((err) => {
      console.error("Failed to remove task: ", err);
      return false;
    });
}
```

# Rubric (10 pts)

- **PWA Implementation (3 pts)**
  - App is functional
  - Manifest file is included
  - Service workers are correctly implemented
  - The app should function offline using service workers and IndexedDB
- **User Interface (3 pts)**
  - UI is functional and responsive
  - Include capabilities to **add**, **view**, and **delete** tasks
  - Deadline Indicators displayed correctly
- **Offline Indicator (1 pt)**
  - Clearly show in the UI when the app is offline
- **Data Storage (2.5 pts)**
  - Data is successfully stored and managed within IndexedDB or localStorage.
  - Ensure tasks can be **added**, **retrieved**, and **deleted**.
- **Documentation & Code Quality (0.5 pt)**
  - README.md file, instructions...
  - Code is well-organized, properly structured, and commented

Quote: "Believe you can and you're halfway there."  
– Theodore Roosevelt

### Add New Task

Task:

Deadline:



Add Task

### To-Do List

Finish the report (Due: Tomorrow)

Delete

Buy groceries (Due: June 5)

Delete

Schedule a meeting (Due: Today)

Delete

Submit assignment (Due: May 20)

Delete



Online



Offline Mode