Day 5

Today I focussed primarily on trying to finish the front end of the application.
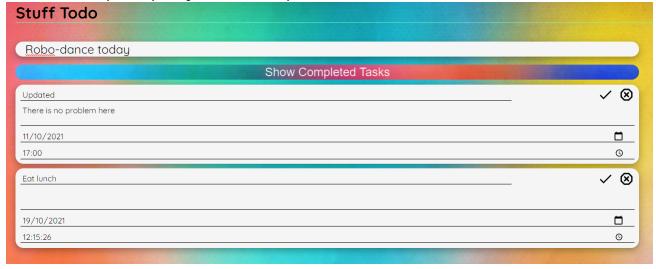
- I implemented the show hide functionality for completed tasks. this is achieved by sending a boolean through to the API. if the boolean is false, meaning hide, then the API will filter out the completed tasks from the results before returning them to the front end

```java
public Collection<TaskDto> getTasks(UUID taskListId, boolean showCompleted) {
    Collection<TaskDto> tasks = tasksRepository.getTasks(taskListId);
    return showCompleted ? tasks : tasks.stream().filter(taskDto -> taskDto.isHidden() == false).collect(Collectors.toList());
}
```

- I found a bug in my code when storing the expiry and refresh expiry tokens. I was assuming that the "expires_in" value represented minutes, but it actually represents seconds. So the session was expiring long before my code thought it did!

```javascript
this.http.post<AuthResponse>(environment.KeycloakAdminBaseUrl + this.authEndpoint, body.toString(), options).subscribe(
    response => {
        sessionStorage.setItem("access_token", response.access_token);
        sessionStorage.setItem("refresh_token", response.refresh_token);
        sessionStorage.setItem("token_expiry", moment(new Date()).add(response.expires_in, 's').toISOString());
        sessionStorage.setItem("refresh_token_expiry", moment(new Date()).add(response.refresh_expires_in, 's').toISOString());
        resolve();
    },
    error => {
        console.error(error);
        reject();
    }
);
});
```

- I also implemented a simple Natural Language recognition system. If the either "today" or "tomorrow", or "mon|tues|wed|thu|fri|sat|sun" are present in the title for the new note, they are parsed out. the word is removed from the title and the relevant date is supplied for the due date. the weekdays work by finding the next suitable day of the week

```
})
export class NaturalLanguageService {

  private static weekdaysRegEx = "mon|tue|wed|thu|fri|sat|sun";
  private weekDays = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'];
  constructor() { }

  parseTitle(title: string) {
    let due: Date;
    if (title.includes('today')) {
      title = title.replace('today', '').trim();
      due = new Date();
    } else if (title.includes('tomorrow')) {
      title = title.replace('tomorrow', '').trim();
      due = new Date();
      due.setDate(due.getDate() + 1);
    } else if (new RegExp(NaturalLanguageService.weekdaysRegEx).test(title.toLowerCase())) {
      const result = new RegExp(NaturalLanguageService.weekdaysRegEx).exec(title.toLowerCase())[0];
      due = this.getNextDayOfWeek(new Date(), this.getDayNumber(result));
      title = this.removeDayTextFromTitle(title, result).trim();
    }
}
```

- I also updated the styling of the application. I made a few general updates to improve the appearance, but I also made some important function changes. I added the due date and time for the task to the view and I implemented a very crude menu system, which shows in place of the sidebar when the screen is too small for the sidebar