IBU | International Burch University

CEN 308 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

To-Do List

Prepared by:

Sead Gačanović

Nermin Kapetanović

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

23. 01. 2022.

Contents

1. Introduction

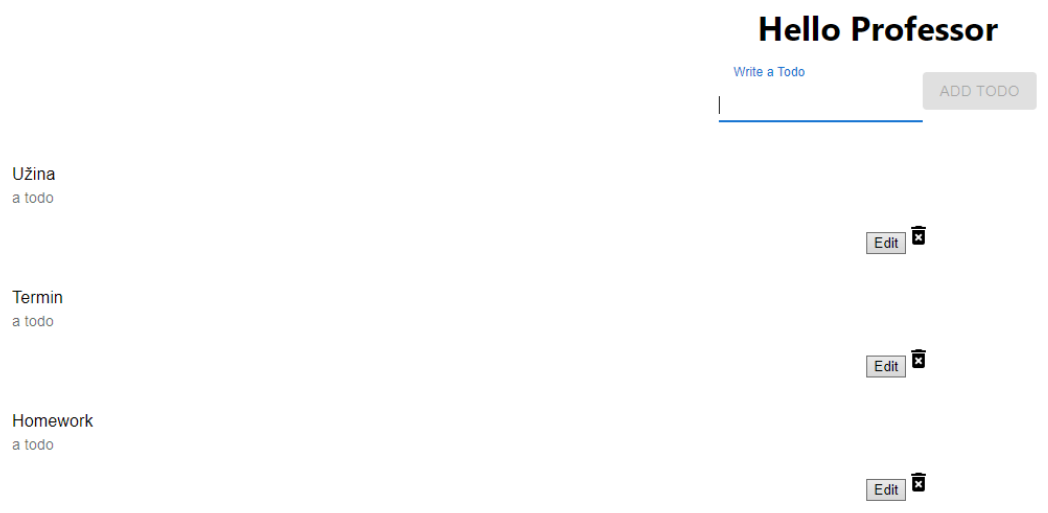
1.1. About the Project

For our project we decided to develop a To-Do List application. A user visits the page and has the ability to add, remove and update items from the list.

The project is hosted on Firebase, on this link: https://todo-app-cp-98935.firebaseapp.com

1.2. Project Functionalities and Screenshots

The main features of the application are: creating, getting, updating and deleting To-Do"s.


Hello Professor

Write a Todo

ADD TODO

Užina
a todo

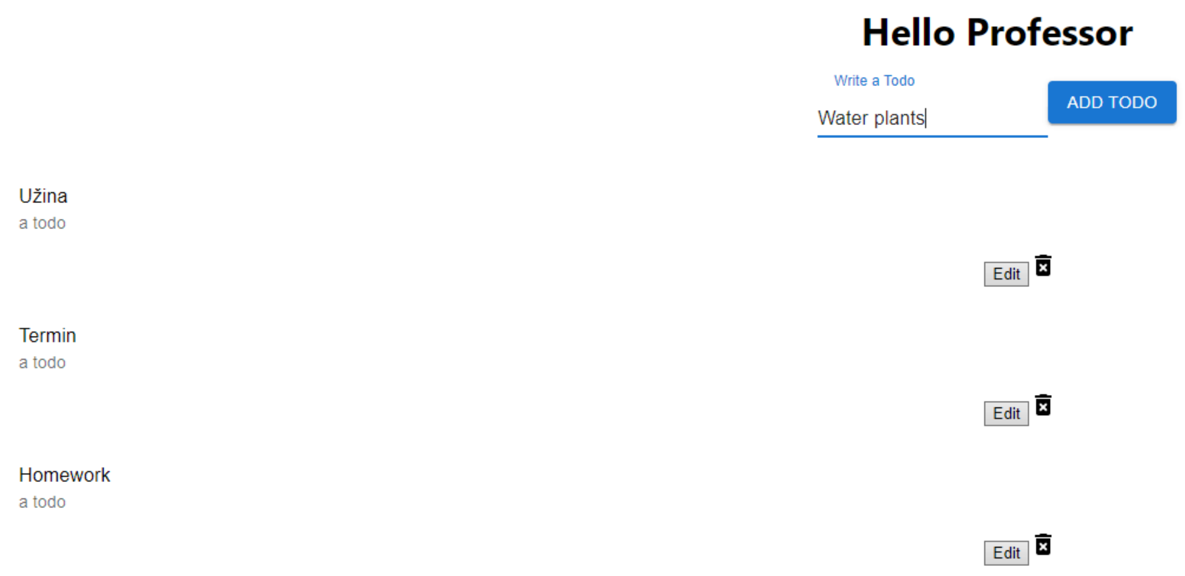
Edit

Termin
a todo

Edit

Homework
a todo

Edit



Hello Professor

Write a Todo

Water plants

ADD TODO

Užina

a todo

Edit

Termin

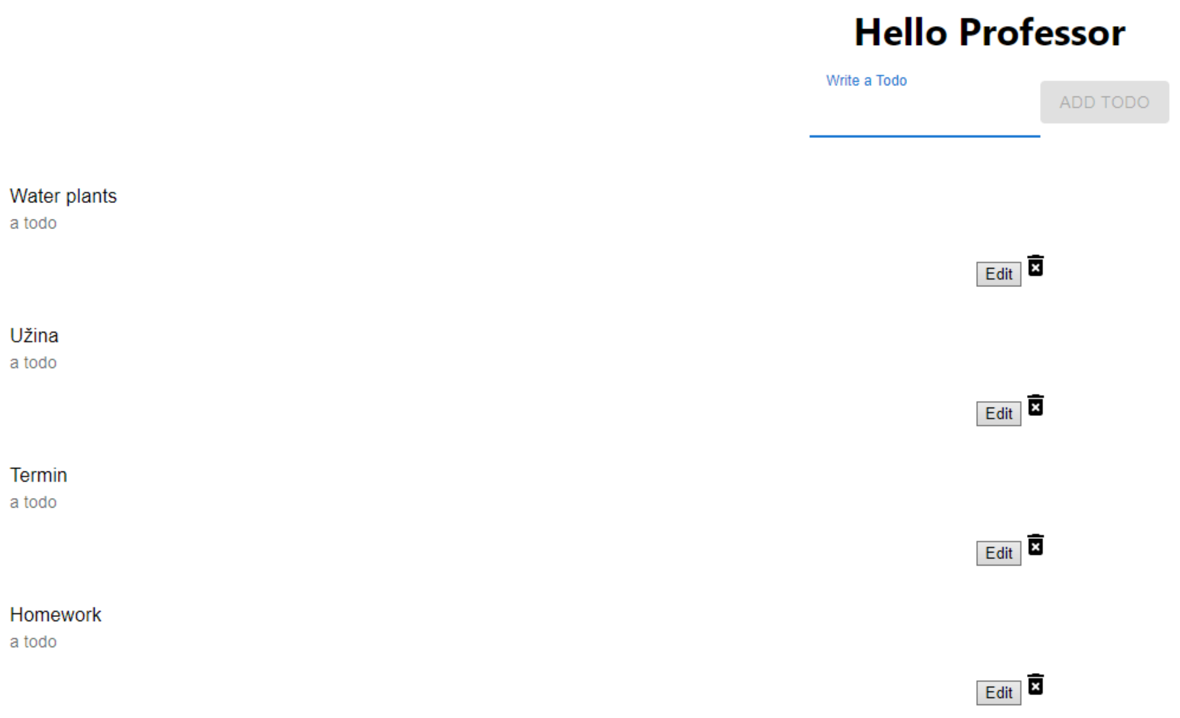
a todo

Edit

Homework

a todo

Edit

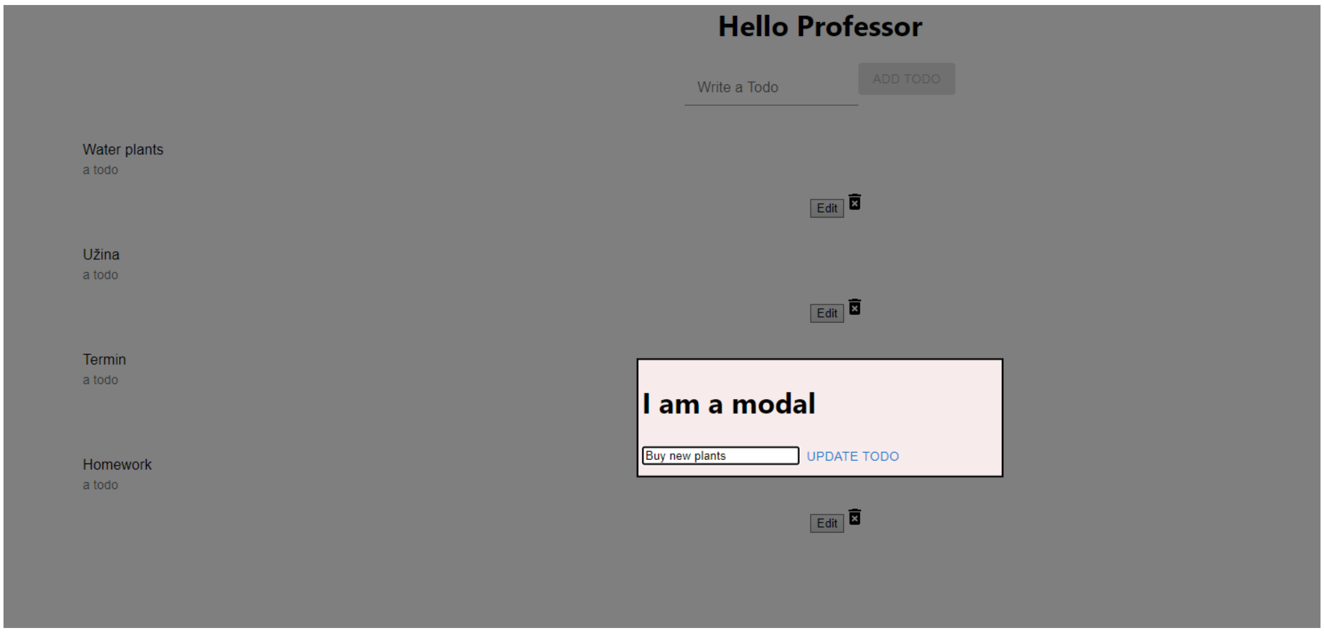


Water plants
a todo

Užina
a todo

Termin
a todo

Homework
a todo



Hello Professor

Write a Todo

Water plants
a todo

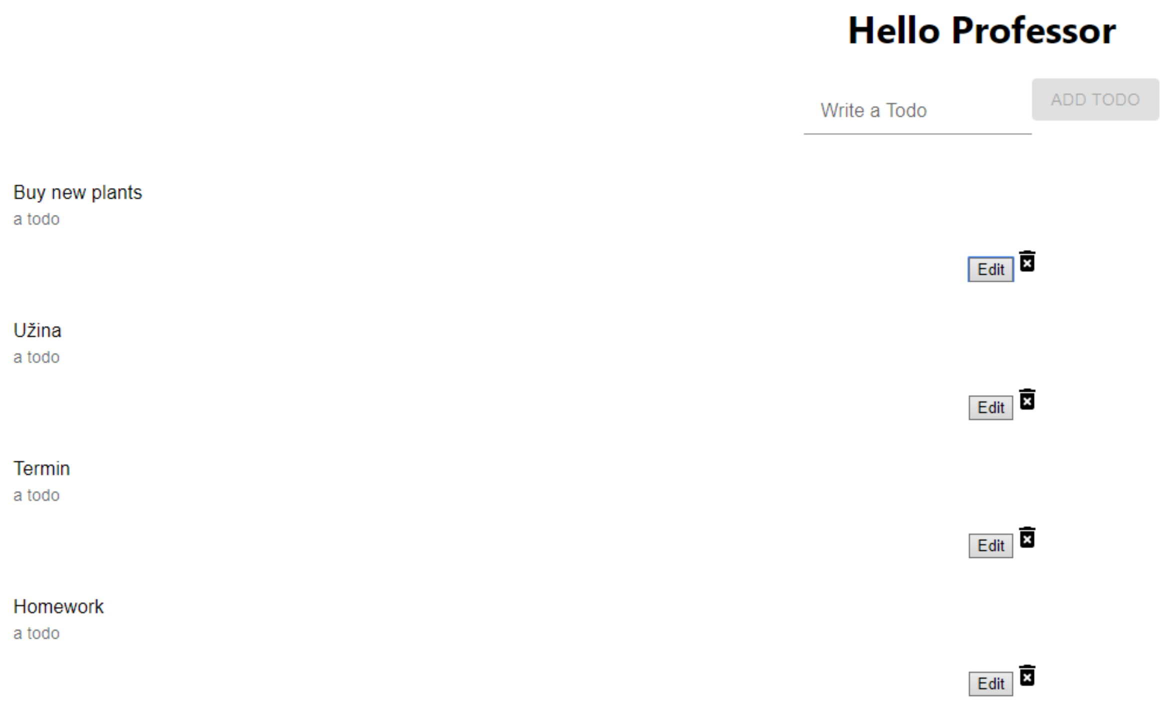
Užina
a todo

Termin
a todo

I am a modal

Buy new plants

Homework
a todo

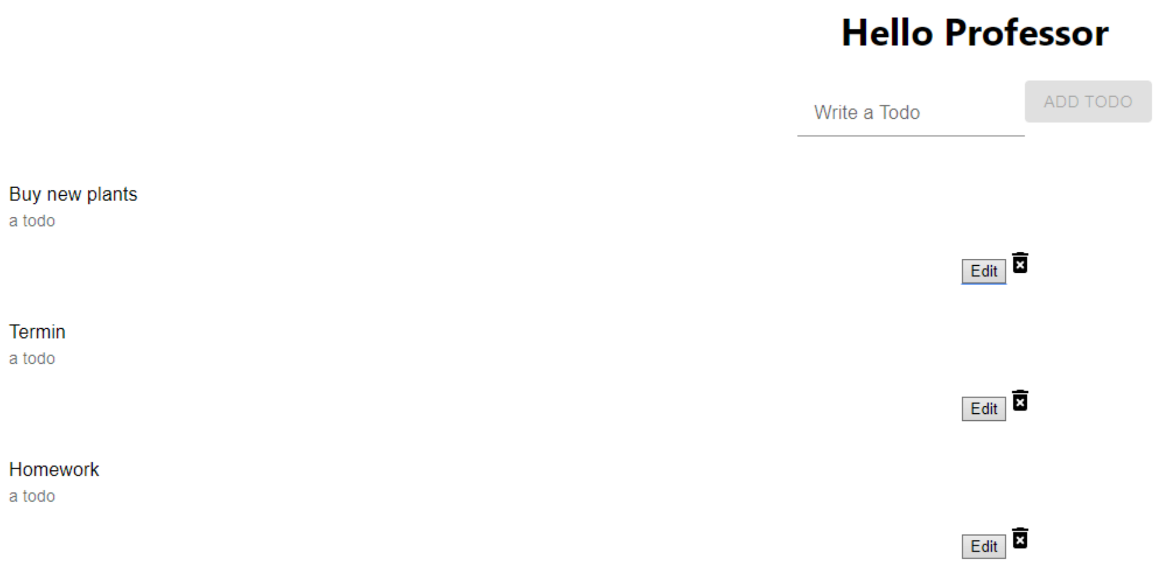

Hello Professor

Buy new plants
a todo

Užina
a todo

Termin
a todo

Homework
a todo



2. Project Structure

2.1. Technologies

We used Firebase for our backend and database. We thought that it would be interesting to use their database since it's a real-time database and it was really easy to set it up. Our app is also a live app and is hosted on Firebase at the following link: https://todo-app-cp-98935.firebaseapp.com . We have one table, or should I say collection, since Firebase only works with collections.

Regarding the frontend, we used React. We have just started working with React and Firebase, so it was a little bit challenging. Also, we used Material-UI for some help with the design. Despite the app does not have so many features, it is still responsive.

Besides following basic coding standards and guidelines, we also followed React coding standards on the frontend. For example, React UI component's names should be PascalCase, while all other helper files (non-component files) should be camelCase.

2.2. Database Entities

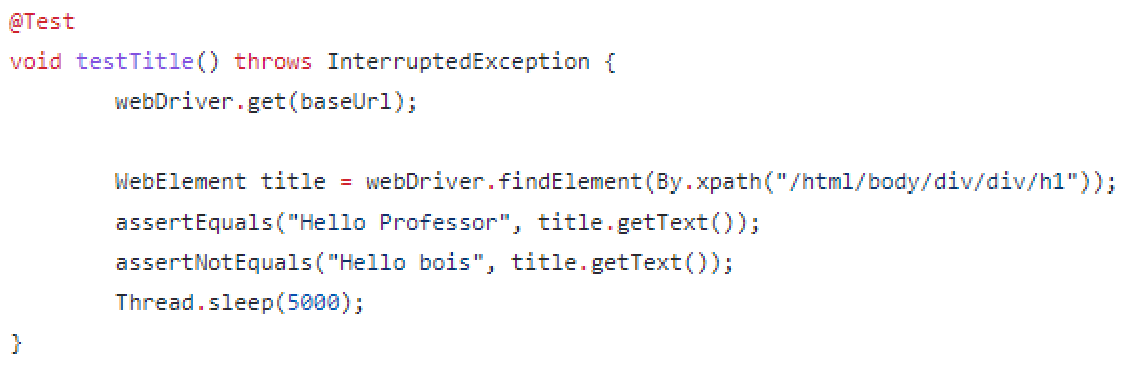
As we said, we have one table/collection, which is named „todos" and it"s fields should have the following attributes: timestamp (stores the server time the to-do was added) and todo (stores the text of the to-do).

2.3. Patterns

Unfortunately, we did not manage to implement any architectural or design patterns in our project

2.4. Testing

We first find the title text by xpath, and compare that value to “Hello Professor”, this is a positive test. We then check that the value is not equal to “Hello bois”, this is a negative test.



```java
@Test
void testTitle() throws InterruptedException {
        webDriver.get(baseUrl);

        WebElement title = webDriver.findElement(By.xpath("/html/body/div/div/h1"));
        assertEquals("Hello Professor", title.getText());
        assertNotEquals("Hello bois", title.getText());
        Thread.sleep(5000);
}
```

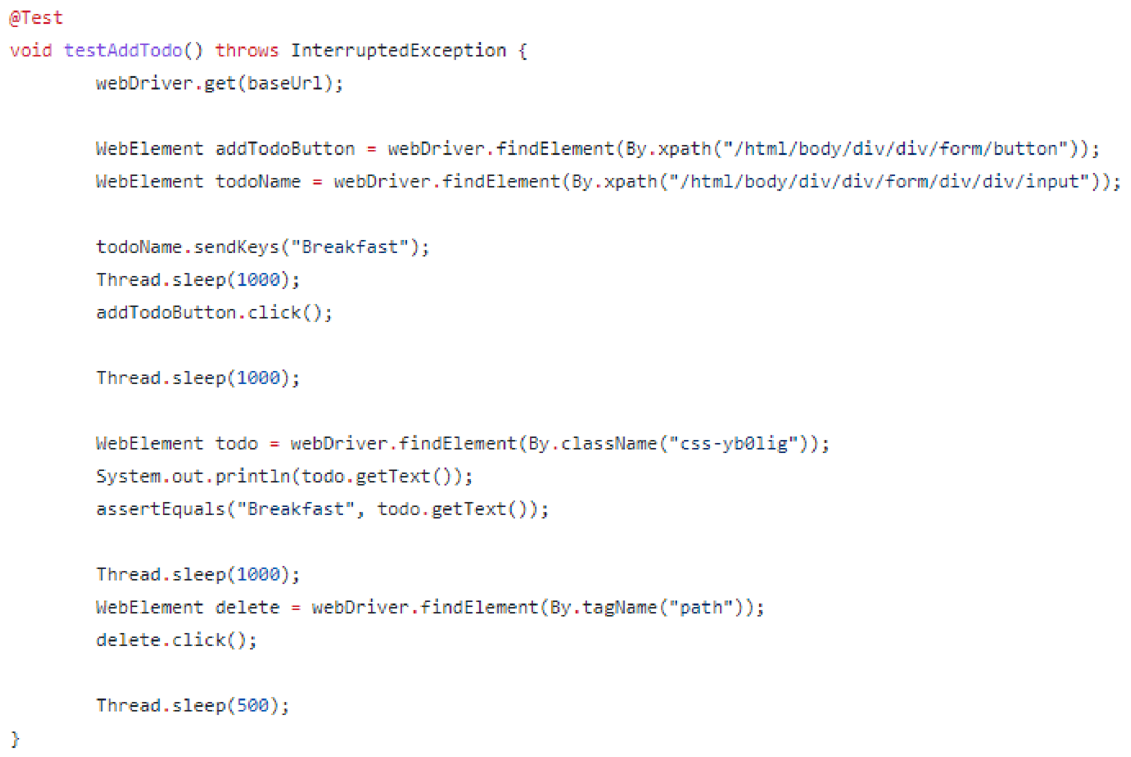
We first, by xpath, connect the todo button and the textbox, which is related to the todo button.

We then use sendKeys method to send the value “Breakfast” to the textbox and then we use the click

Method to click on the addTodoButton and make a “Breakfast” todo.

By className we then find our todo element, and get the text value, we then assert that text value is equal to “Breakfast”.

By tagName we find the delete button and use it to delete recently added todo “Breakfast”.



```
@Test
void testAddTodo() throws InterruptedException {
        webDriver.get(baseUrl);

        WebElement addTodoButton = webDriver.findElement(By.xpath("/html/body/div/div/form/button"));
        WebElement todoName = webDriver.findElement(By.xpath("/html/body/div/div/form/div/div/input"));

        todoName.sendKeys("Breakfast");
        Thread.sleep(1000);
        addTodoButton.click();

        Thread.sleep(1000);

        WebElement todo = webDriver.findElement(By.className("css-yb0lig"));
        System.out.println(todo.getText());
        assertEquals("Breakfast", todo.getText());

        Thread.sleep(1000);
        WebElement delete = webDriver.findElement(By.tagName("path"));
        delete.click();

        Thread.sleep(500);
}
```

3. Conclusion

We think the app looks and feels nice and serves its purpose, but there definitely is room for improvement and we have some ideas on what should be improved.

For example, having users register an account so that all to-do’s are assigned to their specific users, etc.

We struggled with finding a way to implement the design and architectural patterns, but we tend to believe that we will be able to implement them in our future projects.