Contents

[CRUD Project: Grade Management system 1](#_Toc140705069)

[Main Content: 1](#_Toc140705070)

[Navbar 1](#_Toc140705071)

[List Section 2](#_Toc140705072)

[Details Section 2](#_Toc140705073)

[List and Details Interactions 3](#_Toc140705074)

[Footer 3](#_Toc140705075)

[Personal Information 3](#_Toc140705076)

[Full name: 3](#_Toc140705077)

[Faculty number at TU-Sofia: 3](#_Toc140705078)

[Email: 3](#_Toc140705079)

[More information: 3](#_Toc140705080)

[Examples: 3](#_Toc140705081)

# CRUD Project: Grade Management system

This is a grade management system built with React and typescript. The project is a simple SPA (single page application), that enables users to perform basic CRUD (create, read, update) operations, like see the grades in list, save new grade to the list, update an old one just by clicking over it and changing the necessary information, delete a grade from the list.

# Main Content:

The page content is divided by 4 sections:

* Navbar
* List Section
* Details Section
* Footer

## Navbar

The navbar hold the name of the site and by hovering over it, makes it shine. But also it contains a GitHub link to my personal repository and again by hovering over it, makes the icon move to improve the user experience and make the site more interactive.

## List Section

This is one of the main sections in the page, it is responsible with listing all Grades, ordered by ID. Every Grade is a List item <li> in a unordered list <ul>. The properties in the interface are shown below and it is located in the src folder of the project with name interfaces.ts:  
A computer screen shot of a program code

Description automatically generated

Every list item has a delete button, that deletes the selected list Item. Once deleted, all ID’s are updated to go in order, like 1, 2, 3, 4 – even if we delete item with id: 3, after the deletion the ids are updated to an order of 1, 2, 3. This section is the only one, that is scrollable to be able to display all grades, without the page becoming to large to scroll down.

**IMPORTANT!**

The GradeList.tsx is located in the src folder, inside another folder “components”, and it is added to the main file App.tsx with an import and then used in a <div> like a actual HTML tag with its properties. The styling is done with an external CSS by importing it with import in the GradeList.tsx. The styling of the position of the elements is done in the App.css file imported in the main file App.tsx in the src folder with the help of Flexbox and for that reason in the project are added more <div>s to accomplish the task.

## Details Section

The Details section holds the form, that you is used to create new Grades. The GradeDetails.tsx is located in the src folder, inside another folder “components”, and it is added to the main file App.tsx with an import and then used in a <div> like a actual HTML tag with its properties. All fields specified in the interface (except for the field ID) are available on the form to be filled. All input fields are mandatory, so you are not able to submit the form if one of the input fields are empty, even the date requirement. It has a Save button, that submits the form and then the information is displayed in the List section with the right order, because the functions remember and then update the ID by adding or deleting a grade. It also has a clear button, that clears out the form, once saved the information, but also the fields are cleared, after pressing the save button for better user experience. The page loads with an empty fields, that ready to be filled and saved. The styling is done with an external CSS by importing it with import in the GradeDetails.tsx, which CSS file is located in the same folder. The styling of the position of the elements is done in the App.css file imported in the main file App.tsx in the src folder with the help of Flexbox and for that reason in the project are added more <div>s to accomplish the task.

## List and Details Interactions

When clicking on any list item, its information appears on the details form. Users can edit the fields and upon clicking save, the list item is updated with the new information, displayed in the List section. If the list item is clicked, and then deleted, the details section form is cleared. If the saving part is done, the fields are automatically cleared for the user. The page loads with empty fields, so that the user can add a grade immediately. When all of the grades are deleted, the fields of the form are also cleared. When adding a new Grade in the details form and clicking on save, the new Grade is added as the last item on the list with a proper ID.

## Footer

The footer isn’t containing any relevant logic. It is positioned with a help of a external CSS with name App.css, which is imported into the main file App.tsx, using Flexbox and a couple of other CSS properties to make it better looking. It contains only a Copy Rights texts and these have an animation, that is displayed, when hover over them to make more interactive and fun.

# Personal Information

## Full name:

Gabriela Zhivkova Stoyanova

## Faculty number at TU-Sofia:

121221029

## Email:

gabigs2002@gmail.com

gstoyanova@tu-sofia.bg (TU-Sofia work email)

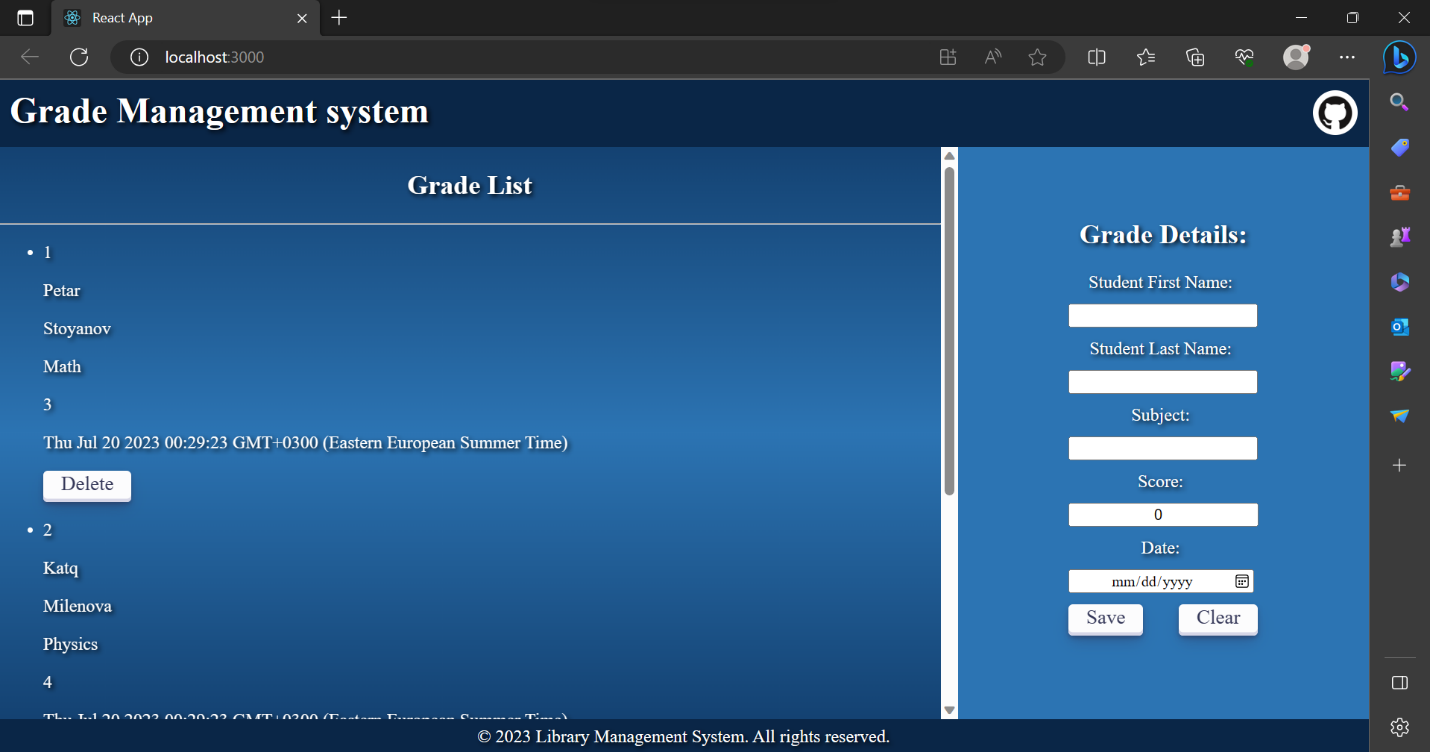
## More information:

GitHub: <https://github.com/GabrielaStoyanova> (included on the page as well)

Specialty in TU-Sofia: Computer and Software Engineering (CSE)

Group: 44 Stream: 9

# Design of the page and examples:

The design:

Examples:

(Click over it to watch a review of the site)

