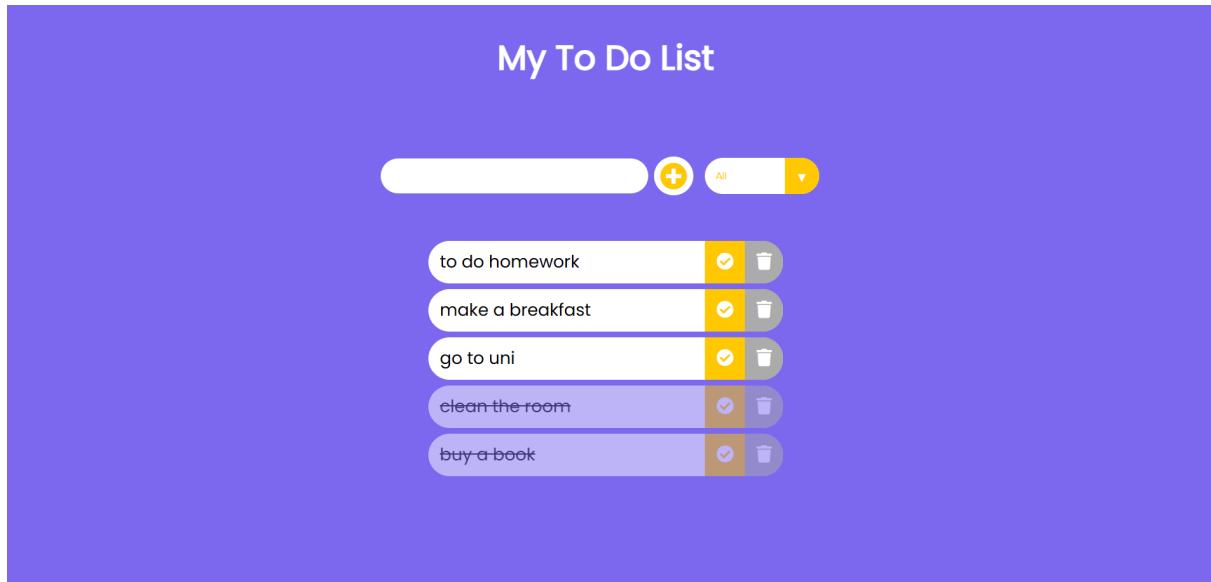
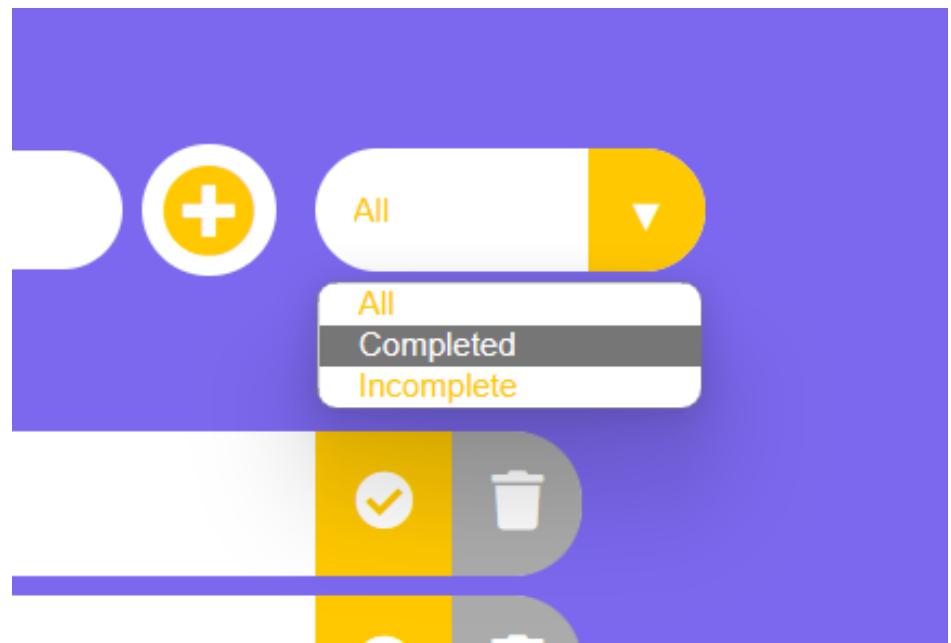


Aiperi Zhenishova COM22B



This is my design of the project “ToDo List”. Here you can see the main page with buttons, with which you can add tasks, delete them.

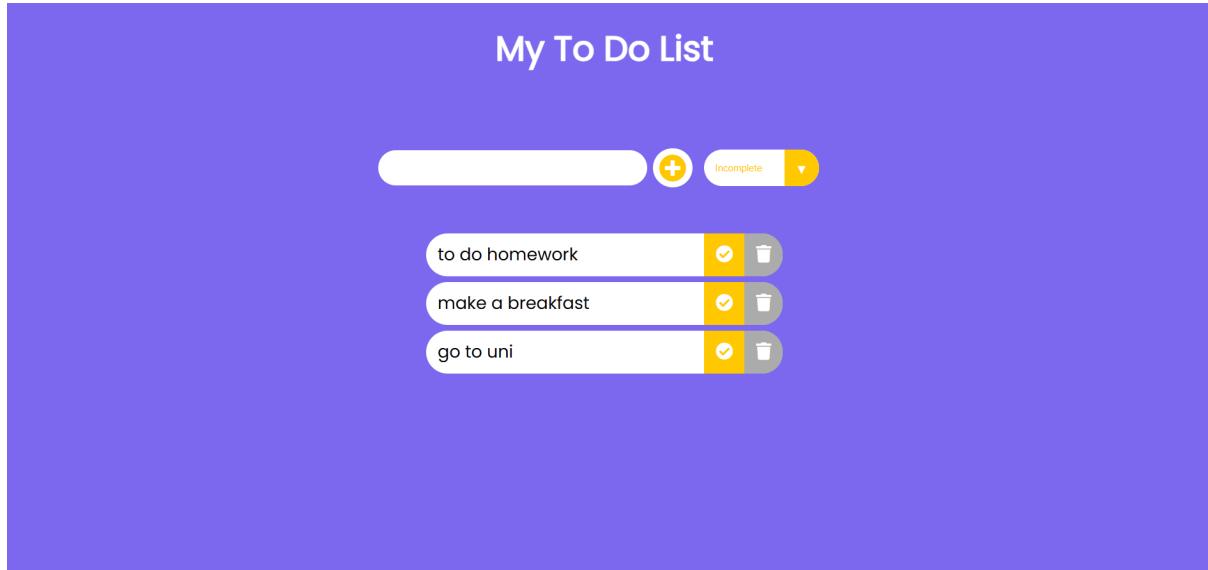


## My To Do List

A screenshot of a to-do list application titled "My To Do List". The interface includes a navigation bar at the top with a purple header, featuring a white plus sign (+) icon and a "Completed" filter button. Below the header, the main content area displays two items in a list:

- clean the room
- buy a book

Each item has a small grey box to its right containing a white checkmark icon and a white trash can icon. The entire application has a clean, modern design with a white background and a purple header.



On the ride side are buttons “All”, “Completed”, “Incomplete” . When you click “All” it is possible to see all your tasks. When you click to “Completed” there appears completed tasks and the “Incomplete” button shows not completed tasks.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
    <link
    href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
    rel="stylesheet">

    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
    integrity="sha512-iBBXm8fW90+nuLcSKlbmrPcLa0OT92xO1BIzZ+ywDWZCvqsWgcccV3gFoRBv0z+8dLJgyAHIhR35VZc2oM/gI1w=="
    crossorigin="anonymous"
    referrerPolicy="no-referrer" />
    <link rel="stylesheet" href="style.css">
```

```

<title>To Do List</title>
</head>
<body>

<header>
    <h1>My To Do List</h1>
</header>
<form action="">
    <input type="text" class="todo-input">
    <button class="todo-button" type="submit">
        <i class="fas fa-plus-circle fa-lg"></i>
    </button>
    <div class="select">
        <select name="todos" class="filter-todo">
            <option value="all">All</option>
            <option value="completed">Completed</option>
            <option value="incomplete">Incomplete</option>
        </select>
    </div>
</form>

<div class="todo-container">
    <ul class="todo-list"></ul>
</div>

<script src="script.js"></script>

</body>
</html>

```

This is HTML code where I add google fonts, css and Javascript. Here I write all necessary things that have to be on screen.

The application uses the following external libraries:

- Font Awesome - Icons for a visually appealing interface.
- Google Fonts - Poppins - Stylish font for improved readability.

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

```

```
}
```

```
body {
```

```
    background-color: mediumslateblue;
```

```
    color: white;
```

```
    font-family: "Poppins", sans-serif;
```

```
    min-height: 100vh;
```

```
}
```

```
header {
```

```
    font-size: 1.5rem;
```

```
}
```

```
header,
```

```
form {
```

```
    min-height: 20vh;
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
}
```

```
form input,
```

```
form button {
```

```
    padding: 0.5rem;
```

```
    padding-left: 1rem;
```

```
    font-size: 1.8rem;
```

```
    border: none;
```

```
    background: white;
```

```
    border-radius: 2rem;
```

```
}
```

```
form button {
```

```
    color: rgb(255, 200, 0);
```

```
    background: white;
```

```
    cursor: pointer;
```

```
    transition: all 0.3 ease;
```

```
    margin-left: 0.5rem;
```

```
}
```

```
form button:hover {
```

```
    color: white;
```

```
    background: rgb(255, 200, 0);
```

```
}
```

```
.fa-plus-circle {
    margin-top: 0.3rem;
    margin-left: -8px;
}

.todo-container {
    display: flex;
    justify-content: center;
    align-items: center;
}

.todo-list {
    min-width: 30%;
    list-style: none;
}

.todo {
    margin: 0.5rem;
    padding-left: 0.5rem;
    background: white;
    color: black;
    font-size: 1.5rem;
    display: flex;
    justify-content: space-between;
    align-items: center;
    transition: 0.5s;
    border-radius: 2rem;
}

.todo li {
    flex: 1;
}

.trash-btn,
.complete-btn {
    color: white;
    border: none;
    padding: 1rem;
    cursor: pointer;
    font-size: 1.5rem;
}
```

```
.complete-btn {
    background: rgb(255, 200, 0);
}

.trash-btn {
    border-top-right-radius: 1.75rem;
    border-bottom-right-radius: 1.75rem;
    background: rgb(171, 171, 171);
}

.todo-item {
    padding: 0rem 0.5rem;
}

.fa-trash,
.fa-check-circle {
    pointer-events: none;
}

.completed {
    text-decoration: line-through;
    opacity: 0.5;
}

.slide {
    transform: translateX(10rem);
    opacity: 0;
}

select {
    -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;
    outline: none;
    border: none;
}

.select {
    margin: 1rem;
    position: relative;
    overflow: hidden;
    border-radius: 50px;
}
```

```
select {
    color:rgb(255, 200, 0);
    width: 10rem;
    height: 3.2rem;
    cursor: pointer;
    padding: 1rem;
}

.select::after {
    content: "\25BC";
    position: absolute;
    background:rgb(255, 200, 0);
    top: 0;
    right: 0;
    padding: 1rem;
    pointer-events: none;
    transition: all 0.3s ease;
}

.select:hover::after {
    background: white;
    color: rgb(255, 200, 0);
}
```

This is for styling my HTML.

## Colors

- Background color: mediumslateblue
- Text color: white
- Button background color: white
- Button text color: rgb(255, 200, 0)
- Completed tasks text color: rgb(255, 200, 0)
- Trash button background color: rgb(171, 171, 171)
- Hover effect: white background with rgb(255, 200, 0) text color

## Fonts

- Font-family: "Poppins", sans-serif
- Font size: Various (1.5rem, 1.8rem)

## Layout

- Minimal margin and padding for a clean look.
- Flexbox used for header and form layout.

```
const todoInput = document.querySelector(".todo-input");

const todoButton = document.querySelector(".todo-button");

const todoList = document.querySelector(".todo-list");

const filterOption = document.querySelector(".filter-todo");

document.addEventListener("DOMContentLoaded", getLocalTodos);

todoButton.addEventListener("click", addTodo);

todoList.addEventListener("click", deleteCheck);

filterOption.addEventListener("change", filterTodo);

function addTodo(event) {

    event.preventDefault();

    const todoDiv = document.createElement("div");

    todoDiv.classList.add("todo");

    const newTodo = document.createElement("li");

    newTodo.innerText = todoInput.value;

    newTodo.classList.add("todo-item");

    todoDiv.appendChild(newTodo);

    //ADDING TO LOCAL STORAGE

    saveLocalTodos(todoInput.value);
```

```
const completedButton = document.createElement("button");

completedButton.innerHTML = '<i class="fas fa-check-circle"></li>';

completedButton.classList.add("complete-btn");

todoDiv.appendChild(completedButton);

const trashButton = document.createElement("button");

trashButton.innerHTML = '<i class="fas fa-trash"></li>';

trashButton.classList.add("trash-btn");

todoDiv.appendChild(trashButton);

todoList.appendChild(todoDiv);

todoInput.value = "";

}

function deleteCheck(e) {

const item = e.target;

if(item.classList[0] === "trash-btn") {

    const todo = item.parentElement;

    todo.classList.add("slide");

    removeLocalTodos(todo);

    todo.addEventListener("transitionend", () => {
        todo.remove();
    });
}
}
```

```
todo.addEventListener("transitionend", function() {
  todo.remove();
}) ;

}

if(item.classList[0] === "complete-btn") {
  const todo = item.parentElement;
  todo.classList.toggle("completed");
}

}

function filterTodo(e) {
  const todos = todoList.childNodes;
  todos.forEach(function(todo) {
    switch(e.target.value) {
      case "all":
        todo.style.display = "flex";
        break;
      case "completed":
        if(todo.classList.contains("completed")) {
          todo.style.display = "flex";
        } else {
          todo.style.display = "none";
        }
    }
  })
}
```

```
        }

        break;

    case "incomplete":

        if(!todo.classList.contains("completed")) {

            todo.style.display = "flex";

        } else {

            todo.style.display = "none";

        }

        break;

    }

}

) ;

}

function saveLocalTodos(todo) {

    let todos;

    if(localStorage.getItem("todos") === null) {

        todos = [];

    } else {

        todos = JSON.parse(localStorage.getItem("todos"));

    }

    todos.push(todo);

    localStorage.setItem("todos", JSON.stringify(todos));

}
```

```
function getLocalTodos() {  
  
  let todos;  
  
  if(localStorage.getItem("todos") === null) {  
  
    todos = [];  
  
  } else {  
  
    todos = JSON.parse(localStorage.getItem("todos"));  
  
  }  
  
  todos.forEach(function(todo) {  
  
    const todoDiv = document.createElement("div");  
  
    todoDiv.classList.add("todo");  
  
    const newTodo = document.createElement("li");  
  
    newTodo.innerText = todo;  
  
    newTodo.classList.add("todo-item");  
  
    todoDiv.appendChild(newTodo);  
  
    const completedButton = document.createElement("button");  
  
    completedButton.innerHTML = '<i class="fas  
fa-check-circle"></i>';  
  
    completedButton.classList.add("complete-btn");  
  
    todoDiv.appendChild(completedButton);  
  
    const trashButton = document.createElement("button");
```

```
    trashButton.innerHTML = '<i class="fas fa-trash"></li>';

    trashButton.classList.add("trash-btn");

    todoDiv.appendChild(trashButton);

    todoList.appendChild(todoDiv);

} ) ;

}

function removeLocalTodos(todo) {

let todos;

if(localStorage.getItem("todos") === null) {

    todos = [];

} else {

    todos = JSON.parse(localStorage.getItem("todos"));

}

const todoIndex = todo.children[0].innerText;

todos.splice(todos.indexOf(todoIndex), 1);

localStorage.setItem("todos", JSON.stringify(todos));

}
```

## JavaScript

### How to add a Task

- Enter a task in the input field.
- Click the "Add" button or press Enter.
- The task is added to the list and stored in local storage.

### Completing a Task

- Click the checkmark button next to a task.
- The task is marked as completed with a strikethrough style.

### Deleting a Task

- Click the trash bin button next to a task.
- The task is removed from the list and local storage.

### Filtering Tasks

- Use the dropdown menu to filter tasks by All, Completed, or Incomplete.