# **Державний вищий навчальний заклад**

# **«Запорізький національний університет»**

# **Міністерства освіти і науки України**

# **Лабораторна робота №2**

# 

# **Виконала студентка**

# **гр. 8.1212:**

# **Зінченко А.О.**

# **м. Запоріжжя**

# **2022-2023 н.р.**

### Крок 1. Ознайомлення з Vue.js:

Ознайомтеся з основними концепціями Vue.js, такими як компоненти, директиви, події та інше.

### Крок 2. Оберіть власний шаблон:

Оберіть один з [варіантів задання](https://bitbucket.org/lisnyak/vue-lab2-src/src/master/).

### Крок 3. Проект Vue.js:

Зробіть копію проекту Vue CLI, створеного у попередній лабораторній роботі.

### Крок 4. ****Створення компонентів****:

Створіть компоненти для відображення списку завдань, видалення та додавання нових завдань у відповідності до обраного макета.

## **Варіант №3**

html/todo-3.html

<!DOCTYPE html>

<html lang="en">

<head>

*<!-- Required meta tags -->*

<meta charset="utf-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1"/>

*<!-- Bootstrap CSS -->*

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"

rel="stylesheet"

integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x"

crossorigin="anonymous"/>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css"/>

<title>Todo example 1</title>

<style>

</style>

</head>

<body>

<section class="vh-100" style="background-color: #eee;">

<div class="container py-5 h-100">

<div class="row d-flex justify-content-center align-items-center h-100">

<div class="col col-lg-9 col-xl-7">

<div class="card rounded-3">

<div class="card-body p-4">

<h4 class="text-center my-3 pb-3">To Do App</h4>

<form class="row row-cols-lg-auto g-3 justify-content-center align-items-center mb-4 pb-2">

<div class="col-12">

<div class="form-outline">

<input type="text" id="form1" class="form-control" />

<label class="form-label" for="form1">Enter a task here</label>

</div>

</div>

<div class="col-12">

<button type="submit" class="btn btn-primary">Save</button>

</div>

<div class="col-12">

<button type="submit" class="btn btn-warning">Get tasks</button>

</div>

</form>

<table class="table mb-4">

<thead>

<tr>

<th scope="col">No.</th>

<th scope="col">Todo item</th>

<th scope="col">Status</th>

<th scope="col">Actions</th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">1</th>

<td>Buy groceries for next week</td>

<td>In progress</td>

<td>

<button type="submit" class="btn btn-danger">Delete</button>

<button type="submit" class="btn btn-success ms-1">Finished</button>

</td>

</tr>

<tr>

<th scope="row">2</th>

<td>Renew car insurance</td>

<td>In progress</td>

<td>

<button type="submit" class="btn btn-danger">Delete</button>

<button type="submit" class="btn btn-success ms-1">Finished</button>

</td>

</tr>

<tr>

<th scope="row">3</th>

<td>Sign up for online course</td>

<td>In progress</td>

<td>

<button type="submit" class="btn btn-danger">Delete</button>

<button type="submit" class="btn btn-success ms-1">Finished</button>

</td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</section>

*<!-- Option 1: Bootstrap Bundle with Popper -->*

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"

integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"

crossorigin="anonymous"

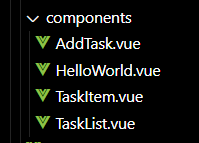
></script>

</body>

</html>

Хід роботи

1. Створення файлів під компоненти відповідно до завдання:



1. Вміст компонентів:

TaskList.vue

<template>

  <div>

    <h2 class="mb-4">Task List</h2>

    <table class="table">

      <thead>

        <tr>

          <th scope="col">No.</th>

          <th scope="col">Todo item</th>

          <th scope="col">Status</th>

          <th scope="col">Actions</th>

        </tr>

      </thead>

      <tbody>

        <task-item

          v-for="(task, index) in tasks"

          :key="task.id"

          :task="task"

          @delete-task="deleteTask(index)"

        ></task-item>

      </tbody>

    </table>

    <button @click="getTasks" class="btn btn-warning">Get tasks</button>

  </div>

</template>

<script>

import TaskItem from './TaskItem.vue';

export default {

  props: {

    tasks: Array

  },

  components: {

    TaskItem

  },

  methods: {

    deleteTask(index) {

      this.$emit("task-deleted", index);

    },

    getTasks() {

      this.$emit("get-tasks");

    }

  }

};

</script>

<style scoped>

/\* Стилі для компонента TaskList \*/

button.btn-warning {

  background-color: #ffc107;

  color: #333;

}

button.btn-warning:hover {

  background-color: #e0a800;

}

</style>

AddTask.vue

<template>

  <div>

    <form class="row row-cols-lg-auto g-3 align-items-center">

      <div class="col-12">

        <div class="form-outline">

          <input v-model="newTask" class="form-control" placeholder="Enter a new task" />

          <label class="form-label">Enter a task here</label>

        </div>

      </div>

      <div class="col-12">

        <button @click="addTask" class="btn btn-primary">Save</button>

      </div>

    </form>

  </div>

</template>

<script>

export default {

  data() {

    return {

      newTask: ""

    };

  },

  methods: {

    addTask() {

      if (this.newTask.trim() !== "") {

        this.$emit("task-added", this.newTask);

        this.newTask = "";

      }

    }

  }

};

</script>

<style scoped>

/\* Стилі для компонента AddTask \*/

button.btn-primary {

  background-color: #007bff;

  color: #fff;

}

button.btn-primary:hover {

  background-color: #0056b3;

}

</style>

TaskItem.vue

<template>

<tr>

<th scope="row">{{ task.id }}</th>

<td>{{ task.text }}</td>

<td>{{ task.status }}</td>

<td>

<button @click="deleteTask" class="btn btn-danger">Delete</button>

<button @click="finishTask" class="btn btn-success ms-1">Finished</button>

</td>

</tr>

</template>

<script>

export default {

props: {

task: Object

},

methods: {

deleteTask() {

this.$emit("delete-task", this.task);

},

finishTask() {

this.$emit("finish-task", this.task);

}

}

};

</script>

<style scoped>

/\* Стилі для компонента TaskItem \*/

button.btn-danger {

background-color: #dc3545;

color: #fff;

border: none;

cursor: pointer;

margin-left: 10px;

}

button.btn-success {

background-color: #28a745;

color: #fff;

border: none;

cursor: pointer;

margin-left: 10px;

}

button:hover {

background-color: #e0a800;

}

</style>

Вміст файлу App.vue:

<template>

<div>

<section class="vh-100" style="background-color: #eee;">

<div class="container py-5 h-100">

<div class="row d-flex justify-content-center align-items-center h-100">

<div class="col col-lg-9 col-xl-7">

<div class="card rounded-3">

<div class="card-body p-4">

<h4 class="text-center my-3 pb-3">To Do App</h4>

<add-task @task-added="addTask" />

<task-list :tasks="tasks" @task-deleted="deleteTask" @finish-task="finishTask" />

</div>

</div>

</div>

</div>

</div>

</section>

</div>

</template>

<script>

import AddTask from './components/AddTask.vue';

import TaskList from './components/TaskList.vue';

export default {

components: {

'add-task': AddTask,

'task-list': TaskList,

},

data() {

return {

tasks: [

{ id: 1, text: 'Buy groceries for next week', status: 'In progress' },

{ id: 2, text: 'Renew car insurance', status: 'In progress' },

{ id: 3, text: 'Sign up for an online course', status: 'In progress' },

],

};

},

methods: {

addTask(newTaskText) {

const newTask = {

id: this.tasks.length + 1,

text: newTaskText,

status: 'In progress',

};

this.tasks.push(newTask);

},

deleteTask(index) {

this.tasks.splice(index, 1);

},

finishTask(index) {

this.tasks[index].status = 'Finished';

},

},

};

</script>

<style>

/\* Додані стилі для App.vue \*/

body, html {

height: 100%;

margin: 0;

}

.card {

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

.btn-primary {

background-color: #007bff;

color: #fff;

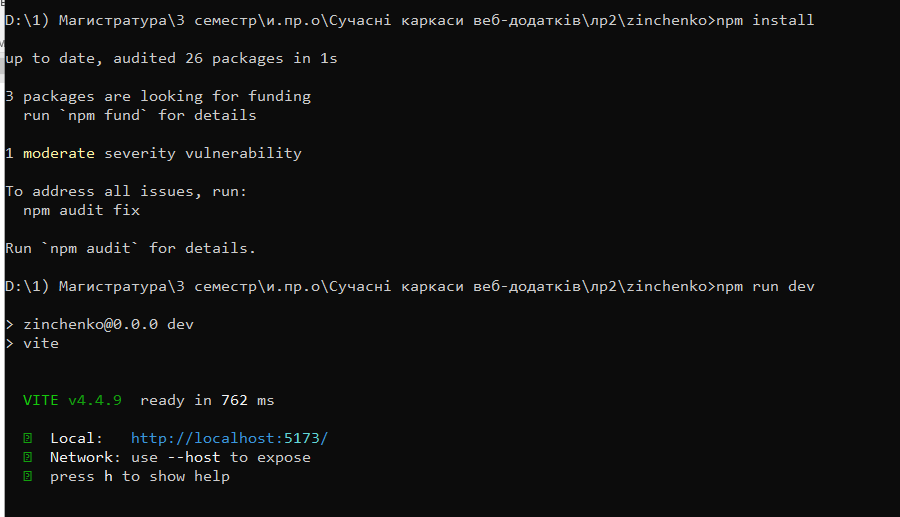
}

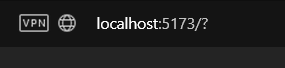
.btn-primary:hover {

background-color: #0056b3;

}

</style>

Запуск:  




## 

## **Контрольні питання**

1. Які основні концепції Vue.js були використані під час створення додатку?

* Компоненти дозволяють розділити інтерфейс користувача на малі, незалежні модулі, що полегшує підтримку і розширення додатку;
* Vue.js автоматично відстежує зміни даних та оновлює DOM відповідно до цих змін, що дозволяє писати декларативний код без необхідності вручну оновлювати DOM;
* спеціальні директиви, такі як v-if, v-for, v-bind, v-on, використовуються для взаємодії з DOM та даними у шаблоні компонента;
* можливість обробки подій, таких як кліки, зміни інпутів, і багато інших, є ключовою концепцією Vue.js. Для цього використовуються директиви v-on.

1. Які кроки необхідно виконати для створення проекту Vue.js за допомогою Vue CLI?
2. Встановити Vue CLI за допомогою команди: npm install -g @vue/cli;
3. створити новий проект Vue: vue create project-name;
4. виберати налаштування проекту або використати налаштування за замовчуванням;
5. Яким чином ви реалізували можливість додавання нових завдань та видалення завдань у додатку?

Можна використовувати локальний стан додатку, наприклад, масив для зберігання завдань.

1. Які основні директиви Vue використовуються для взаємодії з DOM-елементами та даними в шаблоні компонента?

* v-bind або : використовується для прив'язки значень з даних до атрибутів DOM-елементів;
* v-on або @ використовується для відслідковування подій та виклику методів при виникненні подій;
* v-model використовується для двостороннього зв'язку між введенням користувача та даними в компоненті;
* v-if та v-else використовується для умовного відображення DOM-елементів в залежності від значень даних;
* v-for використовується для ітерації через масиви або об'єкти та відображення кожного елемента у списку.