**Міністерство освіти і науки України**

**Національний технічний університет України**

**“Київський політехнічний інститут”**

**Факультет прикладної математики**

**Кафедра спеціалізованих комп’ютерних систем**

**Лабораторна робота № 3**

з дисципліни «ВЕБ-дизайн»

«**Розробка Web-додатка засобами Javascript/VueJS**»

Виконав:

студент групи КВ-83

**Лазуткін Олег**

**Зміст**

**Мета**: ознайомитись із засобами фреймворка VueJS та навчитись створювати асинхронні запити до Web-сервера.

**Загальне завдання**: адаптувати програмний код ЛР№2 до вимог фреймворка VueJS та забезпечити завантаження необхідних даних з Web-сервера

**Інструменти розробки**: Javascript, VueJS, NodeJS, Express.

**Зауваження**: студент може обрати інший серверний фреймворк (наприклад, Django).

**Завдання**

**Мій Telegram:**

**Github-репозиторій з лабораторної:**

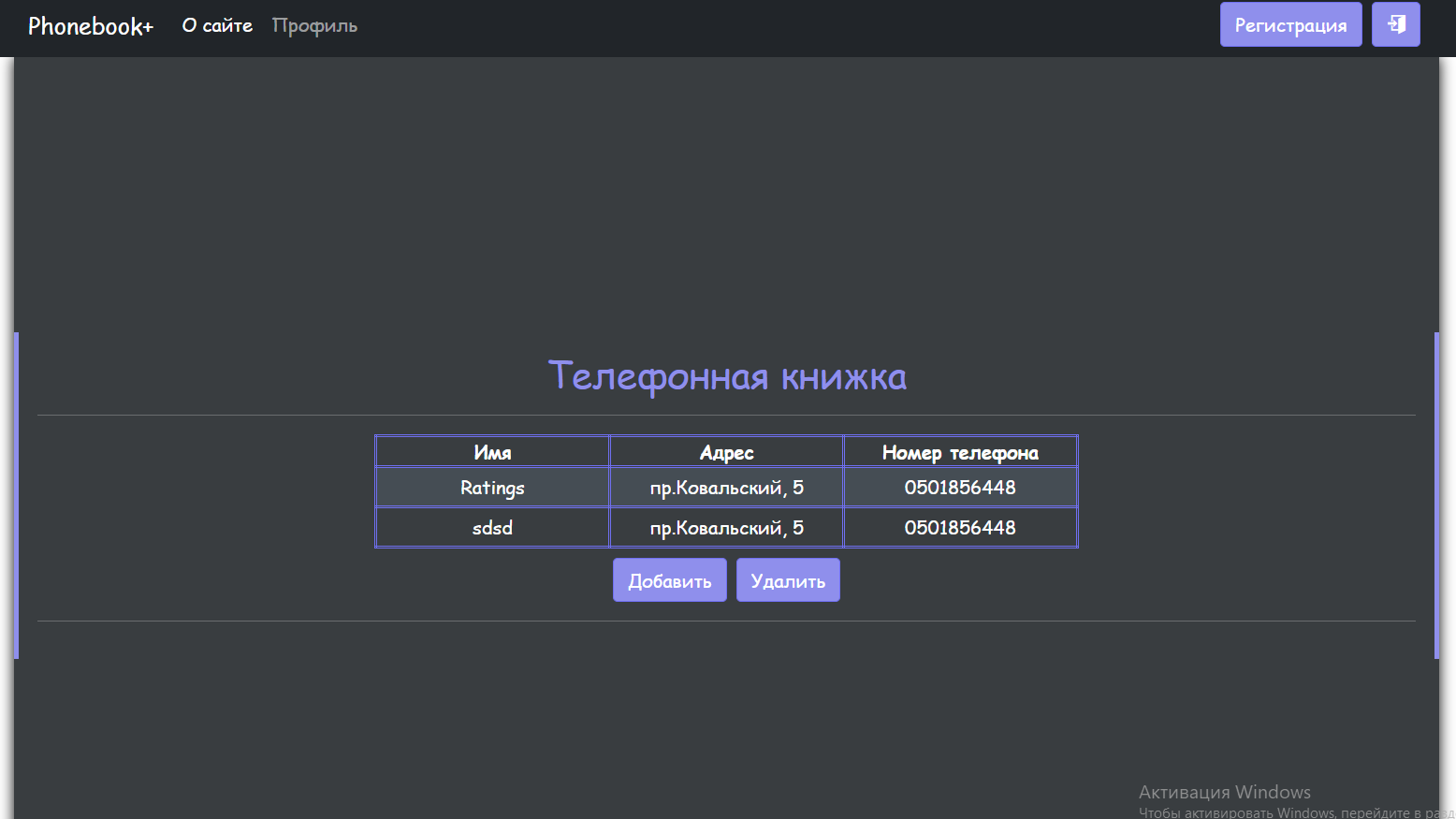
**Тема:  Адаптація лабораторної роботи №2 до вимог Vue. Cтворення веб-серверу за допомогою NodeJS + Express.**

       У цій лабораторній роботі було створено додаток за допомогою фреймворка Vue. Було використано vue-router для навігації та vuex для взаємодії з данними.

  Клієнт взаємодіє з сервером за допомогою бібліотеки fetchAPI.

Сервер реалізовано за допомогою NodeJS + Express. Як сховище данних використовується SQLite3. Для більш простої взаємодії з базою данних використано  ORM Sequelize.

**Скріншоти**



**Код**

**App.vue**

<template>

  <router-view />

</template>

<style>

#app {

  font-family: Avenir, Helvetica, Arial, sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

  text-align: center;

  color: #2c3e50;

}

#nav {

  padding: 30px;

}

#nav a {

  font-weight: bold;

  color: #2c3e50;

}

#nav a.router-link-exact-active {

  color: #42b983;

}

</style>

**Home.vue**

<template>

  <div class="main">

    <div class="main\_\_wrapper">

      <div class="main\_\_inner">

        <h2 class="title">Телефонная книжка</h2>

*<!-- <div class="divider"></div> -->*

        <hr />

        <div class="phonebook">

          <div *v-if*="$store.state.model.items.length" class="main-app">

            <table>

              <tr>

                <th>Имя</th>

                <th>Адрес</th>

                <th>Номер телефона</th>

              </tr>

              <record

*v-for*="item in $store.state.model.items"

                v-bind:key="item.id"

                v-bind:record="item"

              />

            </table>

          </div>

          <h1 *v-else*>Записей нет</h1>

          <form>

            <div class="buttons">

              <router-link

                to="/add"

                type="button"

                id="add"

                class="btn btn-dark"

              >

                Добавить

              </router-link>

              <button

                @click="deleteItems()"

                type="button"

                id="delete"

                class="btn btn-dark"

              >

                Удалить

              </button>

            </div>

          </form>

        </div>

        <hr />

      </div>

    </div>

  </div>

</template>

<script>

*import* record *from* "../components/record.vue";

*export* *default* {

  name: "Home",

  components: {

    record,

  },

  methods: {

    deleteItems() {

      this.$store.commit("DELETE");

    },

  },

  created() {

    this.$store.commit("LOAD");

  },

};

</script>

**Add.vue**

<template>

  <div class="main">

    <div class="main\_\_wrapper">

      <div class="main\_\_inner">

        <h2 class="title">Добавить</h2>

*<!-- <div class="divider"></div> -->*

        <hr />

        <div class="phonebook">

          <form>

            <div class="text">Введите имя:</div>

            <input

              v-model="name"

              type="text"

              class="form-control"

              placeholder="Имя"

              id="name"

            />

            <div class="text">Адрес:</div>

            <input

              v-model="adress"

              type="text"

              class="form-control"

              placeholder="Адрес"

              id="adress"

            />

            <div class="text">Номер телефона:</div>

            <input

              v-model="tel"

              type="text"

              class="form-control"

              placeholder="Телефон"

              id="tel"

            />

            <div class="buttons">

              <router-link

                to="/"

                @click="add()"

                type="button"

                id="add"

                class="btn btn-dark"

              >

                Добавить

              </router-link>

              <router-link

                to="/"

                type="button"

                id="delete"

                class="btn btn-dark"

              >

                Журнал

              </router-link>

            </div>

          </form>

        </div>

        <hr />

      </div>

    </div>

  </div>

</template>

<script>

*export* *default* {

  data() {

*return* {

      name: "",

      tel: "",

      adress: "",

    };

  },

  methods: {

    add() {

*if* (+this.tel && (this.tel.length == 12 || this.tel.length == 10)) {

        this.$store.commit("ADD", {

          name: this.name,

          tel: this.tel,

          adress: this.adress,

        });

      } *else* {

        alert("Неверный формат номера");

      }

    },

  },

};

</script>

**Model.js**

*export* *default* class Model {

  constructor() {

    this.items = [];

  }

  add(name, tel, adress) {

    const item = {

      id: Math.round(Math.random() \* 10000).toString(),

      name: name,

      tel: tel,

      adress: adress,

      chosed: false,

    };

    this.items.push(item);

    this.sendItem();

  }

  delete() {

    let i = 0;

*while* (i != this.items.length) {

*if* (this.items[i].chosed) {

        this.items.splice(i, 1);

        i--;

      }

      i++;

    }

    this.sendItem();

  }

  choose(id) {

    const index = this.items.findIndex((item) => {

*return* item.id == id;

    });

    console.log(index);

    this.items[index].chosed = !this.items[index].chosed;

    console.log(id);

  }

  async loadItems() {

*await* fetch("http://localhost:3000/")

      .then((res) => res.json())

      .then((res) => {

        console.log(res.items);

        this.items = res.items;

      });

  }

  async sendItem() {

*await* fetch("http://localhost:3000/push", {

      method: "POST",

      body: JSON.stringify({ items: this.items }),

      headers: {

        "Content-Type": "application/json",

      },

    });

  }

}