

Факультет инфокоммуникационных технологий.

Практическая работа №4 по теме:  
«Реализация клиентской части средствами Vue.js» по дисциплине:  
Web-программирование.

Выполнил: Кирильцев Роман  
Группа: К33401

Преподаватель: Говоров Антон Игоревич

Санкт-Петербург, 2021

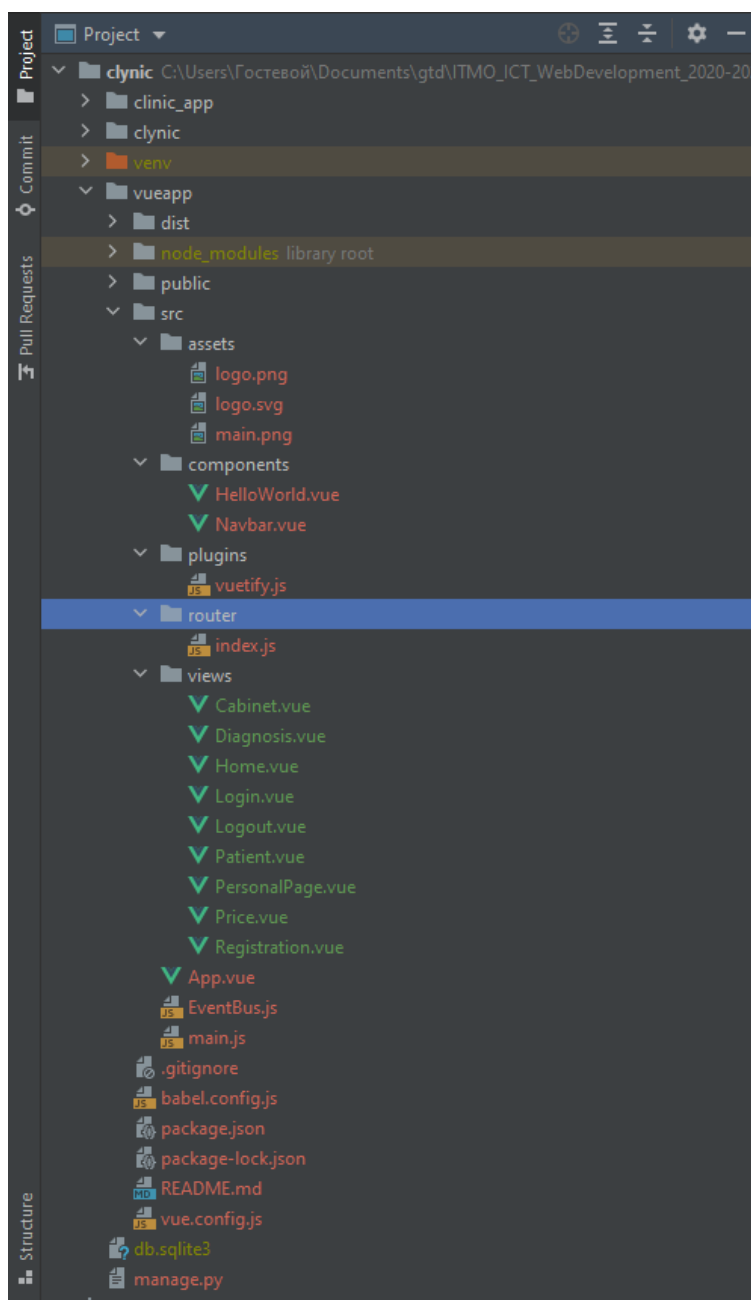
**Цель работы:** овладеть практическими навыками и умениями реализации web-сервисов средствами Vue.js.

**Практическое задание :**

Реализация клиентской части приложения средствами vue.js.

Для программной системы, реализованной в рамках лабораторной работы 3 был реализован web-интерфейс, взаимодействующий с серверной частью посредством обращения к API, созданного с помощью Django REST.

Структура проекта на Vue.js:



Пример листинга одной из компонент - Patient.vue

```
<template>
  <v-data-table
    :headers="headers"
    :items="patienty"
    class="elevation-1"
  >
    <template v-slot:top>
      <v-toolbar
        flat
      >
        <v-toolbar-title>Пациенты</v-toolbar-title>
        <v-divider
          class="mx-4"
          inset
          vertical
        ></v-divider>
        <v-spacer></v-spacer>
        <v-dialog
          v-model="dialog"
          max-width="500px"
        >
          <template v-slot:activator="{ on, attrs }">
            <v-btn
              color="blue darken-2"
              dark
              class="mb-2"
              v-bind="attrs"
              v-on="on"
            >
              Добавить пациента
            </v-btn>
          </template>
        </v-dialog>
      </v-toolbar>
      <v-card>
        <v-card-title>
          <span class="headline">{{ formTitle }}</span>
        </v-card-title>
        <v-card-text>
          <v-container>
            <v-row>
              <v-col
                cols="12"
                sm="6"
                md="4"
              >
                </v-col>
              <v-col
                cols="12"
                sm="6"
                md="4"
              >
                <v-text-field
                  v-model="editedItem.fio"
                  label="ФИО пациента"
                ></v-text-field>
              </v-col>
              <v-col
                cols="12"
                sm="6"
              >
```

```

        md="4"
      >
        <v-text-field
          v-model="editedItem.birthdate"
          label="Дата рождения"
        ></v-text-field>
      </v-col>
      <v-col
        cols="12"
        sm="6"
        md="4"
      >
        <v-text-field
          v-model="editedItem.phone"
          label="Номер телефона"
        ></v-text-field>
      </v-col>
      <v-col
        cols="12"
        sm="6"
        md="4"
      >
        <v-text-field
          v-model="editedItem.passport"
          label="Паспорт"
        ></v-text-field>
      </v-col>
      <v-col
        cols="12"
        sm="6"
        md="4"
      >
    </v-col>
  </v-row>
</v-container>
</v-card-text>

<v-card-actions>
  <v-spacer></v-spacer>
  <v-btn
    color="gray darken-1"
    text
    @click="close"
  >
    Отмена
  </v-btn>
  <v-btn
    color="gray darken-4"
    text
    @click="save"
  >
    Сохранить
  </v-btn>
</v-card-actions>
</v-card>
</v-dialog>
<v-dialog v-model="dialogDelete" max-width="500px">
  <v-card>
    <v-card-title class="headline">Вы уверены, что хотите удалить
этого пациента?</v-card-title>
    <v-card-actions>
      <v-spacer></v-spacer>
      <v-btn color="gray darken-1" text
@click="closeDelete">Назад</v-btn>

```

```

        <v-btn color="gray darken-1" text
@click="deleteItemConfirm">Удалить</v-btn>
        <v-spacer></v-spacer>
    </v-card-actions>
</v-card>
</v-dialog>
</v-toolbar>
</template>
<template v-slot:item.actions="{ item }">
    <v-icon
        small
        class="mr-2"
        @click="editItem(item)"
    >
        mdi-pencil
    </v-icon>
    <v-icon
        small
        @click="deleteItem(item)"
    >
        mdi-delete
    </v-icon>
</template>
</v-data-table>
</template>

<script>
import axios from 'axios'
export default {
  data: () => ({
    dialog: false,
    dialogDelete: false,
    headers: [
      {
        text: 'Пациент',
        align: 'start',
        sortable: false,
        value: 'fio'
      },
      { text: 'Дата рождения', value: 'birthdate' },
      { text: 'Номер телефона', value: 'phone' },
      { text: 'Паспорт', value: 'passport' },
      { text: 'Actions', value: 'actions', sortable: false }
    ],
    patienty: [],
    editedIndex: -1,
    editedItem: {
      name: '',
      price: ''
    },
    defaultItem: {
      name: '',
      price: ''
    }
  }),
  computed: {
    formTitle () {
      return this.editedIndex === -1 ? 'Новый пациент' : 'Редактировать
данные пациента'
    }
  },
  watch: {
    dialog (val) {
      val || this.close()

```

```

    },
    dialogDelete (val) {
      val || this.closeDelete()
    }
  },
  created () {
    this.patientList()
  },
  methods: {
    async patientList () {
      try {
        const response = await axios
          .get('http://127.0.0.1:8000/api/patient/')
        if (response.status !== 200) {
          throw new Error(response.status)
        }
        this.patients = response.data
        return response.data
      } catch (e) {
        console.error('AN API ERROR', e)
      }
    },
    editItem (item) {
      this.editedIndex = this.patients.indexOf(item)
      this.editedItem = Object.assign({}, item)
      this.dialog = true
    },
    deleteItem (item) {
      this.editedIndex = this.patients.indexOf(item)
      this.editedItem = Object.assign({}, item)
      this.dialogDelete = true
    },
    deleteItemConfirm () {
      if (this.editedIndex > -1) {
        this.patients.splice(this.editedIndex, 1)
        this.closeDelete()
        axios.delete('http://127.0.0.1:8000/api/patient/' +
JSON.stringify(this.editedItem.id), {
          id: this.editedItem.id,
          fio: this.editedItem.fio,
          birthdate: this.editedItem.birthdate,
          phone: this.editedItem.phone,
          passport: this.editedItem.passport
        })
      } else {
        this.patients.push(this.editedItem)
      }
      this.close()
    },
    close () {
      this.dialog = false
      this.$nextTick(() => {
        this.editedItem = Object.assign({}, this.defaultItem)
        this.editedIndex = -1
      })
    },
    closeDelete () {
      this.dialogDelete = false
      this.$nextTick(() => {
        this.editedItem = Object.assign({}, this.defaultItem)
        this.editedIndex = -1
      })
    },
    async createPatient () {

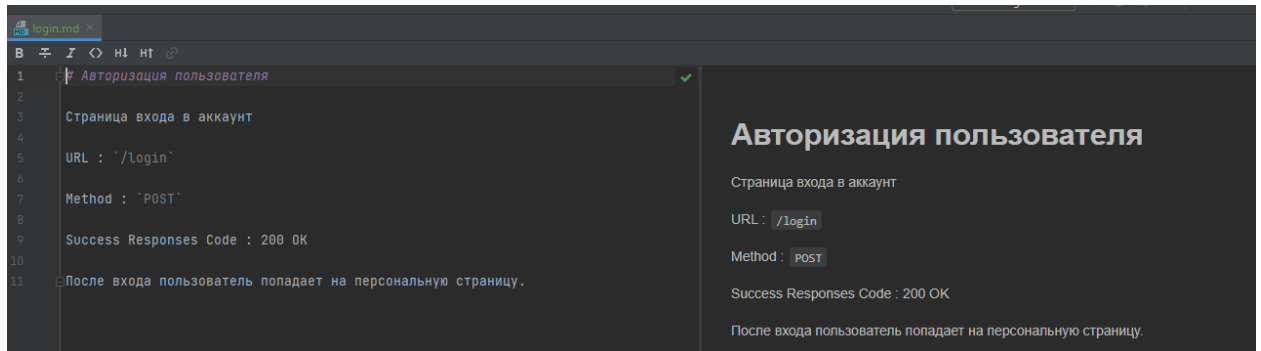
```

```

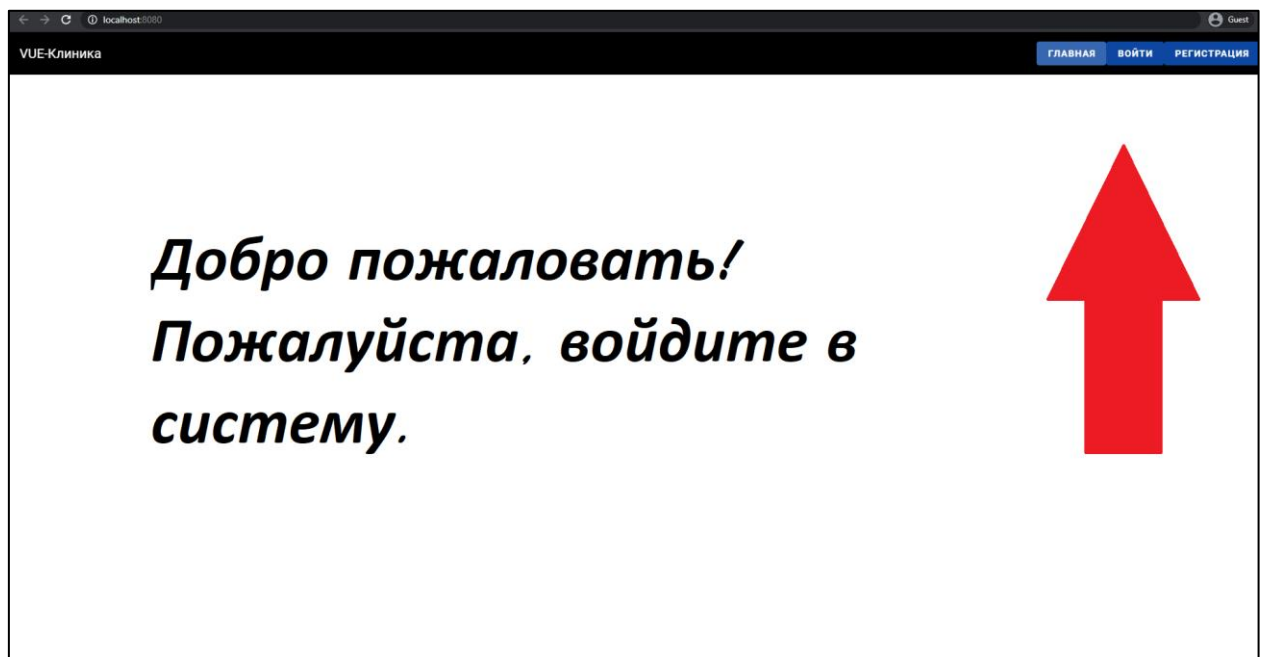
    try {
      const response = await this.axios
        .post('http://localhost:8000/api/patient/create/', {
          fio: this.editedItem.fio,
          birthdate: this.editedItem.birthdate,
          phone: this.editedItem.phone,
          passport: this.editedItem.passport
        })
      if (response.status !== 201) {
        throw new Error(response.status)
      }
      window.location.reload()
    } catch (e) {
      console.error('API ERROR', e)
    }
  },
  async updatePatient () {
    this.editedIndex = 1
    try {
      const response = await this.axios
        .put('http://localhost:8000/api/patient/' +
JSON.stringify(this.editedItem.id) + '/', {
          fio: this.editedItem.fio,
          birthdate: this.editedItem.birthdate,
          phone: this.editedItem.phone,
          passport: this.editedItem.passport
        })
      if (response.status !== 200) {
        throw new Error(response.status)
      }
      window.location.reload()
    } catch (e) {
      console.error('API ERROR', e)
    }
  },
  save () {
    if (this.editedIndex > -1) {
      this.updatePatient()
    } else {
      this.createPatient()
    }
  }
}
}
</script>
<style scoped>
</style>

```

Для проекта была реализована документация, описывающая работу разработанных интерфейсов средствами MkDocs.



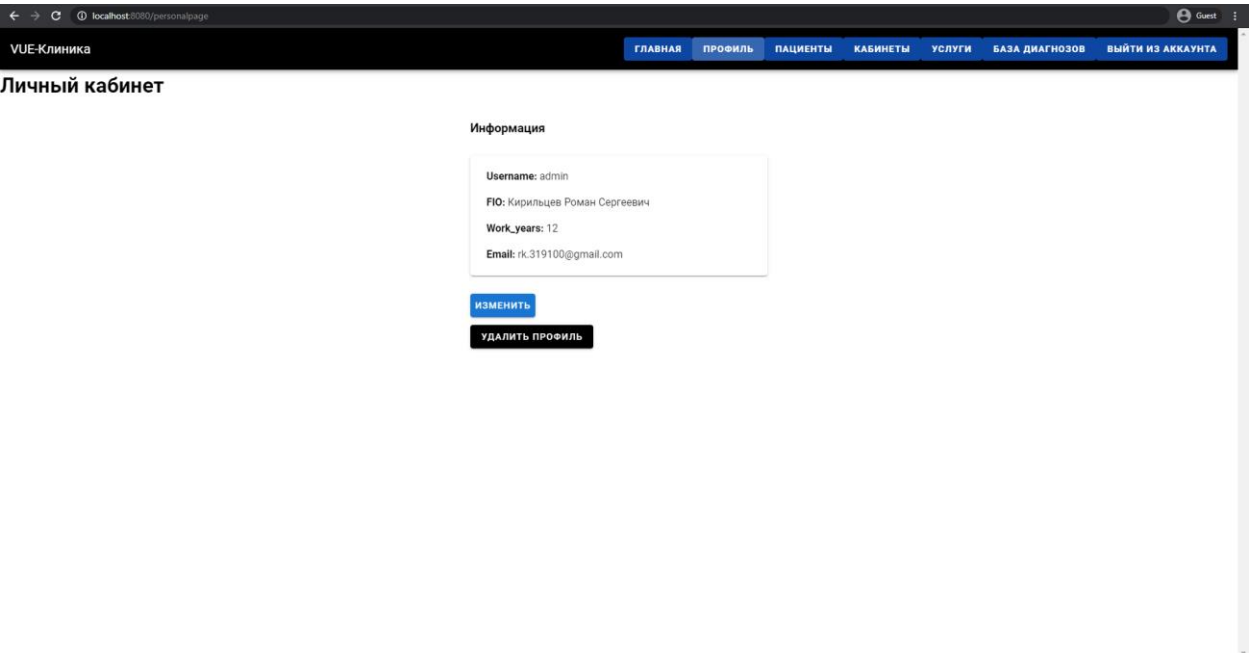
Пример интерфейса для неавторизованного пользователя:





Пример интерфейса для авторизованного пользователя:

Главная страница





Список кабинетов

ВУЕ-Клиника

ГЛАВНАЯ ПРОФИЛЬ ПАЦИЕНТЫ КАБИНЕТЫ УСЛУГИ БАЗА ДИАГНОЗОВ ВЫЙТИ ИЗ АККАУНТА

Кабинеты

ДОБАВИТЬ КАБИНЕТ

Номер кабинета	Ответственный	Номер телефона	Actions
12	РОМАН	89049049090	 

Rows per page: 10 1-1 of 1



База диагнозов

ВУЕ-Клиника

ГЛАВНАЯ ПРОФИЛЬ ПАЦИЕНТЫ КАБИНЕТЫ УСЛУГИ БАЗА ДИАГНОЗОВ ВЫЙТИ ИЗ АККАУНТА

Диагнозы

ДОБАВИТЬ ДИАГНОЗ

Диагноз	Actions
Помутнение	 

Rows per page: 10 1-1 of 1