Міністерство освіти і науки України Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

	Po ₃	paxy	ункова	робота
--	-----------------	------	--------	--------

3 дисципліни: «Веб-технології та веб-дизайну »

Виконав:

Студентки групи КН-208

Гетьман Соломії

Прийняв:

Гасько Р.Т.

Проект призначений для формування реєстру первинних імунодефіцитів, що забезпечить облік хворих на дану патологію, внесення їхніх результатів обстеження, моніторинг стану хворого за результатами застосування імуноглобулінів, забезпечення можливості аналізу динаміки перебігу хвороби та корегування імуноглобулінотерапії у разі необхідності.

Основними складовими проекту щодо формування реєстру первинних імунодефіцитів є:

- •Серверна частина з базою даних, яка містить інформацію про об'єкти інформатизації.
- •Клієнтська частина для внесення інформації про об'єкти інформатизації та формування аналітичних звітів.

Користувачі та їх функціонал

Admin -

- 1. Створює клієнтів (login, passvord, рівень доступу, email, прізвище, ім'я, по батькові, регіон, лікарня(центр), категорія)
- 2. оновлює інформацію про ліки, лабораторії
- 3. відповідає за налаштування контенту

Doctor -

- 1. Вносить та редагує дані про:
 - 1.1. загальну інформацію про хворого
 - 1.2. шлях до діагнозу
 - 1.3. діагнозу первинного імунодефіциту
 - 1.4. стовбурових клітин
 - 1.5. замінну імунотерапію
- 2. Додає інформацію про:
 - 2.1.шлях до діагнозу
 - 2.2. діагнозу первинного імунодефіциту
 - 2.3. стовбурових клітин
 - 2.4. замінну імунотерапію
- 3. Пошук за:
 - 3.1. номером реєстрації
 - 3.2. віком
 - 3.3. за генетичним анамнезом
 - 3.4. імуноглобуліном
 - 3.5. по регіону
- 4. Вивід історії записів по хворих:

- реєстраційний номер
- вік
- дата встановленого діагнозу
- дата останнього введення замісної терапії
- призначене дозування

Expert_center:

- 1. Пошук пацієнтів в межах обласного центру:
 - реєстраційний номер
 - вік
 - дата встановленого діагнозу
 - дата останнього введення замісної терапії
 - призначене дозування
- 2. Аналітика по місяцях, роках
- 3. Відкріплення та прикріплення хворих від центрів дитячого відділення до дорослого

Expert_region:

- 1. Пошук пацієнтів в межах регіонального центру:
 - реєстраційний номер
 - вік
 - дата встановленого діагнозу
 - дата останнього введення замісної терапії
 - призначене дозування
 - центр
- 2. Аналітика по місяцях, роках, по центрах.

Expert _country:

- 1. Пошук пацієнтів по областях, центрах:
 - реєстраційний номер
 - вік
 - дата встановленого діагнозу
 - дата останнього введення замісної терапії
 - призначене дозування
 - центр
 - область
- 2. Ааналітика по місяцях, роках, по центрах, по областях.

Backend:

- Реєстрація пацієнта;
- Реєстрація лікаря;
- Редагування даних про пацієнта;
- Додавання та редагування нової інформації.

Frontend:

- Інтерфейс сервісу;
- Реєстрація пацієнта;
- Редагування даних пацієнта;
- Додавання нової інформації;
- Перегляд лікарем його пацієнтів;
- Пошук пацієнта за критерієм.

Mobile:

- Реєстрація лікаря;
- Редагування даних про пацієнта;
- Додавання та редагування нової інформації.

Учасники команди:

- Стасюк Андрій
- Гетьман Соломія
- Куба Максим
- Славчаник Олеся
- Мамчур Назар

Посилання на GitHub з даним проектом:

https://github.com/SolyaHetman/medical-service

Приклади функцій із fronted:

Method Post

```
EventBus.$on('postToDB', function (payLoad) {
  tempObj = Object.assign(tempObj, payLoad);
  const url = 'http://localhost:3000/users';
  axios.post(url, tempObj)
    .then(res =>console.log('Saved'))
    .catch(err => console.log(err))
});
```

Method Get

```
},
created() {
  console.log(this.$route.params.user);
},
mounted() {
 var self = this;
  const id = this.$route.params.user;
  axios.get('http://localhost:3000/users',{
    params: {
      id: this.$route.params.user
  })
  .then(function(res){
    self.users = res.data;
    console.log('Data :', res.data);
  })
  .catch(function(error){
    console.log('Error :', error)
  })
```

Method Put

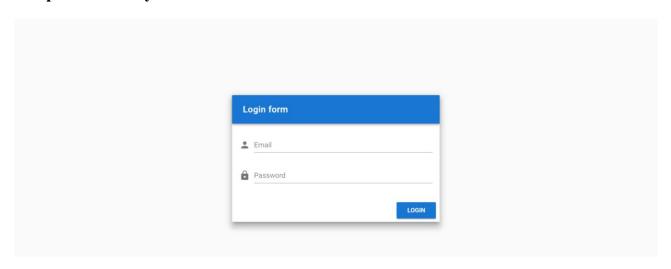
```
},
submit(user){
    this.shouldDisable = true;
    console.log('edit: ', user),
    axios.put(`http://localhost:3000/users/${user.id}`,user)
    .then(response => {
        console.log(response);
    })
    .catch(error => {
        console.log(err);
    });
    this.alertShow = true;
```

Validation

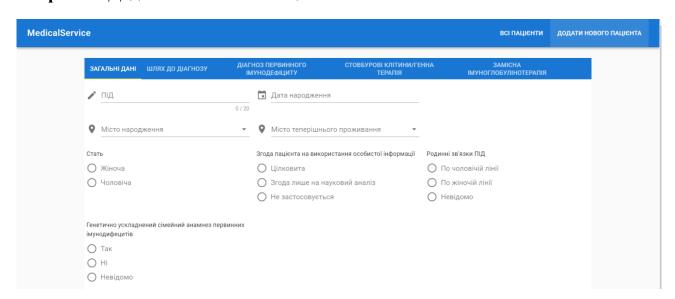
```
export default {
    data(){
    return{
        pidDiagnosis:{
            damage_genes: null,
            genetic_research_data: null,
            sequencing_method: null,
            research_lab: null,
            genetic_research_reason: null,

        },
        dataRules: [
            (v) => /^(\d{1,2})-(\d{4})$/.test(v) || 'Введіть ДД-ММ-РР'
        ],
        }
}
```

Сторінка «входу»



Сторінка «Додавання нового пацієнта»



Сторінка «Перегляду основної інформації про зареєстрованого пацієнта»

