

# Catch-up

---

[Готовий додаток](#)

[Стартова точка](#)

## Задача #1

**Реалізувати логіку роботи візарду реєстрації.**

*Сторінка index.html*

*Розробку проводити у файлі `src/pages/index/scripts/wizard.js`*

1. При натисканні на кнопку зареєструватися має відображатися форма `Хто ти? :-)`.
2. При натисканні на кнопку `Продовжити` на формі `Хто ти? :-)` має відкритися форма `Зареєструйся зараз!`.
3. При натисканні на кнопку кроку форма повинна повертатися на потрібний крок.



## Зареєструйся сейчас!

Предоставьте нам информацию ниже  
для создания аккаунта.

Имя и фамилия



Имя и фамилия

Электронная почта



Email

Пароль



Password

Пароль (повторно)



Password



Создать аккаунт



5. При нажатии на кнопку **назад**, форма должна вернуться на предыдущий шаг.



## Зареєструйся сейчас!

Предоставьте нам информацию ниже  
для создания аккаунта.

### Задача #2

Реалізувати логіку реєстрації студента та викладача.

Сторінка `index.html`

Розробку проводити у файлі `src/pages/index/scripts/wizard.js`

1. На кроці `Хто ти? :-)` в об'єкт користувача необхідно записати властивість `type` зі значенням `student` або `teacher` залежно від вибору користувача. Після цього кроку об'єкт повинен мати такий вигляд.

```
{  
  type: 'student'  
}
```

2. На кроці `Зареєструйся зараз!` необхідно взяти значення полів форми та заповнити поля об'єкта `name`, `email`, `password`. Після цього кроку об'єкт повинен мати такий вигляд.

```
{  
  type: 'student',  
  name: 'John Doe',  
  email: 'jdoe@email.com',  
  password: 123456  
}
```

3. На кроці `Зареєструйся зараз!` після натискання кнопки `Створити обліковий запис`. Необхідно записати дані у локальне сховище. **Зверніть увагу** — у додатку може зареєструватися 1 викладач та скільки завгодно студентів.
  1. Якщо реєструється викладач, то локальне сховище (ключ - `teacher`) записується об'єкт.
  2. Якщо реєструється студент, то в такому разі в локальне сховище (ключ - `students`) записується масив з об'єктом студента. При реєстрації 2-го студента необхідно прочитати з локального сховища масив із вже збереженими студентами та дописати до нього новий об'єкт. Після цього виконати збереження нового масиву в локальне сховище.
4. Після натискання кнопки `Створити обліковий запис` необхідно очистити форму та виконати переадресацію користувача на потрібну сторінку. Якщо паролі не збігаються - елементам з ідентифікатором `password` та `password_next` необхідно додати клас `error`. Для редагування користувача скористайтесь `window.location.href`.

1. Якщо реєструється студент – переадресувати потрібно на сторінку `student.html`
2. Якщо реєструється викладач – переадресувати потрібно на сторінку `teacher.html`
5. Додати валідацію для імені користувача - поле форми `Ім'я та прізвище` повинно містити 2 сторки розділені пробілом, довжина кожного рядка повинна бути не менше або дорівнює 3-м символам. Якщо валідація не пройшла, необхідно додати елементу з ідентифікатором `name` клас `error`.

## Задача #3

Реалізувати логіку бронювання уроку.

Сторінка `student.html`

Розробку проводити у файлі `src/pages/student/scripts/lessons.js`


1. При виборі уроку та слота часу необхідно записати до локального сховища масив з даними. Для запису використати ключ `lessons`. Масив має містити об'єкти. Якщо в локальному сховищі вже є масив, необхідно дописати до нього новий об'єкт.
2. **Зверніть увагу** - уроки та слоти часу - це радіокнопки. Вам необхідно буде знайти в розмітці елемент із атрибутом `checked`. Наприклад,  
`document.querySelector('input[type="radio"]:checked')`.

```
{
  name: 'John Doe', // взяти з локального сховища за ключом 'login' об'єкт і звідти
отримати ім'я користувача
  time: 12, // взяти з файлу з константами src/pages/student/scripts/constants.js з
об'єкта timeSlots
  tomorrow: true, // ідентифікатори time_01..time_03 відносяться до сьогоднішнього
дня, time_04..time_06 до завтрашнього
  title: 'Аудіювання', // взяти з файлу з константами
src/pages/student/scripts/constants.js з об'єкта lessons за типом елемента type_1...
  duration: 90 // взяти з файлу з константами src/pages/student/scripts/constants.js з
об'єкта lessons за типом елемента type_1...
}
```

## Выбери тип занятия 😊


time\_4..time\_6

time\_1..time\_3




~ 120 минут  
**Новый урок**  
Будем вместе изучать  
новые темы

type\_1




~ 90 минут  
**Аудирование**  
Учимся воспринимать  
речь на слух

type\_2



~ 60 минут  
**Рефреш знаний**  
Обзор уже изученных  
тем

type\_3



~ 60 минут  
**Тестирование**  
Быстрый чек  
изученной темы

type\_4

## Забукай слоты 🕒

Сегодня

12:00-14:00

14:00-16:00

16:00-18:00

Завтра

12:00-14:00

14:00-16:00

16:00-18:00

Выйти

Забукать занятие

## Задача #4\* (опціонально)

Реалізувати логіку відображення запланованих уроків.

Сторінка `teacher.html`

Розробку проводити у файлі `src/pages/teacher/scripts/lessons.js`

1. Витягти з локального сховища дані по ключу `lessons`.

- Використовуючи як приклад `src/pages/teacher/scripts/reviews.js` створити код, який згенерує розмітку і додасть її в html.
- Зверніть увагу, що час заняття необхідно буде розрахувати виходячи з часу початку заняття та тривалості заняття. Час початку заняття заповнюється на сторінці *student.html* - поле об'єкта - `time`. Також визначається коли має відбутися заняття сьогодні/завтра — поле об'єкта `tomorrow`. Наприклад, якщо час початку 12, а тривалість заняття 90 хвилин, то в запланованому уроці має відобразитися 12:00 - 13:30.

## Запланированные уроки



Завтра, 14:00 — 16:00

**Анна Перминова (default)**

Новый урок

### Зверніть увагу!

- Для запису в локальне сховище потрібно використовувати `localStorage.setItem('key', 'value');`

// Приклад запису об'єкта

```
localStorage.setItem('key', JSON.stringify({ name: 'John' }));
```

// Приклад запису масиву з об'єктом

```
localStorage.setItem('key', JSON.stringify([ { name: 'John' } ]));
```

- Для читання з локального сховища потрібно використовувати - `localStorage.getItem('key');`

```
const data = JSON.parse(localStorage.getItem('key'));
```

- Для видалення з локального сховища потрібно використовувати `localStorage.removeItem('key');`

```
localStorage.removeItem('key')
```

- `key` ← ключ за яким відбувається збереження, читання, видалення запису з/в локального сховища. Ключ задає розробник.
- Допускається додавання до html ідентифікаторів, дата атрибутів, АЛЕ забороняється зміна

розмітки.