**Task 1**

Необхідно розробити головну сторінку веб-сайту. Тематика сайту спортивна. Сайт може бути присвячено  будь-якому виду спорту, команді чи спортсмену. Наприклад, футбольний сайт Динамо Київ, боксерський сайт Василя Ломаченко, тенісний Еліни Світоліної тощо. На головній сторінці необхідно розмістити базову інформацію згідно обраної тематики сайту (наприклад, історія клуба, біографія спортсмена, якому присвячено сайт).

***Завдання:***

1. Підібрати шаблон для майбутнього веб сайту відповідно до мокапів, наведених в завданні. Шаблон може бути будь-якого рівня складності. Головно вимога - наявність всіх елементів, що позначені в мокапах.
2. Зверстати головну сторінку сайту.

***Основні вимоги:***

1. Для створення головної сторінки використовувати тільки HTML5/CSS3, без будь-яких фреймворків.
2. Сторінка створюється з врахуванням коректного відображення тільки на десктопі
3. Підключення сторонніх шрифтів - за бажанням студента.
4. Сторінка повинна містити наступні частини:

       -       лівостороннє горизонтальне **фіксоване** меню (застосувати елемент <nav> та списки для створення меню)

       -       header

       -       content

       -        footer

       -       випадаюче меню при наведенні курсора на пункт «Matches» (застосувати псевдо-клас :hover)

       5. Допускається створення **НЕадаптивної** сторінки.

       Зберегти документ з назвою index.html.

**Task 2**

1. Зверстати ще одну додаткову сторінку веб-сайту (на вибір студента) та зробити перехід на неї з головної.
2. Додати на сторінку кілька паралельних анімацій. Наприклад: проявлення назви команди поверх картинки хедера + виїжджання логотипу.
3. Додати анімацію до пунктів меню. Анімація має спрацьовувати при наведенні курсора.

**Task 3**

Зверстати головну сторінку, fans admin, news використовуючи адаптивну верстку та Bootstrap framework.

***Основні вимоги:***

1. Для створення головної сторінки використати Bootstrap framework, будь-який інший зручний студентові фреймвок, що підтримує адаптивну верстку або ж чистий HTML5/CSS3 з Media Query.
2. Сторінка має бути адаптивною та виглядати коректно при будь-яких розширеннях екрану.
3. Підключення сторонніх шрифтів - за бажанням студента.
4. Сторінка повинна містити наступні частини:

-       горизонтальне або вертикальне меню, яке при малому розміру екрана складається у “гамбургер”

-       header

-       content

-       footer

-       випадаюче меню при наведенні курсора на пункт «Matches»

    5. Зберегти документ з назвою index.html.

**Task 4**

***Завдання:***

1. На сторінки “Fans” та “Admin” додати функції, які викликатимуться при додаванні відгука та новини відповідно.
2. При додаванні **відгука** функція має валідувати введені дані (поля не мають бути пустими), **додавати дату публікації** та публікувати відгук в списку відгуків на поточній сторінці. Після успішної публікації - очистити поля вводу без перезавантаження сторінки.
3. При додаванні **новини**, функція має валідувати введені користувачем дані, виділяти поля, дані яких не валідні і давати можливість користувачу це поправити, повідомляти користувача про успішну публікацію новини (якщо дані валідні) та очищати поля форми без перезавантаження сторінки.

**Task 5**

Додати можливість зберігання звернень та новин на стороні клієнта (використовуючи LocalStorage), а також їх вичитування з локальної бази.

**Завдання:**

1. Додати функцію, яка перевіряє наявність доступу до мережі.

**function isOnline() {**

**return window.navigator.onLine;**

**}**

**2.**Якщо функція повертає false - дані читаються та записуються в localStorage. Наприклад, нове звернення буде спочатку записуватись в localStorage, а вже потім додаватись на сторінку. При першому відкриванні сторінки має перевірятись наявність даних в localStorage та промальовування їх на сторінці.

**Нариклад: при переході на сторінку “News” має виконатись функція, яка перевірить localStorage на наявність новин та промалює їх на сторінці. Те ж саме зі зверненями.**

3. Якщо функція повертає  true, то дані мають відправлятись на сервер і читатись з сервера. В цьому випадку дозволяється додати функцію, яка емулюватиме відправку даних на сервер та читання з сервера.

4. При відкриванні сторінки має реєструватись Event Listener, що слухатиме зміну статуса мережі. Якщо користувач виходить онлайн, то на зміну статуса має спрацювати функція, що витягне дані з LocalStorage та відправить на сервер. Після публікації даних на сторінці вони видаляються із LocalStorage.

5. Для імплементації вище вказаної логіки використати лише **функціональний** підхід.

**Task 6**

Додати можливість зберігання звернень та новин на стороні клієнта (використовуючи LocalStorage та **IndexedDB**), а також їх вичитування з локальної бази.

***Завдання:***

1. Додати глобальну змінну useLocalStorage = true. За замовчування ця змінна матиме булеве значення true. В такому випадку всі дані читаються та записуються в localStorage.
2. Якщо useLocalStorage = false, то вище вказані дії браузер виконуватиме використовуючи IndexedDB.
3. Зміни стосуються **ТІЛЬКИ** роботи з локальними стореджами.

Для імплементації вище вказаної логіки використати **об”єктно-орієнтований** підхід.

**Task 7**

***Завдання:***

1. Додати підтримку сервера (NodeJS) та можливість вичитувати дані з сервера для сторінок “News” та “Fans” через REST.
2. Передавати дані на сервер через REST.
3. Використати БД MongoDB.
4. В окремому файлі описати кроки інсталяції та стартування веб сайту.