**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**"ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА"**

**Навчально-науковий інститут інформаційних технологій та робототехніки  
Кафедра комп'ютерних та інформаційних технологій і систем**

**Розрахунково-графічна робота**

**Варіант 14**

з навчальної дисципліни

“Веб дизайн”

**Виконав:**

                                                Студент групи: 103-ТК

                                                    Нонка Іван

**Керівник:**

                           Здоренко Юрій Миколайович

**Полтава**

**2023**

**Зміст**

1. Теоретичні відомості HTML 3

1.1 Основи HTML 3

1.2 Структура HTML – документа 4

1.3 Назва документа 5

1.4 Тіло документа 5

1. Теоретичні відомості CSS 6
2. Завдання курсової роботи 7
3. Виконання роботи 8
4. Висновок 18

**Теоретичні відомості HTML**

**1.Основи мови HTML**

HyperText Markup Language (HTML) - мова розмітки гіпертексту - призначена для написання гіпертекстових документів, що публікуються в World Wide Web.

Гіпертекстовий документ - це текстовий файл, що має спеціальні мітки, так звані теги, які згодом розпізнаються браузером і використовуються нам для відображення вмісту файлу на екрані комп'ютера. За допомогою цих позначок можна виділяти заголовки документа, змінювати колір, розмір і накреслення літер, вставляти графічні зображення і таблиці. Але основною перевагою гіпертексту перед звичайним текстом є можливість додавання до вмісту документа гіперпосилань - спеціальних конструкцій мови HTML, які дозволяють клацанням миші перейти до перегляду іншого документа.

Існує два способи створення гіпертекстових документів. Можна скористатися одним з HTML-редакторів, для роботи з якими не потрібно спеціальних знань про внутрішню структуру створюваного документа. Цей спосіб дозволяє створювати документи для WWW без знання мови HTML. HTML-редактори автоматизують створення гіпертекстових документів, позбавляють від рутинної роботи. Проте їхні можливості обмежені, вони сильно збільшують розмір одержуваного файлу і не завжди отриманий з їх допомогою результат відповідає очікуванням розробника.

Альтернативою є створення і розмітка документа за допомогою звичайного редактора plain-тексту (таких, як emacs або NotePad). При цьому способі в текст вручну вставляються команди мови HTML. Створюючи документи таким способом, ви точно знаєте, що робите.

Як вже зазначалося, HTML-документ містить символьну інформацію. Одна її частина - власне текст, тобто дані, що становлять вміст документа. Інша - теги (markup tags), так звані прапори розмітки, - спеціальні конструкції мови HTML, які використовуються для розмітки документа і керують його відображенням. Саме теги мови HTML визначають, у якому вигляді буде представлений текст, які його компоненти будуть виконувати роль гіпертекстових посилань, які графічні або мультимедійні об'єкти повинні бути включені в документ. Графічна та звукова інформація, включена в HTML-документ, зберігається в окремих файлах. Програми перегляду HTML-документів (браузери) інтерпретують прапори розмітки і відповідним чином відображають на екрані текст і графіку. Для файлів, що містять HTML-документи, прийняті розширення. Htm або. Html.

Прописні і малі літери при записі тегів не розрізняються. У більшості випадків теги використовуються парами. Пара складається з відкриваючого (Start tag) і закриваючого (end tag) тегів. Синтаксис відкриваючого тега:

*<Імя\_ тегу [атрибути]>*

Прямі дужки, які використовуються в описі синтаксису, означають, що даний елемент може бути відсутнім. Ім'я закриваючого тега відрізняється від імені відкриваючого лише тим, що перед ним ставиться похила риска:

*<Імя\_тега>*

Атрибути тега записуються в наступному форматі:

*ім'я [= "значення"]*

Лапки при заданні значення аргументу не обов'язкові і можуть бути опущені. Для деяких атрибутів значення може не вказуватися. У закриваючих тегів атрибутів не буває.

Дія будь-якого парного тега починається з того місця, де зустрівся відкриваючий тег і закінчується при зустрічі відповідного закриваючого тега. Часто пару, що складається з відкриваючого і закриваючого тегів, називають **контейнером**, а частина тексту, обмежену відкриваючим і закриваючим тегом, - **елементом**.

**2. Структура HTML-документа**

Найголовнішим з тегів HTML є однойменний тег - <html>. Він повинен завжди відкривати ваш документ, так само, як тег повинен неодмінно стояти в останній його рядку. Ці теги означають, що рядки, які знаходяться між ними представляють єдиний гіпертекстовий документ. Без цих тегів браузер або інша програма перегляду не в змозі ідентифікувати формат документа і правильно його інтерпретувати.

Закриваючий тег так само важливий, як і відкриваючий. Якщо, наприклад, документ включений в електронний лист, тег </HTML> дає команду програмі перегляду припинити інтерпретацію тексту, як HTML-коду.

HTML-документ складається з двох частин: заголовок (head) і тіло (body), розташованих в наступному порядку:

<HТМL>

<HЕАD>

...

</HЕАD>

<ВОDY>

...

</ВОDY>

</HТМL>

У HTML-документ можна включати коментарі, які дозволяють приховати частина тексту від браузера. Все, що укладенo між послідовностями символів <!-- і -->, при перегляді сторінки залишається невидимим. Коментарі не можуть бути вкладеними один в одного.

**3. Назва документу**

Включення в документ заголовної частини не є обов'язковим. Завданням заголовка є подання необхідної інформації для браузера та сервера http. Інформація, розміщена всередині заголовка документа, зазвичай не виводиться на екран (крім назви документа).

Назва документу відкривається тегом <HЕAD>, який звичайно розміщують відразу ж за тегом <HTML>. Закриваючий тег </HEAD> показує кінець цього розділу, між ними розташовуються інші теги заголовка документа.

Найчастіше в заголовок документа включають парний тег <TITLE>... </TITLE>, що визначає назву документа. Багато програм перегляду використовують його як заголовок вікна, в якому виводять документ. Програми, що індексують документи в мережі Інтернет, використовують назву для ідентифікації сторінки. Назва документа розміщується я між відкриваючим і закриваючим тегами.

**4. Тіло документа**

На відміну від заголовка, тіло документа є обов'язковим елементом, тому що в ньому розташовується весь матеріал вашого документа (є тільки одне виняток, з яким ми познайомимося далі, - коли документ містить замість тіла групу фреймів). Тіло документа розміщується між тегами <BODY> і </BODY>. Все, що розміщене між цими тегами, інтерпретується браузером відповідно до правил мови HTML, що дозволяють коректно відображати сторінку на екрані монітора.

Тег <BODY> не тільки позначає початок вмісту документа, але і задає його основні властивості: колір фону, тексту і багато інше. Ці властивості визначаються за допомогою атрибутів.

**Теоретичні відомості CSS**

**Каскадні таблиці стилів** (англ. Cascading Style Sheets) — спеціальна мова, що використовується для запису оформлення сторінок, написаних мовами розмітки даних.

Таблиці стилів дають змогу спростити процес створення сторінок і поліпшити їхній зовнішній вигляд. Концепція стилів подібна до ідеї стилів, яка реалізована в сучасних текстових редакторах — текст спочатку вводять, а потім форматують, користуючись стилями. Застосування стилів дає змогу вводити на сторінку потрібні тексти та інші елементи, не задумуючись над їхнім зовнішнім виглядом і розташуванням.

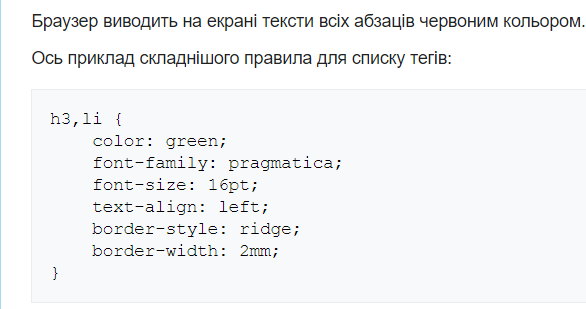
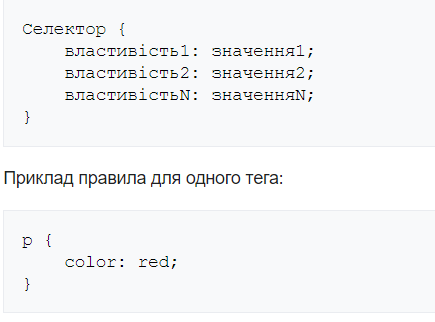
Таблиці стилів програміст зазвичай створює окремо від html-файлу. Під час створення html-файлу він концентрує увагу на змісті сторінки, а не на її зовнішньому вигляді, а під час створення таблиці стилів — навпаки. Отже, стилі дають змогу розмежувати етапи створення html-файлу й удосконалення зовнішнього вигляду сторінки.

Вважатимемо, що таблиця стилів уже створена. Тепер нам потрібно забезпечити взаємодію таблиці з html-файлом. Розглянемо три способи такої взаємодії: зв'язування, імпортування, вбудовування.

Таблиця стилів складається з правил, а правило — з назви тега чи списку назв тегів і описів стилів, які діятимуть у межах деякого html-файлу.

Опис стилю — це послідовність пар властивість:значення, які записують через крапку з комою та охоплюють фігурними дужками.

Отже, загальний вигляд правила такий:



Браузер виведе всі заголовки третього рівня h3 і елементи списків li зеленим кольором, шрифтом Прагматика розміру 16 пунктів, вирівняє їх до лівого краю вікна й охопить рамкою товщиною 2 мм з видавленим контуром (ridge).

**Завдання курсової роботи**

1. Створити сайт по варіанту заданої теми “Кафедра суспільних наук закладу вищої освіти” відповідно до вимог наданих в методичних вказівках:

**Загальні вимоги:**

Інформаційний портал повинен містити:

1. Головну сторінку
2. Шапку сайту(з логотипом)
3. Меню
4. Слайдер
5. Вивід 10 головних новин порталу
6. Підвал сайту(Футер)

Меню складається:

1. Про сайт
2. Галерея картинок
3. Новини
4. Контакти(з виведенням карти Google на сайт)

Дизайн повинен бути адаптивним (мобільна та десктопна версії)

**Виконання завдання**

1. Для початку створимо папку в якій буде зберігатись ввесь проект та необхідну ієрархія для подальшого виконання завдання.

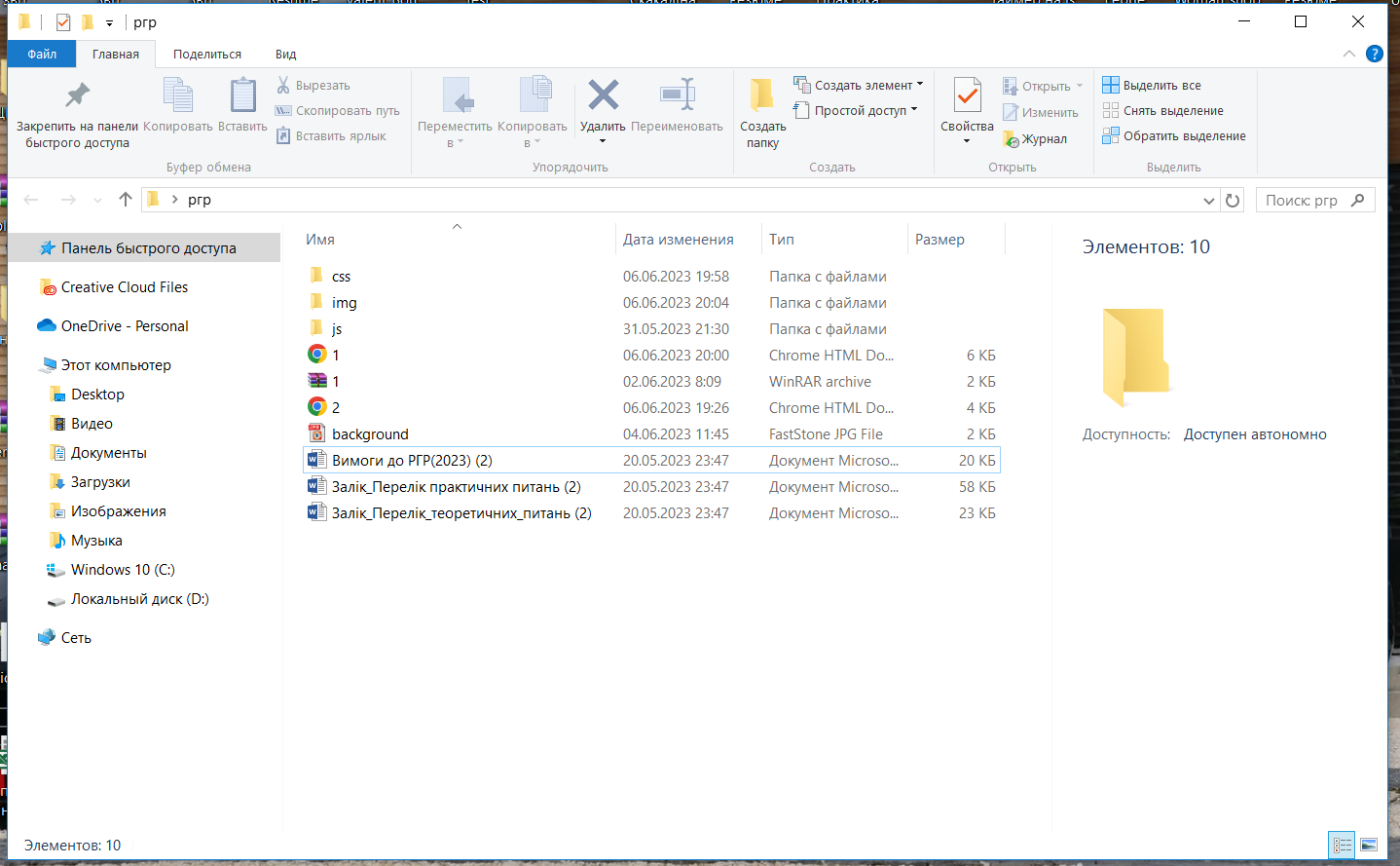


Рис.2 Папка з проектом.

1. Створимо основний документ з розширенням .html. (index.html). Для задання стилів створимо документ в папці css з розширенням .css. (style.css)
2. Для зручності створення проекту я використовував текстовий редактор Sublime Text.
3. Оцінивши складність роботи я розбив її на декілька частин. Нижче представлений код першої частини, яка включає в себе навігацію, заголовок і кнопку.

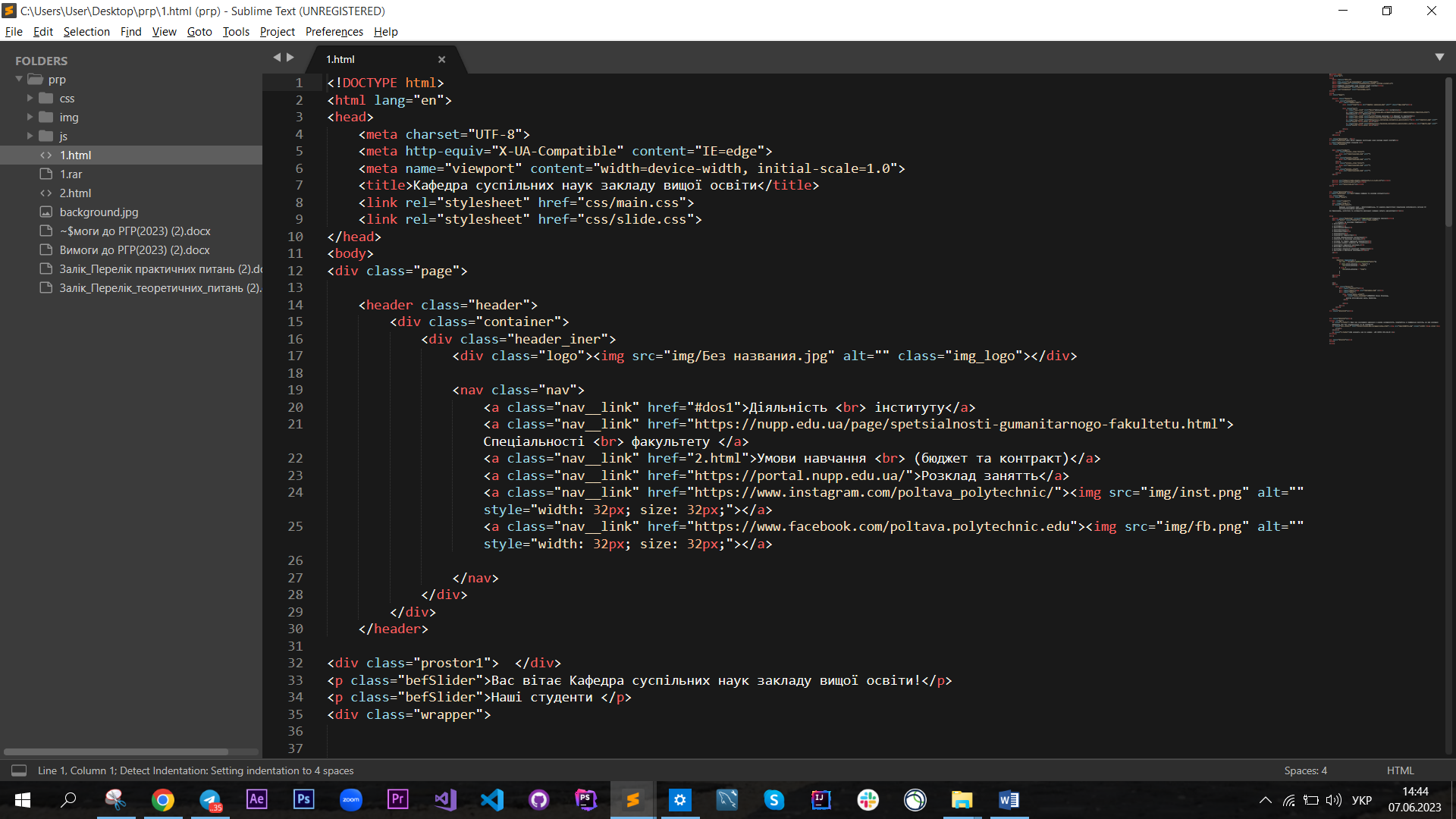


Рис.3 html код до першої частини сайту

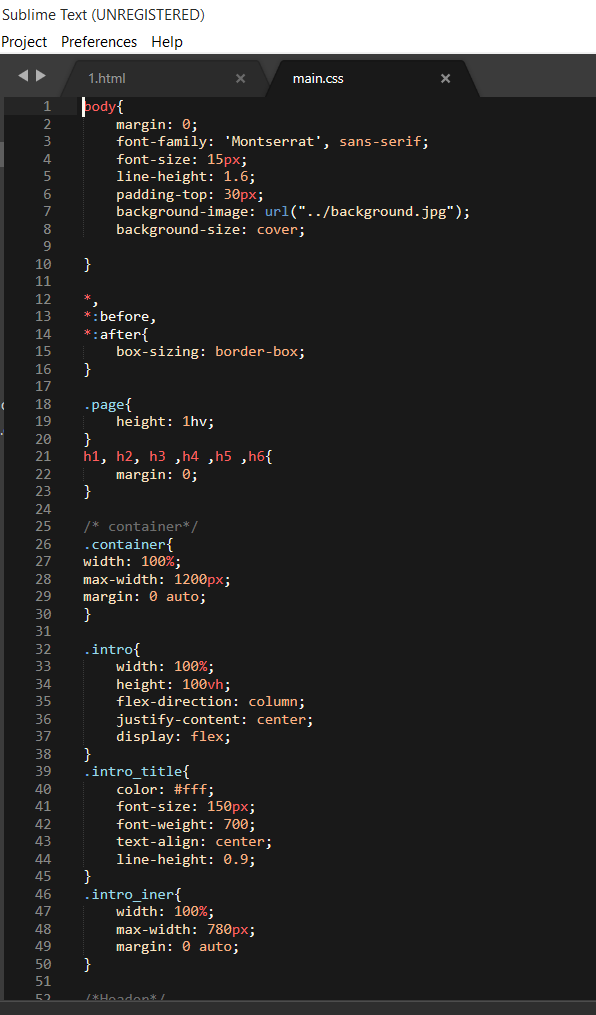
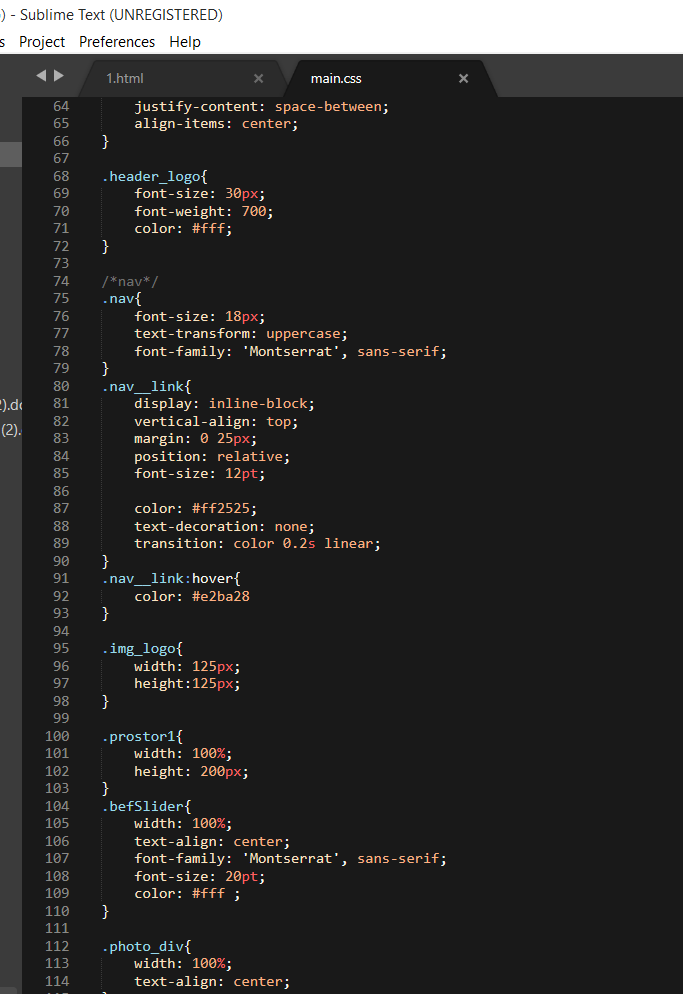
 

Рис. 4, 5 css код до першої частини

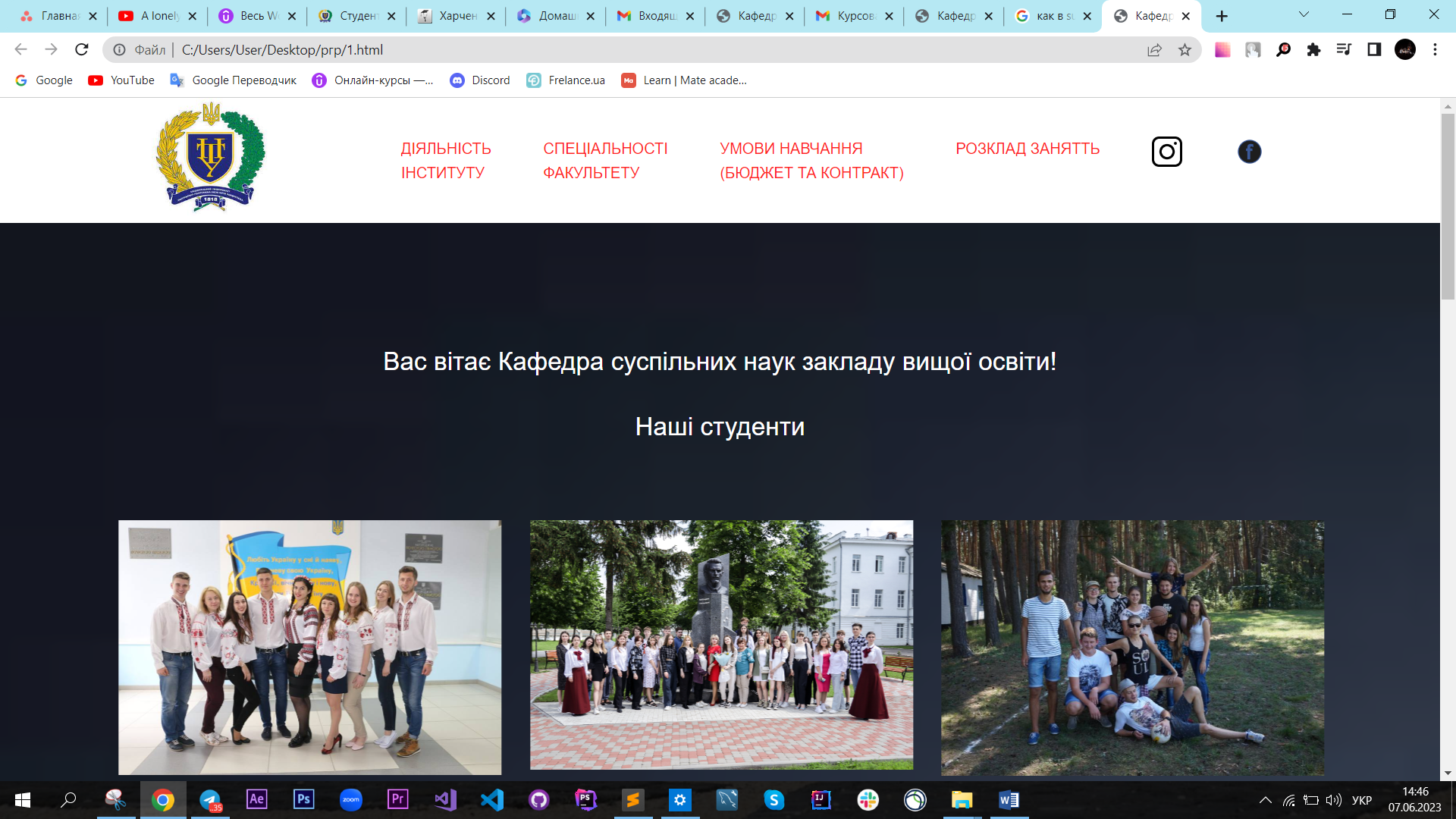


Рис.6 Зовнішній вигляд першої частини.

1. Продовжуємо створювати другу частину сайту.

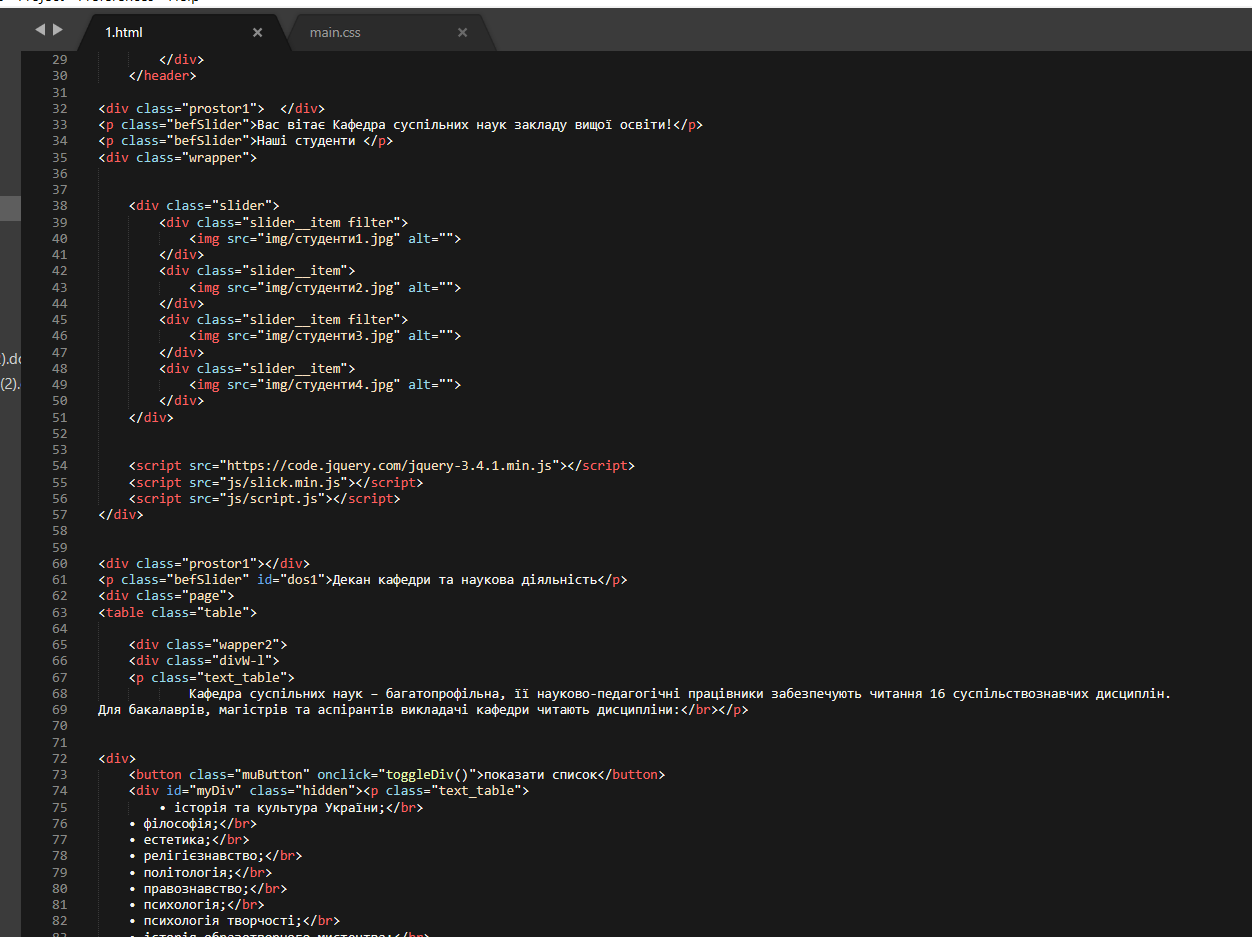


Рис. 7 html –код до другої частини

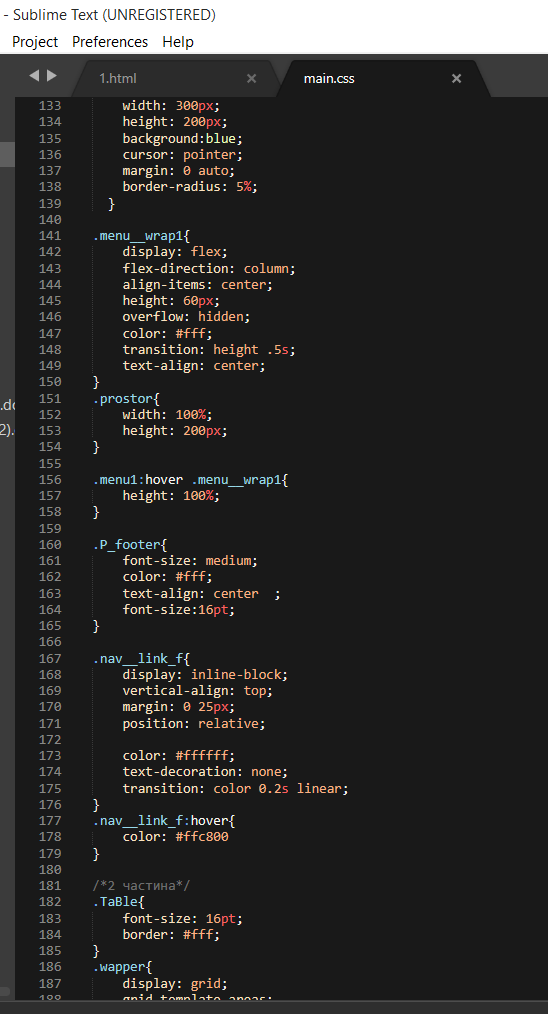
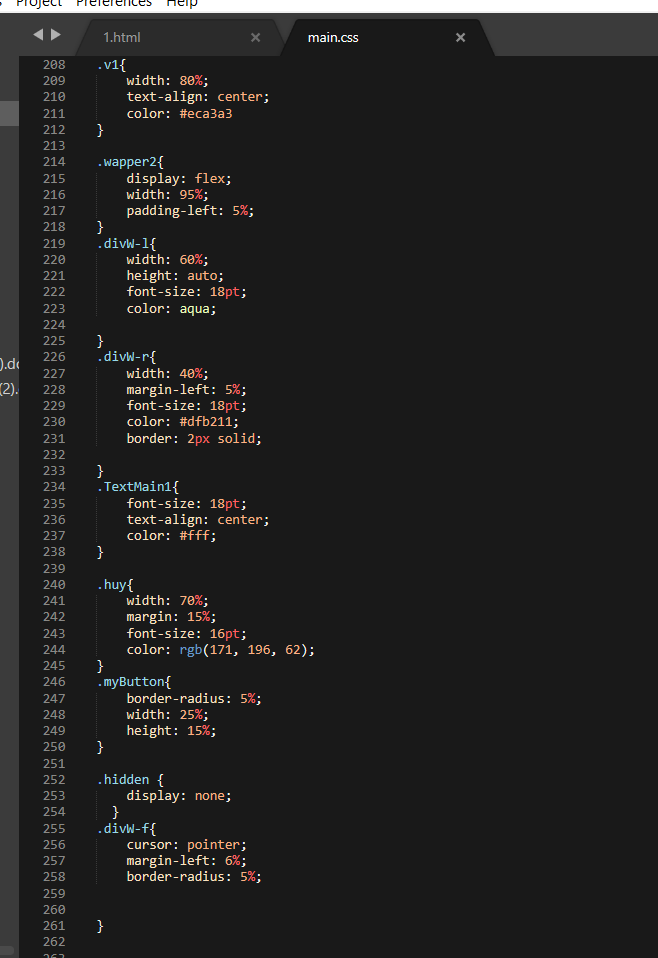
 

Рис. 8, 9 css код до другої частини

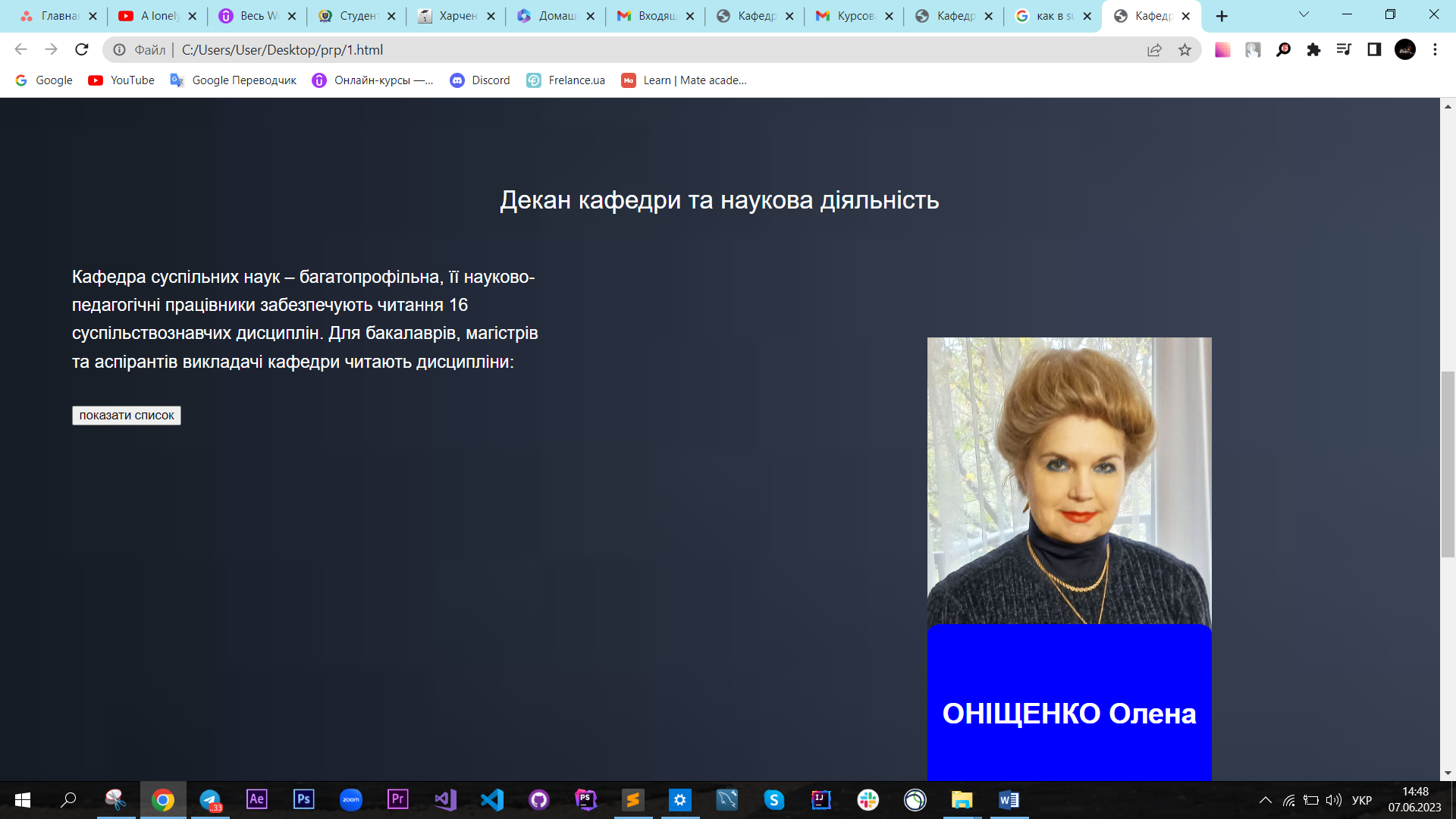


Рис. 10 зовнішній вигляд другої частини

1. Продовжуємо створювати сайт і переходимо до третьої частини.

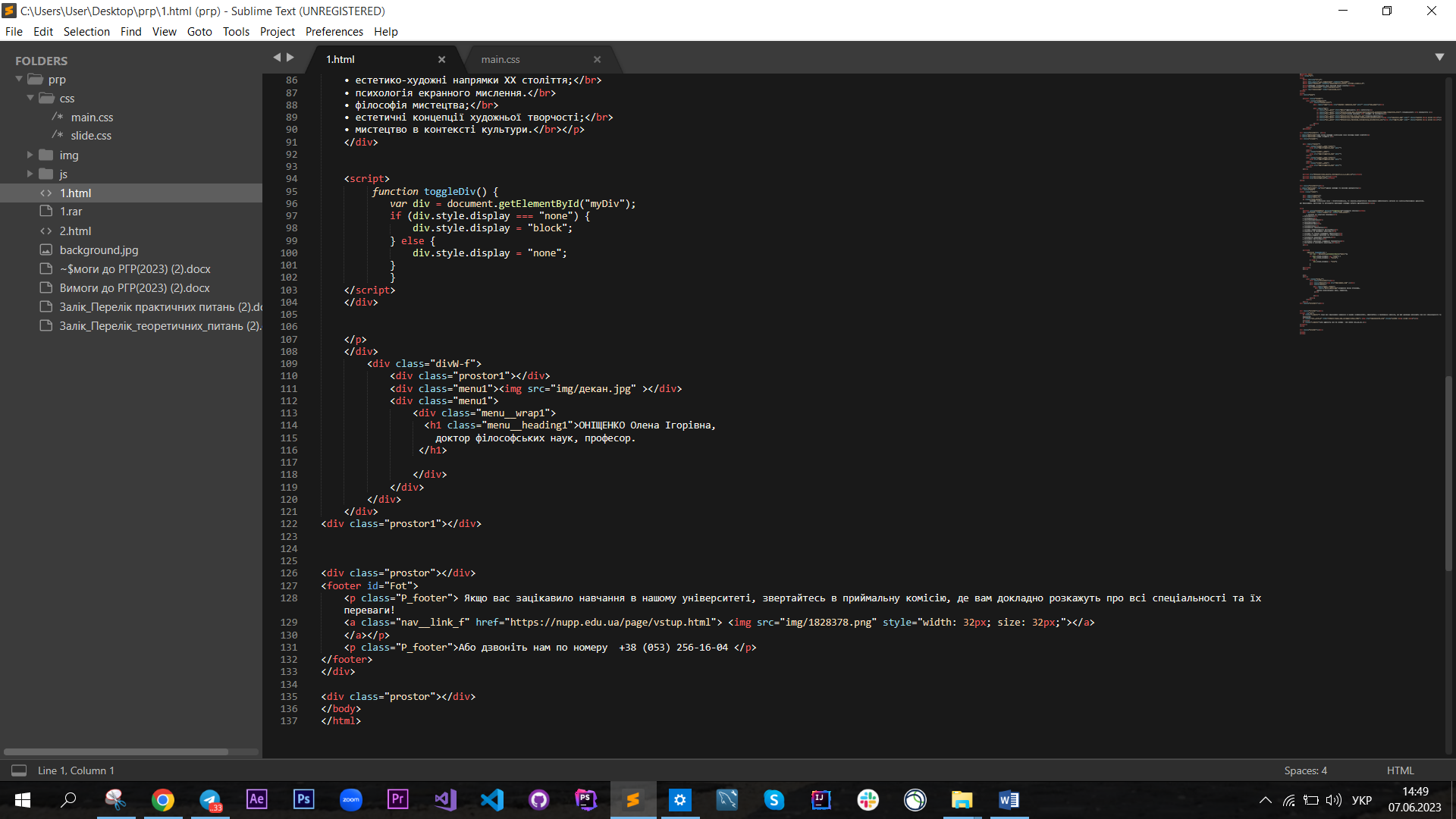


Рис. 11 html код до третьої частини сайту

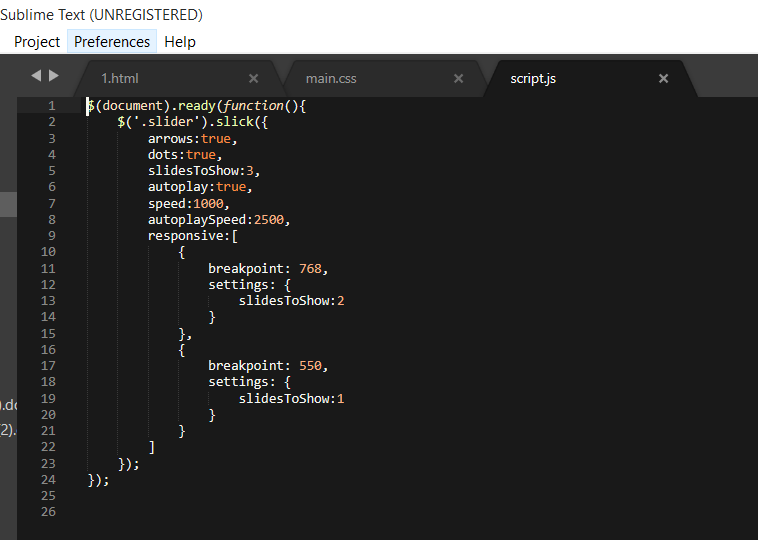


Рис 12 JS скрипт для роботи слайдера

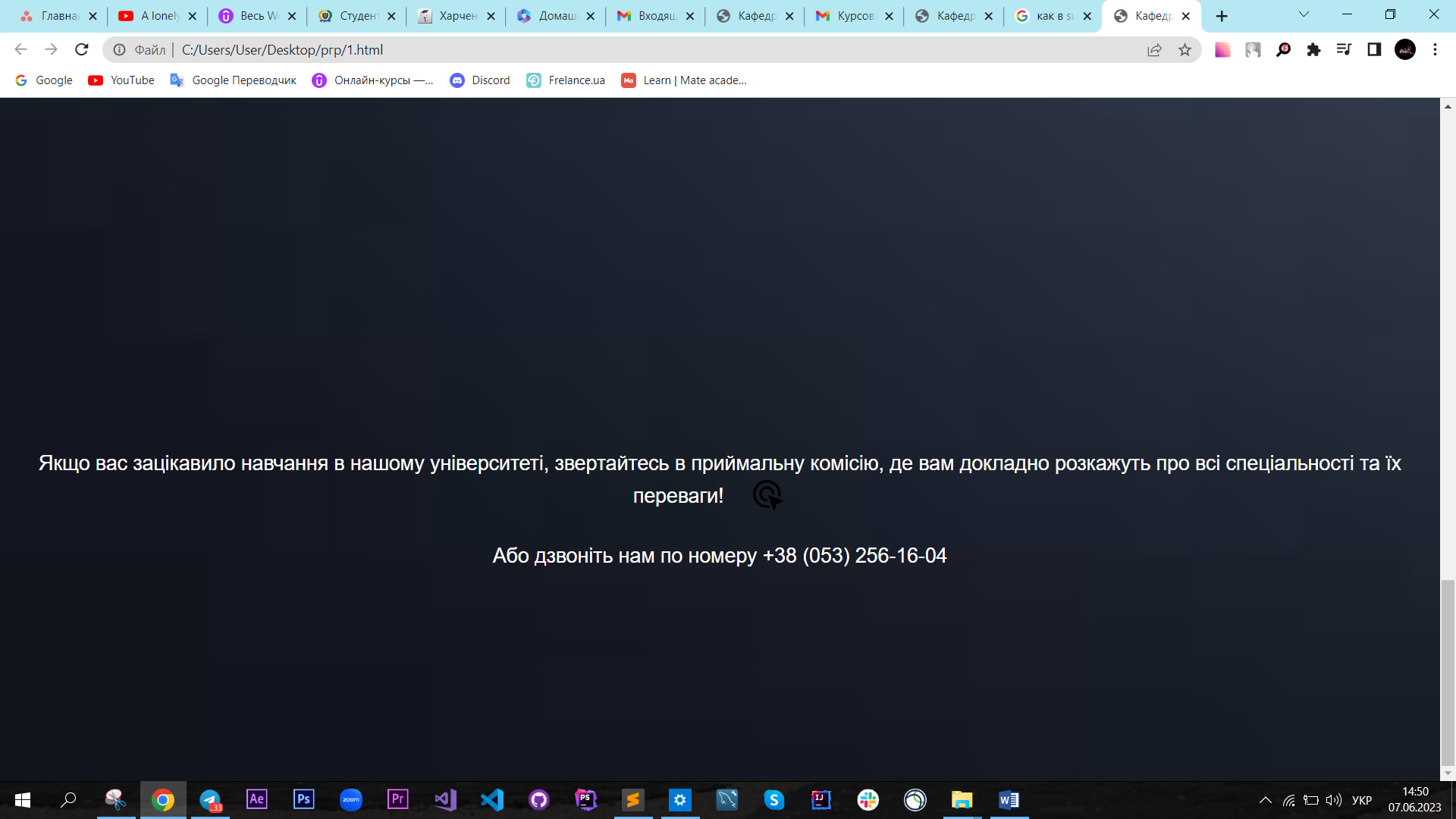


Рис. 14 зовнішній вигляд н

1. Переходимо до другої сторінки сайту

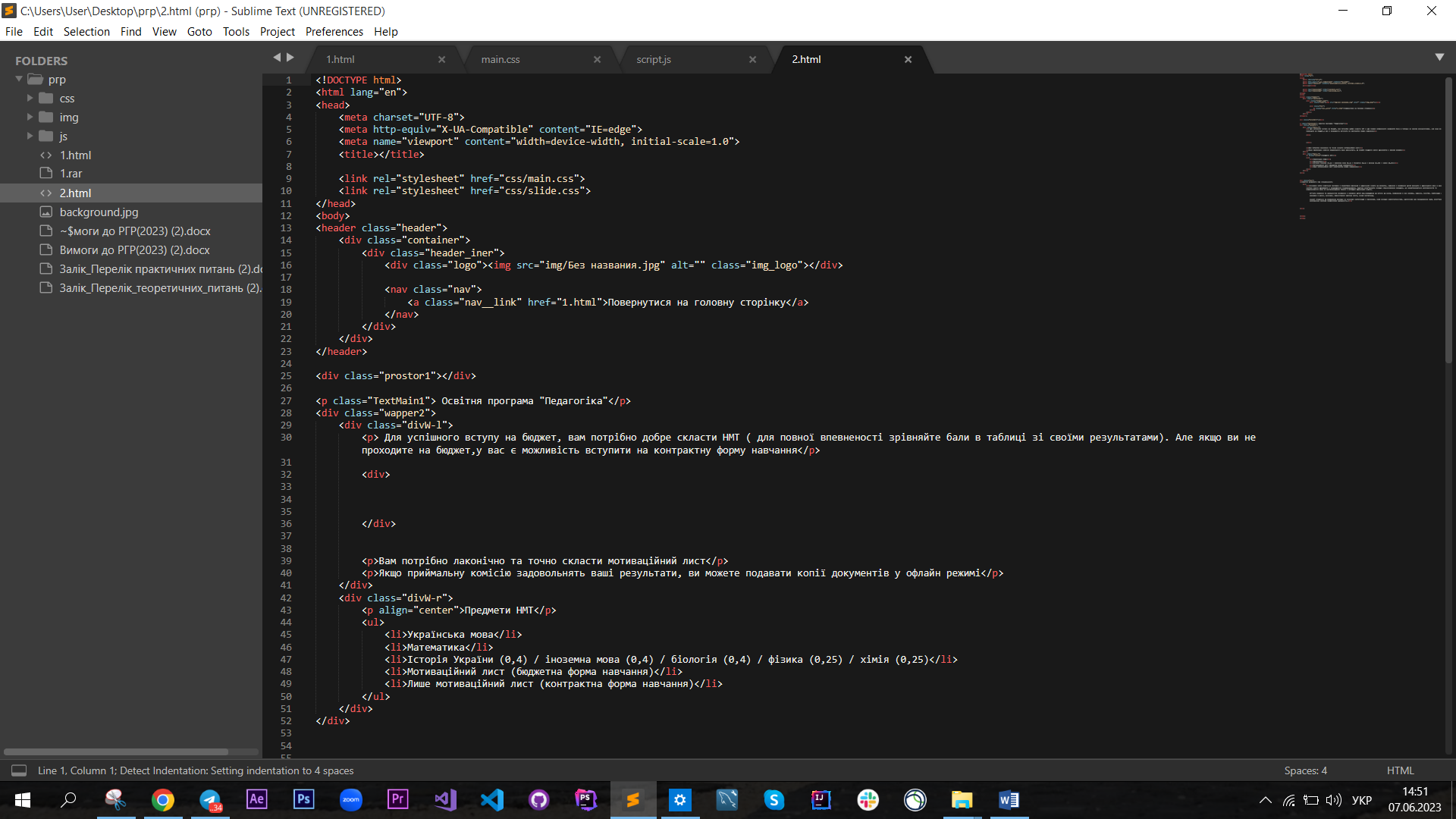


Рис. 15 html код до четвертої частини



Рис. 16 html код до другої сторінки сайту

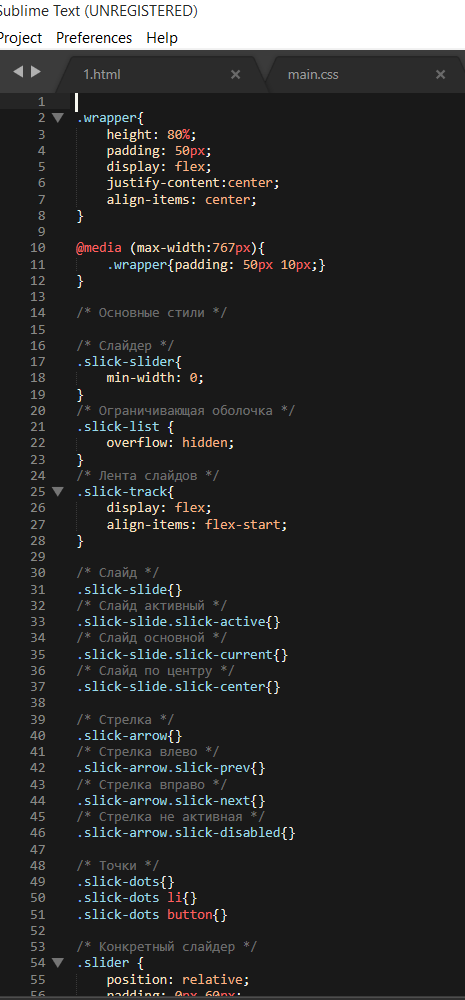
 

Рис. 17, 18 css код бібліотери слайдера

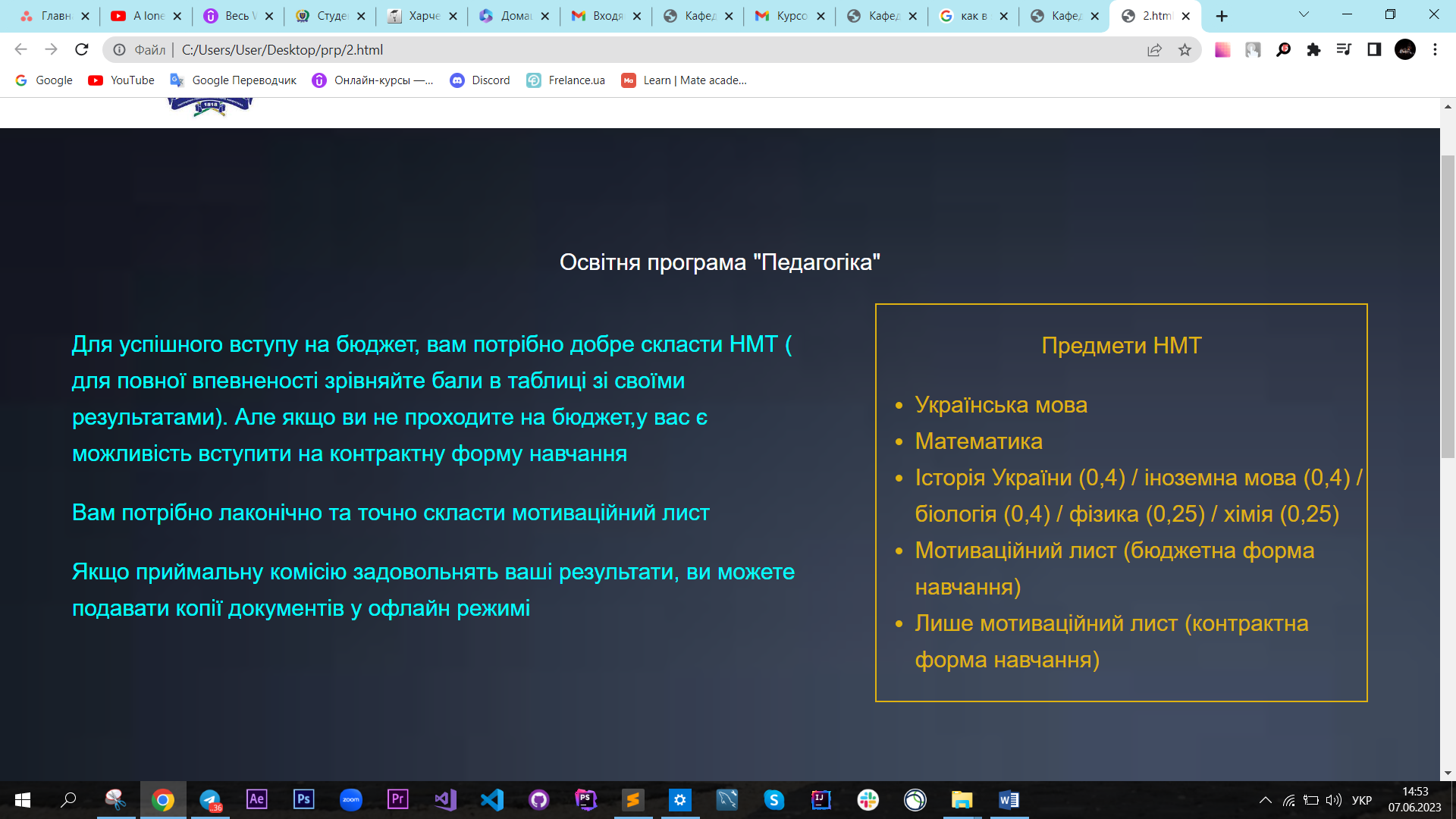


Рис. 19 зовнішній вигляд другої сторінки

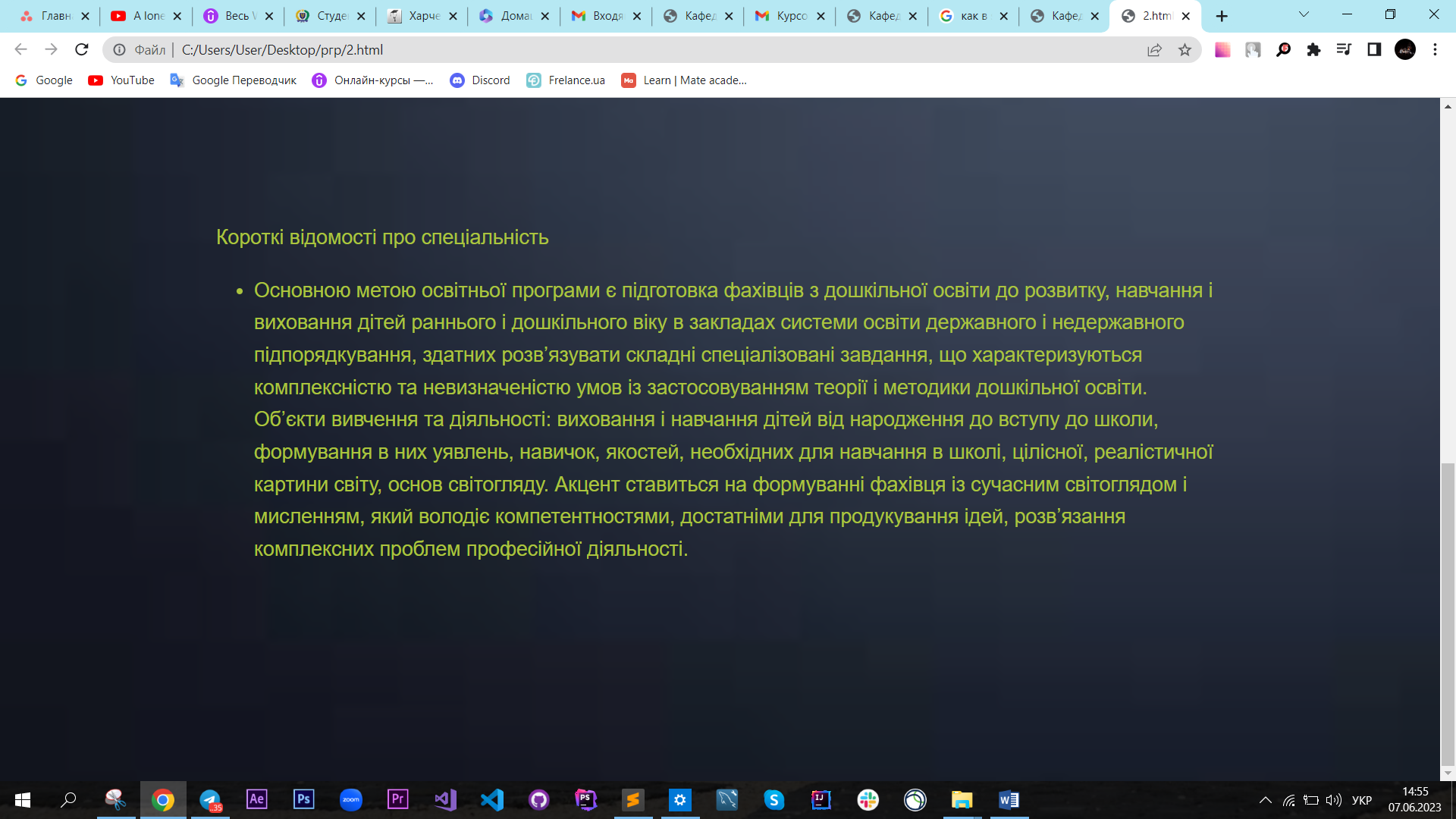


Рис. 20 зовнішній вигляд другої сторінки

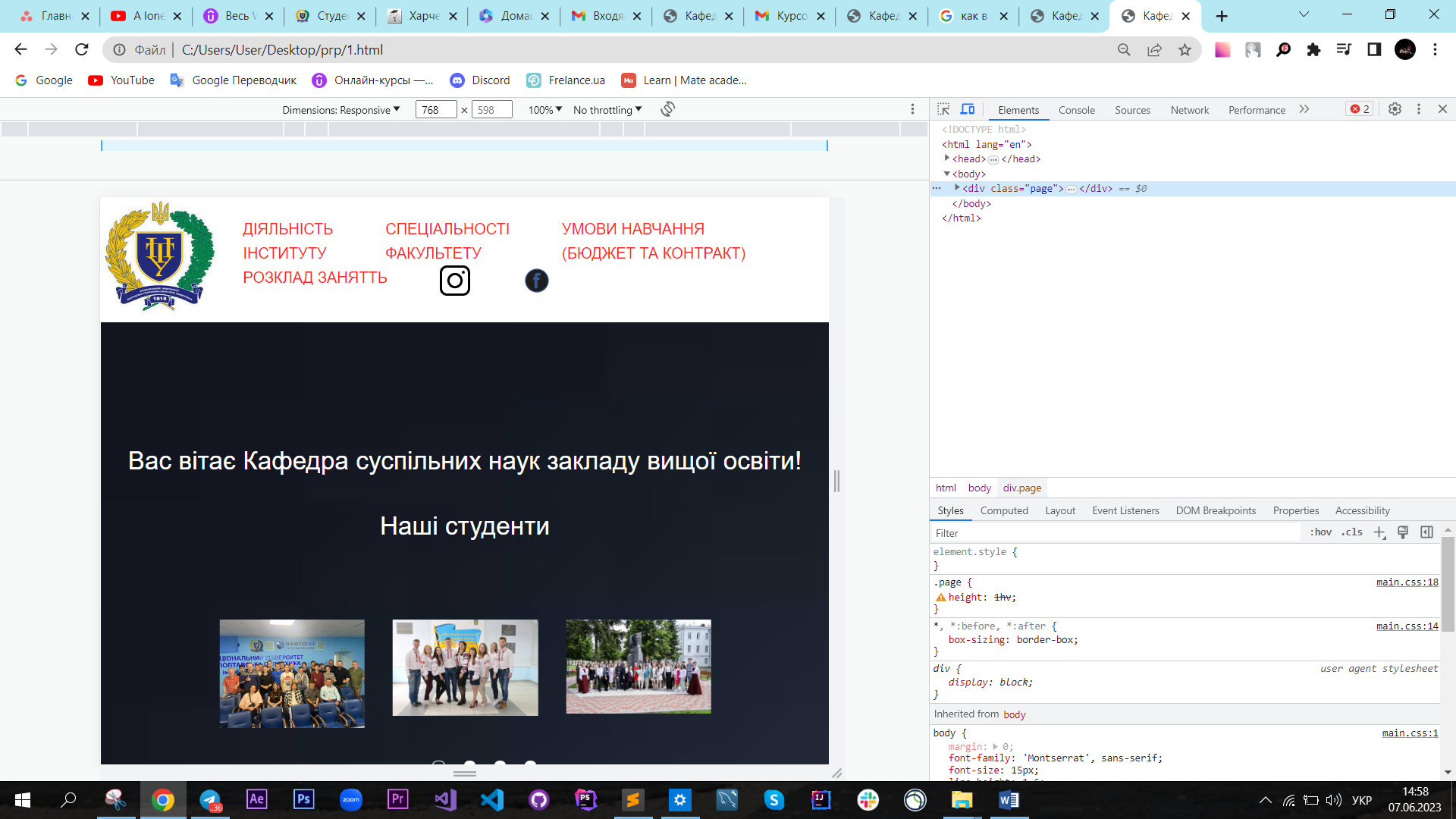


Рис. 21 зовнішній вигляд першої сторінки адаптація для планшетів

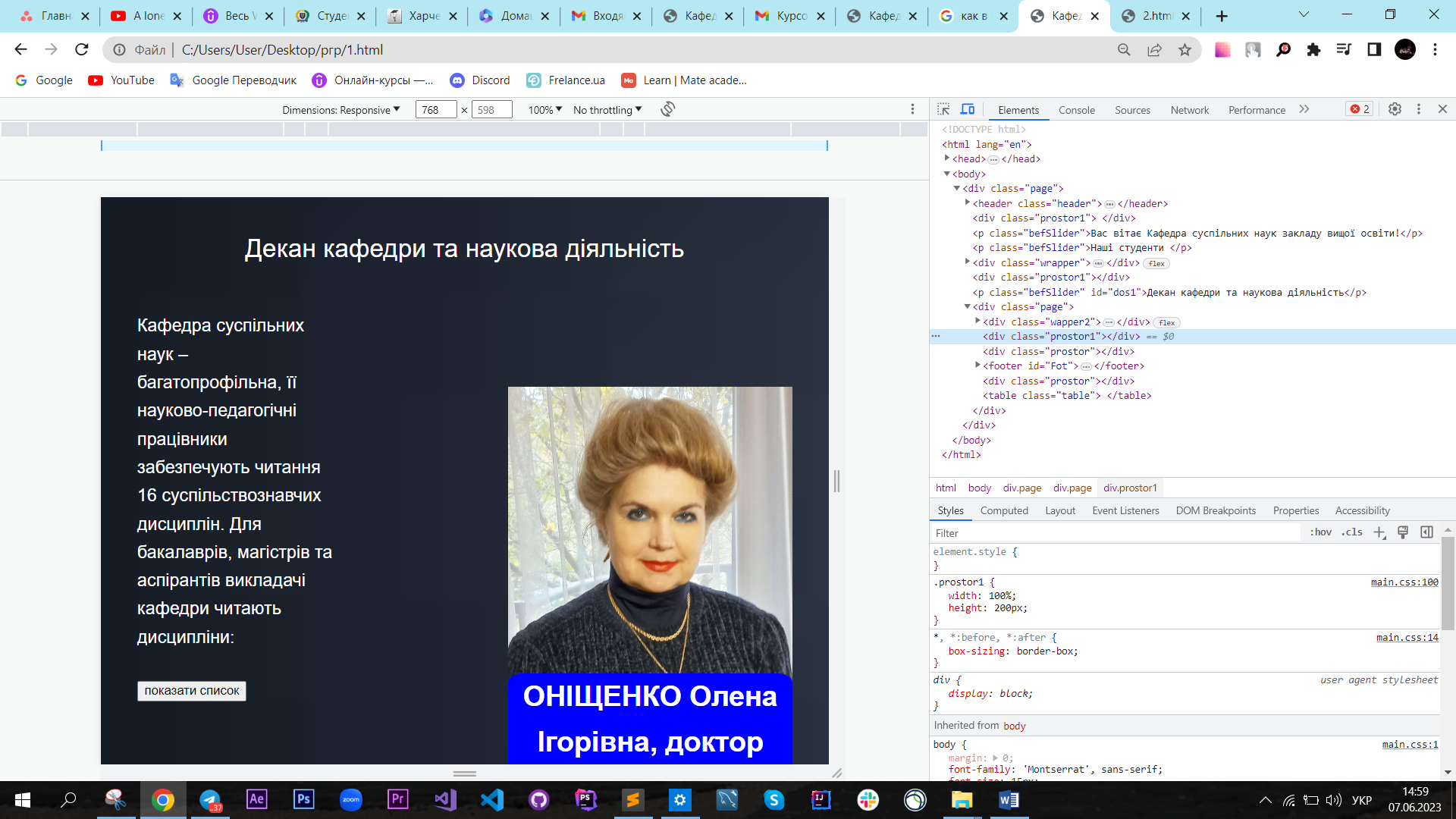


Рис. 22 зовнішній вигляд першої сторінки адаптація для планшетів

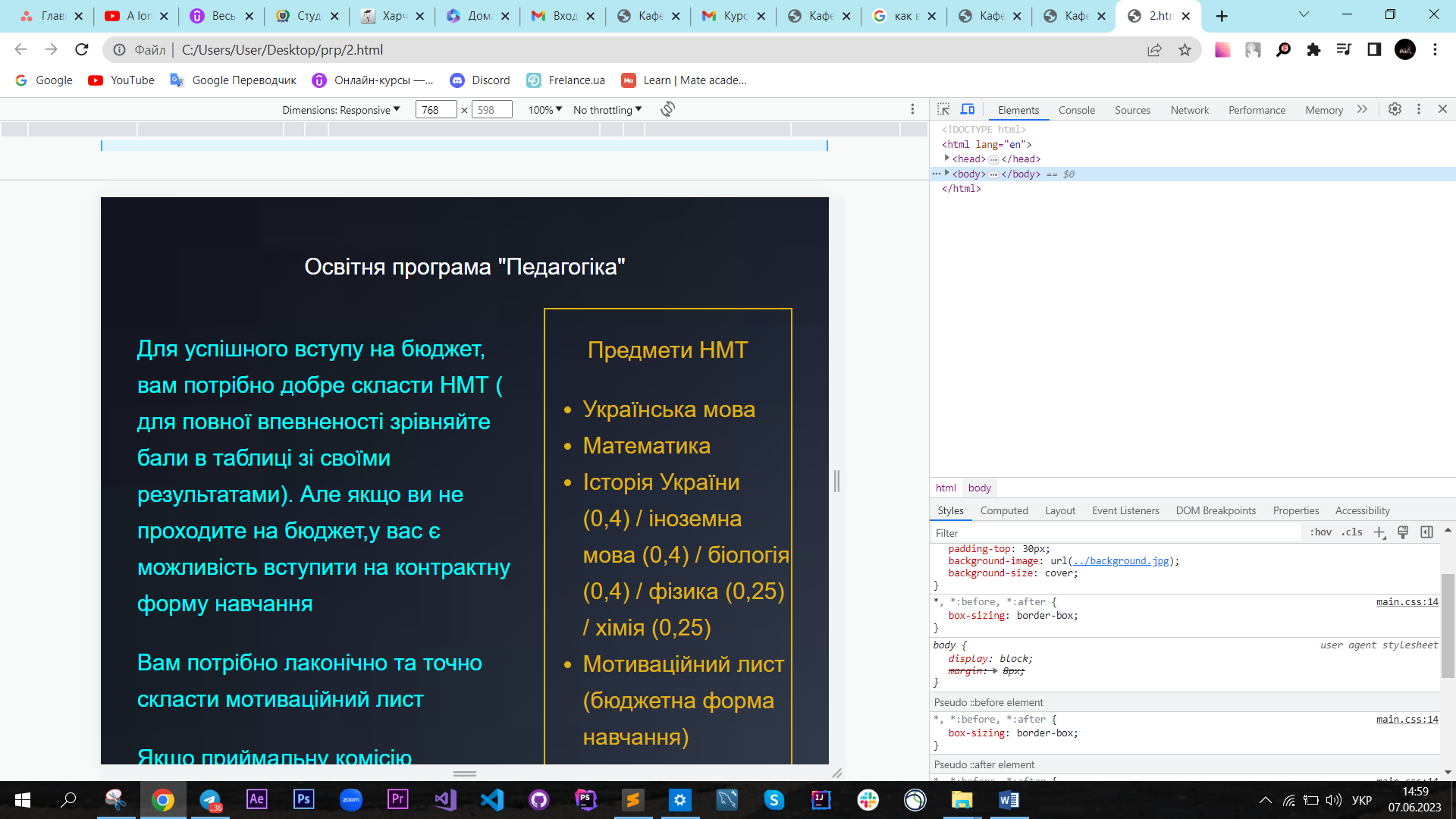


Рис. 23 зовнішній вигляд другої сторінки адаптація для планшетів

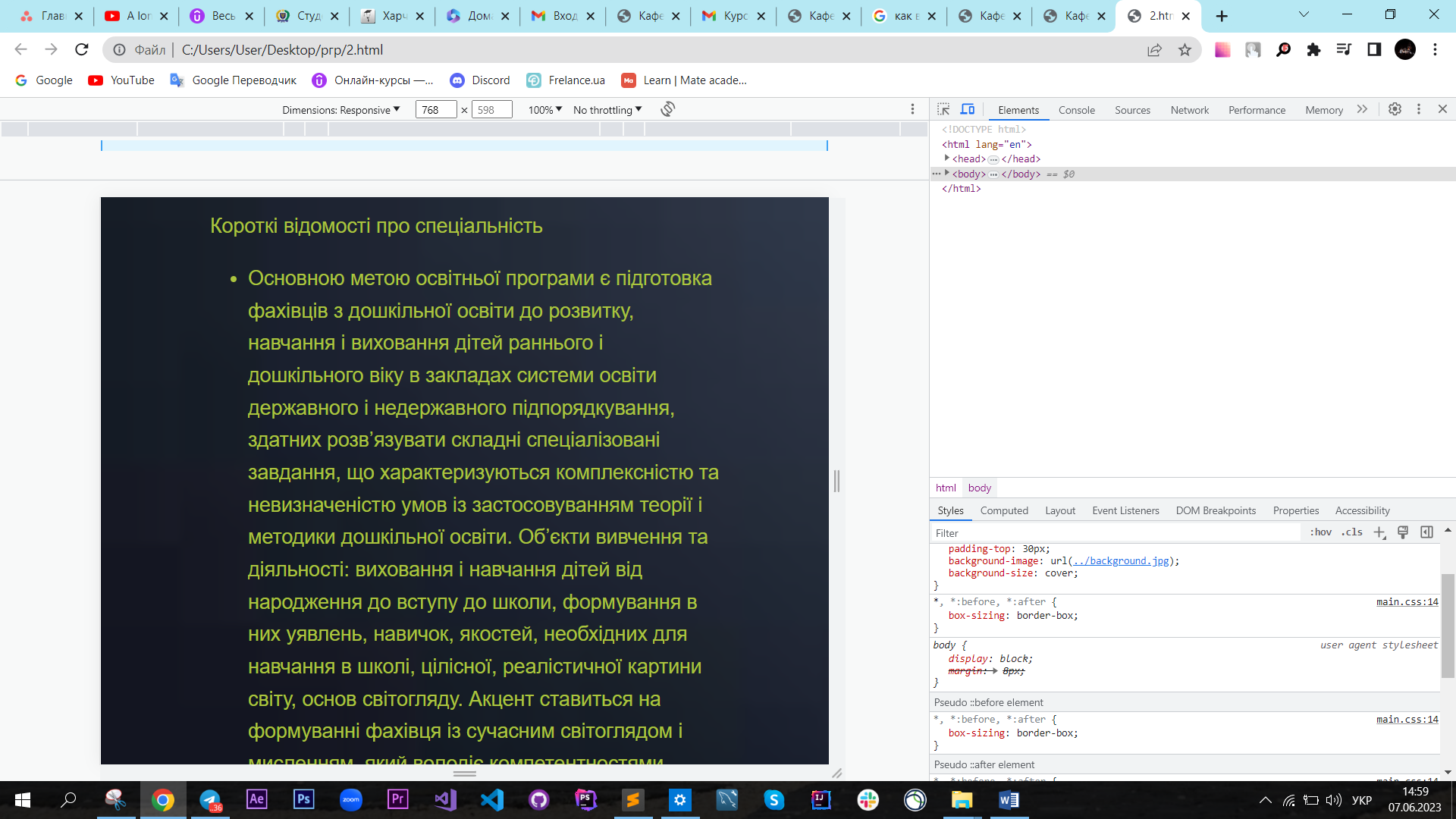


Рис. 24 зовнішній вигляд другої сторінки адаптація для планшетів

**Висновок**

В ході виконання курсової роботи було створено сайт відповідно до заданого макету. Вся робота виконувалась в текстовому редакторі Sublime Text. Для виконання роботи макет було поділено на дві сторінки та 4 блоки Кожна частина підписана в коді відповідними коментарями. В загальному було створено слайдер за допомогою бібліотери Slick та декілька функціональних блоків, що включали в себе декілька силок які переадресовують на відповідні сторінки. Певні елементи ( логотипи ) були взяті з сайту нашого університету. Проект складається з головного документу index.html , style.css в папці css. Та js файлу, що забезпечує роботу слайдера на головній сторінці Зображення та логотипи знаходяться в папці img.