

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

**Пояснювальна записка**  
**до дипломної роботи**  
**бакалавр**

(освітньо-кваліфікаційний рівень)

**на тему «Розробка інтерактивного сайту для гуртожитку»**

Виконав: студент 4 курсу, групи ІПЗ-17д  
напряму підготовки 121 „Інженерія програмного  
забезпечення”

\_\_\_\_\_ Рудак М.Ф.  
(підпис)

Керівник,  
доцент, к.т.н. \_\_\_\_\_ Іванов В.Г.  
(підпис)

Рецензент,  
доцент каф. ПМ,  
ст. викл. \_\_\_\_\_ Батурін О.І.  
(підпис)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ  
бакалаврської роботи студента гр. ПЗ-17д Рудака М.Ф.

Науковий керівник

доцент, к.т.н. \_\_\_\_\_ Іванов .В.Г  
ПІБ, посада

Оцінка наукового керівника: \_\_\_\_\_

Рецензент Батурін О.І., ст. викл. каф. ПМ СНУ ім. В. Даля  
ПІБ, місто роботи, посада

Оцінка рецензента: \_\_\_\_\_

Кінцева оцінка за результатами захисту: \_\_\_\_\_

Голова ЕК,  
проф. кафедри ПМ  
к.т.н., доцент

\_\_\_\_\_  
підпис

Захожай О.І.

**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ**

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 121 „Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

\_\_\_\_\_ Лифар В.О.

«\_\_\_» \_\_\_\_\_ 2021 р.

**З А В Д А Н Н Я**

**НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Рудака Микити Федоровича

**1. Тема роботи Розробка інтерактивного сайту для гуртожитку.**

**керівник роботи к.т.н., доцент Іванов Віталій Геннадійович**

затверджені наказом вищого навчального закладу від «22» квітня 2021 року №  
68/14.04

2. Строк подання студентом роботи 12 червня 2021 р.

3. Вихідні дані до роботи

Об'єктом досліджень є розробка інтерактивного сайту гуртожитку.

3.1 Літературні джерела:

Переваги побудови динамічних сайтів - [hackernoon.com](https://hackernoon.com/the-advantages-ofbuilding-dynamic-websites-bc914071a155) [Електронний ресурс] - Режим доступу: <https://hackernoon.com/the-advantages-ofbuilding-dynamic-websites-bc914071a155>

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ

4.2 Аналіз предметної галузі (огляд літератури), з висвітленням наступних питань:

Основні завдання гуртожитку.

Аналіз конкурентів.

4.3 Основна частина, в якій висвітлити:

Вибір і опис методології проектування.

Технології розробки сайту.

Вибір засобів розробки.

Проектування схеми і опис бази даних.

Вибір СУБД.

Проектування макету сайту

Реалізація веб сайту.

4.4 Висновки

4.4 Перелік використаних джерел

5. Перелік графічного матеріалу немає

6. Дата видачі завдання 23 квітня 2021 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	23.04.21	
2	Укладання і погодження з керівником плану і етапів виконання роботи	23.04.21	
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	26.04.21	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	02.05.21	
5	Розробка технічного завдання	09.05. 21	
6	Укладання програмного продукту	11.05.21	
7	Укладання, оформлення та погодження пояснювальної записки з керівником	25.05.21	
8	Здача готової пояснювальної записки на кафедрі	06.06.21	
9	Укладання доповіді і презентації	07.06.21	

Студент

\_\_\_\_\_ Рудак М. Ф.  
(підпис)

Керівник роботи

\_\_\_\_\_ Іванов В. Г.  
(підпис)

## РЕФЕРАТ

Текст – 51 с., рис. – 21, табл. – 0, додатків – 1, літературних джерел – 19

У ході виконання дипломної роботи були розглянуті актуальність розробки сайту, існуючі рішення та їх недоліки, висунуті вимоги до розробки макету майбутнього сайту та серверної частини. Була освітлена клієнт-серверна архітектура, побудована ER-діаграма, спроектована база даних та проаналізовані технології реалізації серверу. Була розроблена база даних, та сформовані механізми взаємодії з клієнтом у вигляді форми реєстрації.

Кінцевим результатом є розроблений сайт гуртожитку. Всі поставлені задачі виконані та всі цілі досягнуті.

Ключові слова: МОДЕЛЬ, СУТНІСТЬ, САЙТ, Open Server, Visual Studio Code.

## ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

БД – база даних.

VS Code - Visual Studio Code

HTML – HyperText Markup Language, мова гіпертекстової розмітки.

JSON – JavaScript Object Notation, об'єктна нотація JavaScript.

XML – Extensible Markup Language, розширювана мова розмітки.

URL - Uniform Resource Locator, уніфікований покажчик інформаційного ресурсу.

API – Application Programming Interface, інтерфейс прикладного програмування.

HTTP – HyperText Transfer Protocol, протокол передачі HyperText.

## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ І ЗАСОБІВ РОЗРОБКИ ПРОЕКТУ .....	11
1.1 Основні завдання гуртожитку:.....	13
1.2 Аналіз конкурентів .....	14
РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ .....	16
2.1 Вибір і опис методології проектування .....	16
2.2 Технології розробки сайту.....	17
2.3 Вибір засобів розробки.....	19
2.3 Проектування схеми і опис бази даних.....	26
2.3.1 Таблиця «users».....	26
2.3.2 Таблиця «news».....	27
2.4 Вибір СУБД.....	28
2.5 Проектування макета сайту .....	30
2.6 Логотип.....	33
2.7 Реєстраційна форма .....	34
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ САЙТУ .....	36
3.1 Обґрунтування вибору інструментальних засобів .....	36
3.2 Опис призначеного для користувача інтерфейсу.....	39
3.3 Реалізація клієнтської частини.....	43
3.3.1 Реалізація реєстрації користувача.....	44
ВИСНОВКИ .....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	50



## ВСТУП

### Актуальність досліджень.

В даний час Інтернет стає все більш розвиненим середовищем для спілкування. У зв'язку з глобальним розвитком Інтернету окрема галузь веб-програмування стає все більш помітною у програмуванні.

Тепер, щоб привернути увагу, вам просто потрібно заявити про себе в Інтернеті, створивши веб-сайт. Для цих цілей він просто служить веб-сайтом, що містить головну інформацію про проект також контактну інформацію. Сайти використовують для зберігання, передачі, надання різних видів інформації, не виходячи з дому.

Створення і вдосконалення комп'ютерів призвело і продовжує призводити до створення нових технологій в різних сферах наукової та практичної діяльності. Одне з найважливіших досягнень людства є Інтернет, здатний зв'язувати мільйони комп'ютерів по всьому світу. На сьогоднішній день, Інтернет містить близько 612 мільйонів сайтів, що містить інформацію в усіх областях.

Всесвітня павутина - це глобальна комп'ютерна мережа, яка сьогодні містить мільйони веб-сайтів, на яких розміщується різноманітна інформація. Люди мають доступ до цієї інформації через Інтернет-технології. Для пошуку в Інтернеті використовуйте спеціальні програми - веб-браузери, які значно полегшують подорож у нескінченних обр'ях Інтернету. Буде розглянуто питання створення сайту гуртожитку. Ця тема актуальна, оскільки зараз все більше людей перебувають у мережі. І створення такого сайту сприятиме підвищенню уваги до гуртожитку.

Сайт сьогодні є не лише інформаційним інструментом чи візитною карткою, а повноцінним маркетинговим інструментом, який залучає нових студентів.

### Завдання дипломної роботи:

- розробка сайту за допомогою сучасного програмного забезпечення;

- розробка дизайну, який робить сайт більш привабливим і зрозумілим для відвідувачів сайту;
- розробка більш зручного інтерфейсу, що дозволяє зручно проводити час на сайті;

Користуючись сайтом, ви можете:

- надіслати всю необхідну інформацію про гуртожиток та спеціальності, включаючи останні новини;
- встановити прямий контакт з кандидатами та студентами (наприклад, використання флеш-опитувань);

Якісний веб-сайт є основним інформаційним ресурсом для всього гуртожитку. Таким чином, метою дипломного проекту є розробка сайту для гуртожитку №1 "Східноукраїнського національного університету Імені Володимира Даля".

Цільова аудиторія: студенти денної та заочної форм навчання цього навчального закладу, кандидати, студенти.

Мета роботи: створення веб-сайту, заповнення розділів веб-сайту навчального закладу гуртожитку «Східноукраїнського національного університету імені Володимира Даля»

Об'єкт дослідження: розробка веб-сайту.

Предмет дослідження: зміст розділів веб-сайту навчального закладу.

Цілі дослідження:

1. Типи та структура веб-сайтів.
2. Проведення порівняльного аналізу веб-сайтів навчальних закладів.
3. Розробити дизайн веб-сайту, розділів веб-сайту навчального закладу.
4. Створити веб-сайт, розділи сайту з урахуванням розробленого дизайну.

## РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ І ЗАСОБІВ РОЗРОБКИ ПРОЕКТУ

Інтернет - це всесвітня мережа безлічі комп'ютерних мереж різного рівня та окремих комп'ютерів, що обмінюються інформацією між собою за допомогою загальнодоступних телекомунікаційних каналів на основі протоколів зв'язку TCP / IP. Інформація в Інтернеті зберігається на серверах. Сервери, з'єднані високошвидкісними магістралями, становлять основну частину Інтернету. Користувачі зазвичай отримують доступ до інформаційних ресурсів Інтернету через провайдерів або корпоративну мережу. Сьогодні кількість користувачів Інтернету швидко зростає. Станом на червень 2021 року кількість користувачів, які відвідують Інтернет, становить 5 мільярда чоловік.

В Інтернеті є кілька служб (електронна пошта, USENET, TELNET, WWW, FTP тощо), але найпопулярнішою послугою є WWW. Служби WWW будуються за принципом клієнт-сервер. Послуга складається з серверів, доступ до яких здійснюється через клієнтські програми або браузері. Основна маса інформаційних ресурсів у вигляді веб-сторінок або html-файлів знаходиться на веб-сайтах, розташованих на Web-серверах (на хостах) в Інтернеті.

Студентський гуртожиток - це будинок у вищих навчальних закладах, призначений для проживання іноземних студентів, слухачів, курсантів, молодших слухачів, слухачів, клінічних ординаторів, аспірантів, юристів, докторантів тощо на період їх навчання. (Далі - студенти), а також студенти-сироти та діти, позбавлені батьківського піклування, за умови, що вони не мають іншого місця проживання. Іноземні громадяни, які навчаються в університеті, розміщуються в житловому приміщенні на загальних умовах, якщо інше не встановлено договором або іншими нормативними актами. Безкоштовні кімнати в гуртожитках керівник вищого навчального закладу може використовувати за погодженням із студентським самоврядуванням та первинною професійною організацією студентів для створення окремого готелю. Гуртожиток може приймати іноземних студентів-заочників на період екзаменаційної сесії, іноземних учасників на період вступних

іспитів, а також кандидатів від дітей-сиріт, за умови, що вони не мають іншого місця проживання.

Розміщення студентів у студентських гуртожитках здійснюється за рішенням керівника вищого навчального закладу за погодженням із студентським самоврядуванням та первинною профспілковою організацією студентів на підставі наказів відповідно до Житлового кодексу України. У разі порушення законодавства укладається договір (контракт) на проживання.

Житлові приміщення спеціально побудовані або перебудовані для цих цілей у приміщеннях. У підвалах, на схилах пагорбів та в підвалах забороняється проживання студентів. Студентські гуртожитки можна створити в орендованих будинках (кімнатах), прикріплених до гуртожитку.

Гуртожитки можуть бути:

- для фізичного життя (житлові одиниці спільно використовуються кількома людьми і не перебувають у сімейних стосунках).
- для сімей студентів (житлові приміщення, які складаються з однієї або декількох кімнат, використовуються однією сім'єю).
- проживання для студентів, які проживають у помешканні, якщо це необхідно, можна забезпечити в кімнатах для проживання фізичних людей. Ці кімнати повинні розташовуватися в загальних секціях витяжки.

Гуртожитки можна розділити на кімнати для чоловіків, жінок, квартири, пансіонати чи готелі, із самообслуговуванням та самоврядування, залежно від умов проживання, визначених керівником вищого навчального закладу та узгоджених зі студентською радою та об'єднання учнів початкових класів.

Житловий заклад повинен забезпечувати необхідні умови для проживання, самостійного навчання, дозвілля, фізичного виховання та виховної роботи. Санітарні приміщення розподіляються та обладнуються відповідно до діючих санітарних норм та правил будівлі[19].

Розміщення та розміщення студентів у гуртожитках, умови проживання, права та обов'язки регулюються "Положенням про гуртожитки", розробленим у

вищих навчальних закладах на основі законодавства України та погодженим із студентським самоврядуванням (уряд та профспілка учнів початкових класів).

У сім'ях учнів з дітьми повинні бути місця для прогулянок, іграшок та кімнат, де діти зможуть на деякий час залишатися та грати в ігри у своїх кімнатах.

Культурні заходи в багатоквартирних будинках проводяться відповідно до плану, розробленого студентським самоврядуванням та затвердженого керівником закладу. Усі події повинні закінчуватися до 23:00.

Вартість проживання в житловому приміщенні складається з витрат на його утримання та вартості обов'язкових послуг, що надаються студентам, які проживають у житловому приміщенні. Конкретний розмір плати за проживання визначається керівником вищого навчального закладу і розраховується відповідно до законодавства, тарифів, порядку оплати комунальних послуг, залежно від рівня комфортності розміщення.

## **1.1 Завдання гуртожитку**

Основні завдання:

- безпосереднє управління господарською діяльністю з експлуатації гуртожитку, організацією побуту мешканців, дотриманням санітарних та протипожежних норм проживання, дотриманням встановленого порядку;
- проживання в гуртожитку на підставі наказу, паспорта, медичних довідок, квитанції про оплату проживання;
- ознайомлення мешканців із «Правилами для студентського гуртожитку», «Внутрішніми правилами», «Правилами пожежної безпеки» та «Правилами безпеки» при поданні заяв у гуртожиток;
- утримання приміщень гуртожитку відповідно до встановлених санітарних норм та правил;
- забезпечення гуртожитку меблями, обладнанням та іншим інвентарем;

- подання заявок на ремонт в гуртожитку, обладнання, інвентар, утримання в належному стані відведеної території та зелених насаджень;
- забезпечення та контроль відповідної ремонтної діяльності приміщень гуртожитку та території;
- сприяння студентської ради гуртожитку у розвитку студентського самоврядування з питань праці, побуту та дозвілля мешканців;
- здійснення заходів щодо поліпшення побутових та культурних умов в гуртожитку;
- своєчасне відрухування студентів, розірвані зв'язки з університетом[19].

## 1.2 Аналіз конкурентів

В першу чергу ми розглянемо існуючі сайти гуртожитків. При аналізі більшості сайтів за запитом "Гуртожитки" ми отримуємо переадресацію на сайти університетів, де і надана інформація.

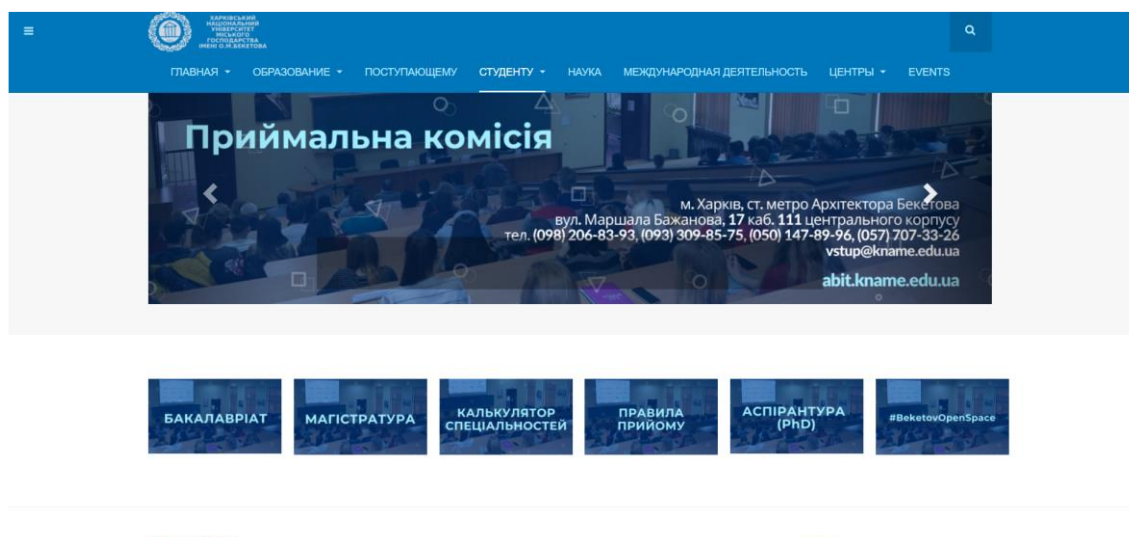


Рисунок 1.1 – Головна сторінка сайту kname.edu.ua

На сайті kname.edu.ua розміщена інформація про гуртожитки університету, на даному сайті присвячений розділ гуртожитках, за якими можна подивитися ту чи іншу інформацію, як такого сайту гуртожитку виявлено не було.

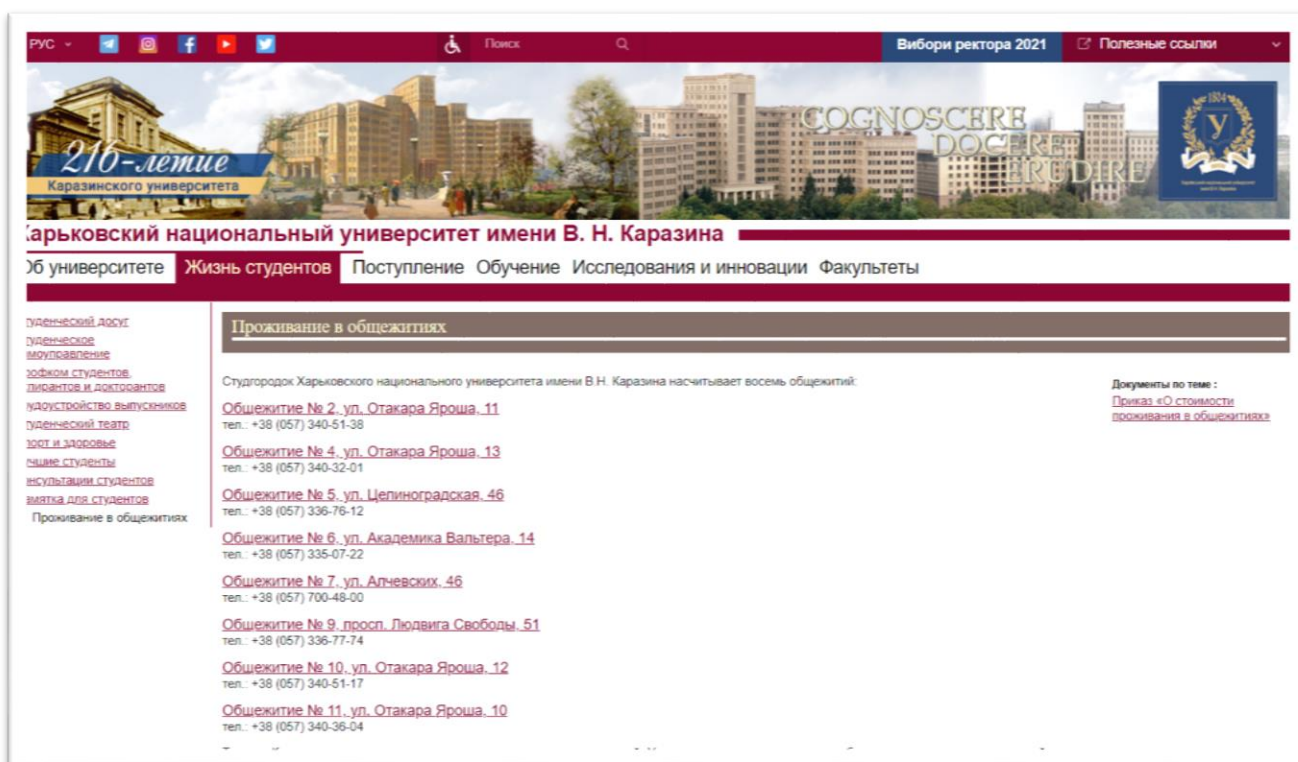


Рисунок 1.2 – Головна сторінка сайту univer.kharkov.ua

Переходячи на вкладку життя студентів ми потрапимо в розділ гуртожитку, де описана інформація, про розміщення гуртожитків за адресую, при переході на цю адресу буде відкрита вкладка «Google maps» з розташуванням. Також трохи нижче ми побачимо коротку інформацію по оплаті за місяць і умови проживання.

## РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ

### 2.1 Вибір і опис методології проектування

Основна концепція і зміст веб-сайту.

Цільова аудиторія сайту користувачі, що володіють хоча б мінімальними знаннями в ІТ технологіях і звикли шукати інформацію в мережі інтернет. Головне завдання сайту - надавати потрібну інформацію відвідувачам.

Виходячи з поставлених завдань, web-сайт повинен надавати такі можливості:

- при першому відвідуванні сайту користувач повинен зрозуміти, для чого служить даний сайт, скласти загальне враження про нього, з'ясувати свої потреби щодо сайту. Для цього йому повинен бути доступний перелік усіх послуг.

Дизайн веб сайту включає в себе: кольорове оформлення, елементи навігації по сайту, текстову інформацію, опис послуг пропонованих організацією.

Дизайн сайту грає важливу роль в створенні самого сайту, а саме він повинен відповідати наступним вимогам:

- зовнішній вигляд сайту повинен відповідати обраній тематики;
- навігація по сайту повинна бути зручною для користувача;
- головні матеріали, пропозиції та категорії послуг повинні бути розташовані на першому плані, щоб зацікавити відвідувача[10];

У верхній частині сторінки поміщається назва сайту, що відображає його тему.

Зображення у вікні браузера складається з основних елементів:

- Назва сайту;
- Головне інформаційне меню з права сторінки;
- Назва сайту знаходиться на кожній сторінці і не змінюється в залежності від місцезнаходження на сайті.



- Меню призначається для навігації по сайту, в ньому відображаються посилання на всі сторінки сайту.
- Основний зміст сторінки змінюється в залежності від місцезнаходження користувача.
- Головна сторінка. Ця сторінка відображається при першому вході на сайт, її оформлення і зміст надає користувачеві перше уявлення про сайт.

Сайт - це сторінка в інтернеті на якій присутня та чи інша інформація, для його виявлення сайту в інтернеті необхідні 3 умови:

1. Доменне ім'я (по якому можна буде отримати доступ до сайту)
2. Хостинг (для розміщення сайту на виділеному сервері)
3. Зміст сайту (наповнення всередині)

## **2.2 Технології розробки сайту**

Одним з найважливіших елементів написання сайту є вибір способу розробки.

Всі сторінки сайт складаються з HTML. По суті це мова розмітки, який включає в себе текст і зображення. Служить для відображення виключно вмісту сайту. Також ще однією особливістю є, то що при використанні тільки мови розмітки, сайт буде статичним і не буде змінюватися при подальшому відвідуванні.

Щоб зробити сайт не статичним використовують JavaScript і PHP. Більшість сайтів написані на цих мовах програмування. Це використовується для вдосконалення сайту так як при використанні php буде містити файли на стороні сервера які містять скрипти для виконання. При використанні сайт буде динамічним і буде змінюватися в залежності від певних дій на сайті[1].

Також є конструктори сайтів, які інтуїтивно зрозумілі, але не настільки гнучкі, тому цей спосіб ми не розглядаємо.

Інтернет швидко росте, і конкретні налаштування включають:

- поширення структури веб-сайту;

- вибір програмного введення;
- відкриття керівництва та інтелектуального інтерфейсу користувача;
- розподіл структури форм живлення;
- розробка оптимальної навігаційної структури системи (веб-сайту).

Рішення для проектування бази даних особливо підходить для вирішення проблем.

Головною вимогою до розвитку інформаційного забезпечення є вибір СУБД (СУБД) для зберігання інформації. MySQL був обраний такою СУБД, оскільки ця СУБД має архітектуру клієнт-сервер, працює в операційних системах сімейства Unix і не займає багато машинної пам'яті. Підтримка сервера MySQL автоматично включається в РНР.

Багато людей починають працювати на новому сайті з вибором кольорів, шрифтів, технологій тощо. Хоча має сенс почати з аналізу проекту, планування майбутньої роботи та пошуку відповідей на низку важливих питань щодо майбутнього сайту.

Визначення з текстом для кожної сторінки, перш ніж працювати над дизайном. Навігація на сайті буде залежати безпосередньо від кількості та глибини вмісту, а його стиль та тон впливатимуть на візуальну мову композиції.

Тактична сторона технічних специфікацій може бути відображена у короткому документі, який визначає підхід та технології, що використовуються при кодуванні сторінок та візуальному маркуванні.

- Чи будуть сторінки генеруватися на базі бази даних на льоту?
- Чи будуть використовуватися каскадні таблиці стилів (CSS)?
- Чи буде встановлений плагін або спеціальне управління машиною користувача?
- Який мінімум потрібно для оптимізації палітри кольорів?

Третій етап включає:

- творчий пошук і розвиток стилю;
- створення прототипів;

– розробка графіки: дизайн баз даних і форм, анімація, оптимізація зображень, обробка HTML.

Працюючи над інтерфейсом, важливо стежити за всіма вхідними ідеями та вдосконалювати їх. Можна розпочати з найважливішої сторінки і працювати над нею, поки не отримуймо принаймні 15 ескізів з різними варіантами, з яких 5 можна буде розвивати далі. Також проєкспериментувати з різними ефектами: реальним аудіо / відео, динамічним html, незвичним використанням тегів тощо.

### **2.3 Вибір засобів розробки**

Локальний сервер «OpenServer» - це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань поширюється по ліцензії -GNU GPL.

Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а також для надання веб-сервісів в локальних мережах.

Хоча спочатку програмні продукти, що входять до складу комплексу, не розроблялись спеціально для роботи один з одним, така зв'язка стала дуже популярною серед користувачів Windows, в першу чергу через те, що вони отримували безкоштовний комплекс програм з надійністю на рівні Linux серверів.

«OpenServer» - комплексне рішення для професійних веб майстрів, програма поставляється в декількох комплектаціях, від стандартного набору основних програм серверного оточення, до розширеного, в який входять як програми для редагування графіки, так і система контролю версія - «Git».

Основні особливості пакета «OpenServer»:

1. Не потрібно установка (Портативність).
2. Можливість роботи з USB накопичувача.

3. Одночасна робота з іншими серверами «XAMP» / «Denwer».
4. Робота на зовнішньому IP.
5. Захист сервера від зовнішнього доступу.

Технічні вимоги «OpenServer»:

Windows (32/64 біт): Windows 7 SP1 і новіші версії;

Забезпечено часткова підтримка WindowsXPSP3 і WindowsVista; Мінімальні апаратні вимоги: 500 МБ вільної RAM і 3 ГБ вільного місця на HDD; Потрібна наявність Microsoft VisualC ++ 2005-2020.

На даний момент «OpenServer» є найактуальнішим пакетом веб-оточення для розробки сайтів, для нього виходять часті оновлення, де всі компоненти такі як Apache, MySQL, PHP, постійно отримують найостанніші версії, найближчий конкурент «Denwer», дуже давно зупинився на старих версіях програмного забезпечення, після чого і був розроблений «OpenServer».

Консоль управління сервером, має приємний зовнішній вигляд, зручний для управління, як починаючому розробнику, так професіоналу. Консоль також дуже функціональна і дозволяє викликати всі файли налаштувань, не вдаючись до перебору безлічі конфігураційних файлів (Рисунок 2.1.).

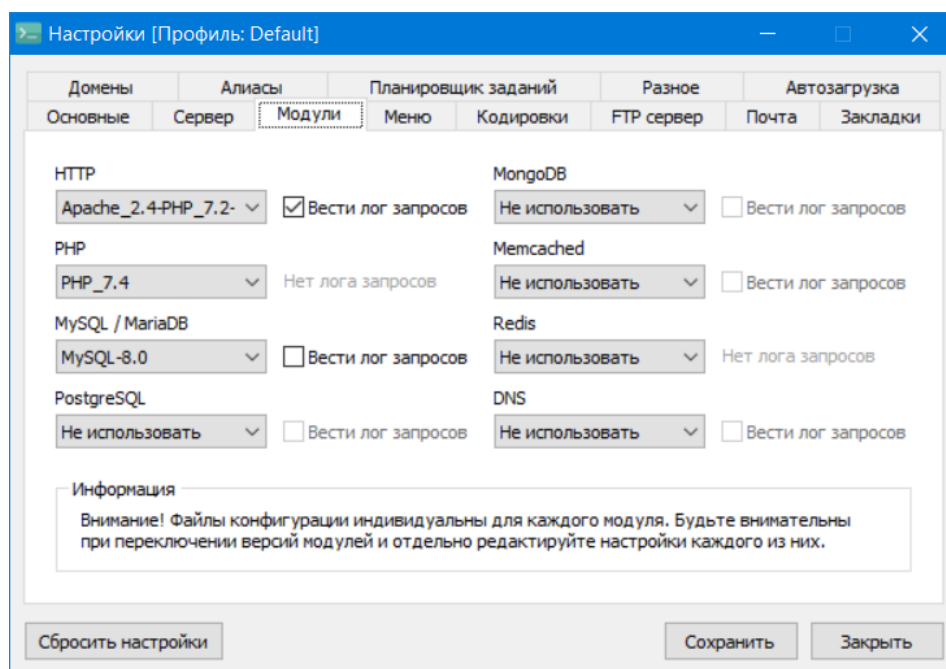


Рисунок 2.1. Інтерфейс консолі локального веб-сервера «OpenServer»

Visual Studio Code - це редактор коду, розроблений Microsoft для Windows та інших операційних систем. Він позиціонується як "полегшений" редактор коду для розробки міжплатформенних веб- та хмарних додатків. Включає налагоджувач, інструменти Git, підсвічування синтаксису, інструменти IntelliSense та рефакторинг. Існує широкий спектр параметрів налаштування: власні теми, комбінації клавіш та файли конфігурації. Він поширюється безкоштовно, розробляється як програмне забезпечення з відкритим кодом, але готові вузли розповсюджуються за власною ліцензією.

Редактор коду Visual Studio підтримує ряд мов програмування, виділення синтаксису, IntelliSense, рефакторинг, налагодження, навігацію кодом, підтримку Git тощо. Багато функцій Visual Studio Code недоступні через графічний інтерфейс; вони часто використовуються через командну палітру або файли JSON (наприклад, власні налаштування). Палітра команд - це тип командного рядка, який викликається за допомогою комбінації клавіш.

Visual Studio також дозволяє замінити кодову сторінку, зберігаючи документ, символи стрічки рядків та мову програмування поточного документа.

Станом на 2021 рік існує розширення Python з відкритим кодом для Visual Studio Code. Він надає розробникам широкі можливості редагування, налагодження та тестування коду. Вихідний код, розроблений Microsoft для Windows, Linux та macOS. Він позиціонується як редактор коду "lek" для крос-платформної розробки веб- та хмарних додатків. Включає налагоджувач, інструменти Git, підсвічування синтаксису, інструменти IntelliSense та рефакторинг. Існує широкий спектр параметрів налаштування: налаштування з них, комбінації клавіш та файл конфігурації. Це програмне забезпечення розповсюджується безкоштовно, розроблений пакет програм з відкритим кодом, але ви готові розповсюджувати його за власною ліцензією.

Розширення також дозволяють легше отримати доступ до таких програм, як Docker, Git та інші. Розширення надають кодові посилання, теми редакторів та підтримку синтаксису для певної мови.

PhpMyAdmin - це веб-програма з відкритим кодом PHP, яка забезпечує веб-інтерфейс для адміністрування СУБД MySQL. PhpMyAdmin дозволяє браузеру, а не просто адмініструвати сервер MySQL, виконувати SQL-команди та переглядати вміст таблиць та баз даних.

Це головний інструмент для роботи з динамічними сайтами. Для створення вмісту потрібно створити фреймворк HTML і відобразити інформацію з бази даних на сторінці за допомогою PHP-скриптів. Перший генерується для бази даних користувачів з правами адміністратора: ви можете створювати, видаляти, редагувати таблиці, додавати нові рядки. Інтуїтивний інтерфейс не вимагає знання мови, достатньо зрозуміти синтаксис: Type, Table, Change, Create. Бажаний порядок можна знайти за допомогою швидкого пошуку.

Це головний інструмент для роботи з динамічним веб-сайтом. Наразі phpMyAdmin дозволяє:

- створення та запуск баз даних
- створювати, копіювати, видаляти, перейменовувати та змінювати таблиці
- підтримує таблиці
- видалити, редагувати та додавати поля
- виконує SQL-запити, включаючи пакетні SQL-запити
- керувати клавішами
- завантажувати текстові файли в таблиці
- створення (\*) та перегляд дамів таблиць
- експортувати (\*) дані у CSV, XML, PDF, ISO / IEC 26300 - Текст та таблиці OpenDocument, формати Word, Excel та LATEX
- адміністрування декількох серверів
- керує користувачами та привілеями MySQL
- перевірити цілісність довідкових даних у таблицях MySQL
- використовувати запит на прикладі (QBE), створювати складні запити, автоматично посилаючись на певні таблиці
- створити графічну схему бази даних у форматі PDF

- пошук у базі даних або її розділах
- змінювати збережені дані у різних форматах, що використовуються в наперед визначених функціях, наприклад, відображаючи дані BLOB як зображення або як посилання для завантаження тощо.
- підтримує таблиці InnoDB та зовнішні ключі
- підтримує mySQLi, вдосконалене розширення MySQL
- перекладений більш ніж на 50 мов

Маркування сторінки – html.

Всесвітня павутина здійснила революцію в інформатиці, дозволивши кожному публікувати документи HTML. Донедавна інформація в цих документах була в основному статичною, що вимагало від сервера відповіді на дії користувача. З появою динамічного HTML, Інтернет-парадигма перейшла від взаємодії із сервером до створення інтерактивних веб-сайтів та веб-додатків. Оскільки динамічний HTML дозволяє HTML-документам взаємодіяти з користувачем та повністю їх змінювати на клієнтському комп'ютері, ви можете створювати повністю функціональні веб-додатки. Це програма SGML (стандартна узагальнена мова розмітки). У документі HTML / SGML теги визначають структуру вмісту документа. Традиційний документ SGML має три різні компоненти: структуру, стиль та зміст. З введенням динамічного HTML був доданий четвертий компонент: модель поведінки. Термін поведінковий шаблон визначає взаємодію між користувачем та сторінкою HTML. Структура представлена низкою сімейств елементів; стиль присутній для кожного елемента та у всіх сімействах таблиць стилів. Вміст представлений усіма елементами та об'єктами TextRange. Сценарії маніпулюють структурою, стилем та вмістом у відповідь на події, щоб створити модель поведінки документа[9].

Очевидно, що HTML збереже свою позицію головного інструменту для позначення тегами документів WWW завдяки своїй простоті та винятковій корисності.

### Сценарій Javascript.

Програма інтерпретується як така завдяки реалізації ECMAScript mov і базується на концепції прототипів.

За давного часу JavaScript в основному використовується для відкриття сценаріїв поведінки браузера, оскільки вони генеруються веб-сайтом, а також для цілей доступу до скриптів для доступу до документів. Інтерпретатор JavaScript, який дозволяє розширити ваші функціональні можливості без зміни самого надбудови, виконується програмами.

Мобільне програмне забезпечення JavaScript було випущено Netscape компанією Sun Microsystems і призначене для розробки інтерактивних документів HTML. Основними напрямками JavaScript є:

- основу динамічних посилань, посилання тобто, які можуть змінюватися, коли документ заблоковано;
- перевірка правильності збережених анкет користувачем перед відправкою їх на сервер;
- перегляд додаткових сценаріїв та інших областей. Дозволяє отримати доступ до додаткової інформації, яку можуть використовувати як клієнт, так і сервер. При розробці доповнень обох типів ядро називається так званим, як позначення стандартних об'єктів. Клієнти запускаються з браузера користувачів[12].

Код JavaScript включається в HTML-код країни і відображається як інтерпретатор, який завантажується в браузер (Рисунок 2.2).



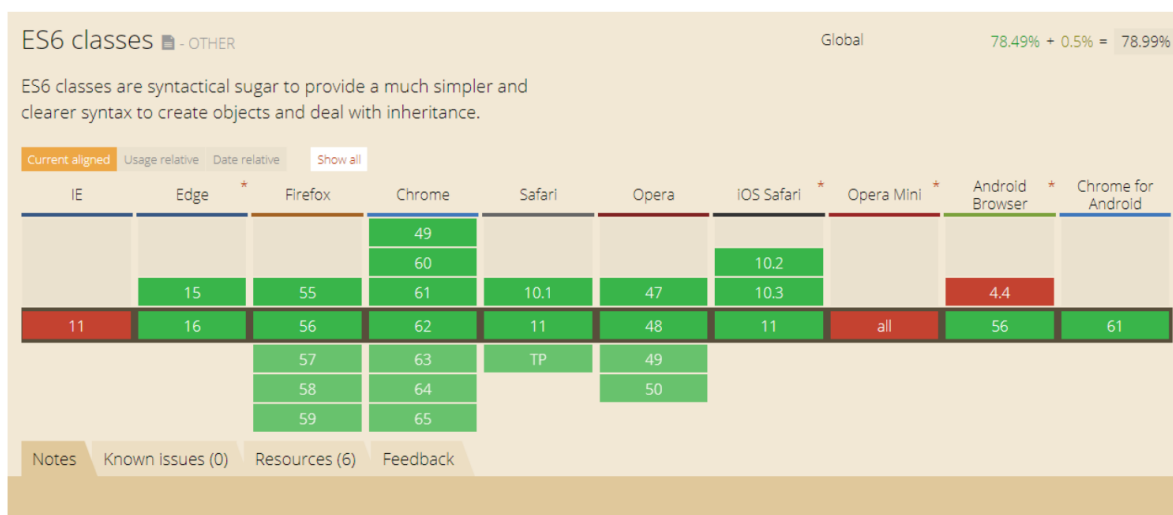


Рисунок 2.2 – Браузерна підтримка класів введених в ES6

Ядро JavaScript розкриває набір основних об'єктів, таких як масиви, дати та математики, цей базовий набір сучасних елементів, таких як операції, структури управління та оператори. Ядро JavaScript можна розширити для різних цілей, додавши нові об'єкти; наприклад:

а) JavaScriptntsky розширює ядро курсу, додаючи браузер (Навігатор або під-веб-браузер) та Модель об'єкта документа (DOM). Наприклад, розширення клієнта дозволяє програмі змінювати елементи у форматі HTML і реагувати на потужні твердження, такі як клацання на цілі, введення форми або навігація країною;

б) Серверний JavaScript розширить ядро переміщення, додають об'єкти, підключить його для запуску JavaScript на сервері. Наприклад, розширення сервера дозволяють програмам підключатися до реляційної бази даних, збирати інформацію та доповнень або маніпулювати файлами сервера.

По мірі зростання Інтернету веб-інтерфейси набули широкого поширення, що дозволяє взаємодіяти з різними програмами через браузер.

Етап програмування включає:

- розробка: CGI, Javascript, Java, форми, бази даних;
- HTML кодування;
- перевірка: зміст, граматичні помилки, посилання;
- тестування в різних браузерах.

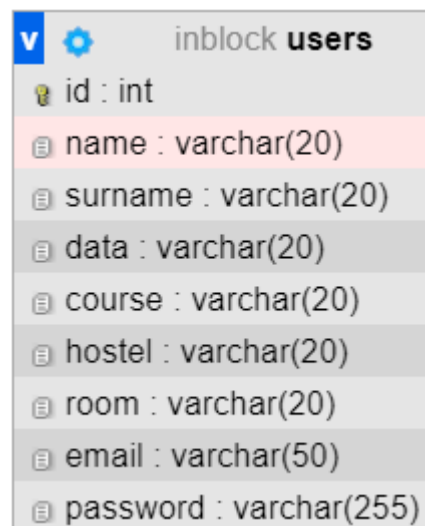
Після проектування та створення макетів на головних сторінках можна виконувати технічну роботу - кодування HTML.

## 2.3 Проектування схеми і опис бази даних

Інтернет стрімко зростає, і рішення для проектування баз даних особливо підходять для вирішення багатьох проблем.

Враховуючи велику кількість ресурсів, доступних в Інтернеті, сучасні веб-сайти зазвичай використовують Інтернет-базу даних. Використання такої бази даних дозволяє швидко та ефективно обробляти запити користувачів та надійно зберігати дані. У дипломній роботі веб-сайт розроблений із використанням зразка бази даних[2].

### 2.3.1 Таблиця «users»



The image shows a screenshot of a database table definition for 'users' in a database named 'inblock'. The table has the following fields:

Field Name	Data Type
id	int
name	varchar(20)
surname	varchar(20)
data	varchar(20)
course	varchar(20)
hostel	varchar(20)
room	varchar(20)
email	varchar(50)
password	varchar(255)

Рисунок 2.3. – Таблиця БД «users»

Дана таблиця відповідає за збереження у БД даних які водили в форму реєстрації користувачі (Рисунок 2.3.).

Таблиця містить наступні поля:

- id – INT – первинний ключ запису.

- name – VARCHAR(20) – строкове значення, ім'я користувача, наприклад «Микита».
- surname – VARCHAR(20) – строкове значення, прізвище користувача, наприклад «Рудак».
- data – VARCHAR(20) – строкове значення, що зберігає дату народження.
- course – VARCHAR(20) – строкове значення, що зберігає дані про курс користувача.
- hostel – VARCHAR(20) – строкове значення, визначає в якому гуртожитку проживає користувач.
- room – VARCHAR(20) – строкове значення, визначає кімнату.
- email – VARCHAR(20) – строкове значення, зберігає email користувача.
- Password – VARCHAR(255) – строкове значення, зберігає пароль користувача.

### 2.3.2 Таблиця «news»

Дана таблиця відповідає за збереження у БД даних про новини які відображаються на сайті (Рисунок. 2.4.).

Column Name	Data Type	Key
id	int	Primary Key
title	varchar(200)	
text	text	
img	text	
data	varchar(20)	

Рисунок 2.4. – Таблиця БД «news»

- id – INT – первинний ключ запису.
- title – VARCHAR(200) – строкове значення, яке зберігає заголовок новини.

- text – TEXT – представляє текст довжиною до 65 КБ ,містить текст новини.
- img – TEXT – представляє текст довжиною до 65 КБ, що зберігає повний шлях до розташування зображення.
- data – VARCHAR(20) – строкове значення, що зберігає дату публікації новини.[18]

## 2.4 Вибір СУБД

Система управління базами даних phpMyAdmin входить до стандартного набору програм Microsoft Office, який використовується майже в усіх організаціях України. Він надає значні можливості для роботи із збереженими даними, обробки та спільного використання.

Всі об'єкти, підключені до однієї бази даних, зберігаються phpMyAdmin в одному файлі з розширенням `mbd`, серед об'єктів, що розробляються, є база даних:

1. Таблиці є основними об'єктами будь-якої бази даних. Таблиці зберігають усі доступні дані в базі даних, крім того, таблиці зберігають структуру бази даних (поля, їх типи та властивості).

2. Запити - використовуються для отримання даних із таблиць та представлення їх користувачеві у зручній формі. Запити виконують такі операції, як вибір, сортування та фільтрування даних. Використовуючи запити, ви можете виконувати перетворення даних за алгоритмом, автоматично заповнювати таблиці даними, імпортованими з інших джерел, виконувати прості обчислення в таблицях та багато іншого.

3. Форми є засобом введення даних. Їх важливість полягає в тому, щоб забезпечити користувача засобами для заповнення лише тих полів, які він повинен заповнити. У той же час у формі можна розмістити спеціальні елементи керування для автоматизації введення.

4. За своїми властивостями та структурою звіти багато в чому схожі на форми, але призначені лише для виведення даних. У зв'язку з цим звіти відрізняються тим, що вони вжили спеціальних заходів для групування вихідних даних та відображення спеціальних елементів дизайну, характерних для друкованих документів[6].

На етапі логічного проектування бази даних будується модель домену, яка повинна відображати взаємозв'язок об'єктів у домені. Діаграма, що відображає взаємозв'язки об'єктів у домені, називається діаграмою типу ER (від англ. Entity - об'єкт, Relationship - зв'язок).

Об'єкт - це клас об'єктів одного типу, інформація про який повинна враховуватися в моделі. Зв'язок - це зв'язок між двома об'єктами. Один об'єкт може бути пов'язаний з іншим об'єктом або з самим собою.

Атрибут сутності - це іменована характеристика, це властивість сутності. Ключ об'єкта - це непотрібний набір атрибутів, значення яких є колективно унікальними для кожного екземпляра об'єкта.

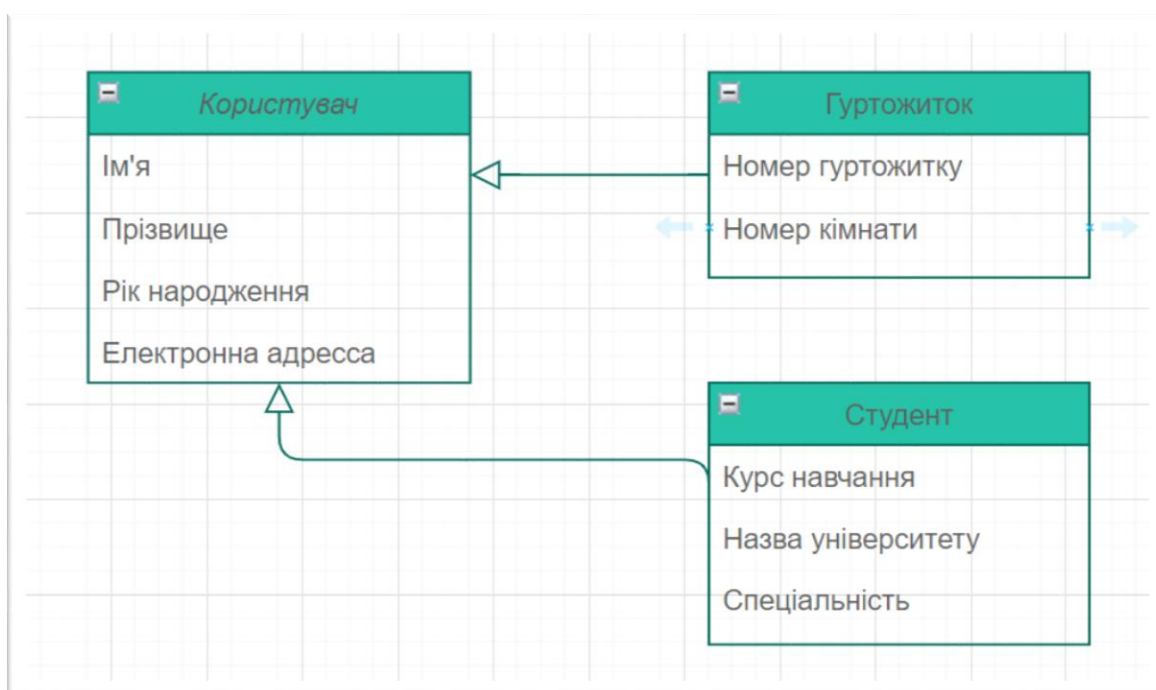


Рисунок 2.5 – ER-діаграма гуртожитку

Таблиці пов'язані між собою, що забезпечує цілісність даних. Забезпечення цілісності даних означає виконання для взаємопов'язаних таблиць наступних умов коригування бази даних:

- в підпорядковану таблицю не може бути додано запис, для якої не існує в головній таблиці ключа зв'язку
- в головній таблиці не можна видалити запис, якщо не видалені пов'язані з нею записи в підлеглий таблиці
- зміна значень ключа зв'язку головної таблиці повинно приводити до зміни відповідних значень в записах підпорядкованої таблиці

## **2.5 Проектування макета сайту**

Одним з важливий частин при створенні сайту є створення макета. Початкова візуалізація проекту відіграє важливу роль при остаточному результаті, так як можна відразу оптимізувати сторінку, блоки інформації розрахувати скільки місце буде займати той чи інший блок. Дизайн це саме те що бачить користувач в першу чергу і на що звертає увагу і дуже важливо зробити його інтуїтивно зрозумілим для сприйняття, щоб користувач відразу розумів, де що знаходиться після першого перегляду сторінки.

Таким чином було прийнято рішення почати розробку макету сайту в «Adobe Photoshop 2021». Первісна сторінка, на яку потрапляє користувач має містити відмінні риси від інших сайтів. Одним з відмінних рис є логотип, який був розроблений спеціально під сайт гуртожитку. В першу чергу, дивлячись на логотип має відразу бути зрозуміло, до чого він ставиться, в нашому випадку логотип "Гуртожитки онлайн" представляє собою обриси будинку і назви посередині. Другим важливим фактором є назва сайту (in block), в перекладі (в блоке), що також підкреслює смисловий концепт проекту.

Точна робота з кольором. Правильно нанесений колір може, наприклад, передавати тонкі відмінності між однорідними елементами. Неправильно застосований колір може заважати роботі програми.

Особливо це стосується червоного. Так сталося, що для всіх людей червоний колір асоціюється з якоюсь небезпекою. Велика кількість червоного кольору десь на екрані привертає увагу, змушуючи користувача попереджати, думаючи, що щось не так.

Керуючись цими принципами розробки інтерфейсу, було вирішено покладатися на простоту та інформаційний зміст, щоб користувач, потрапивши на сайт, отримував чітку інформацію.

Колірна гамма буде орієнтована на помаранчевий-чорні тони, які добре гармонують з кольором тексту.

Інтерфейс - загальновизнаний - стандартно визначена межа між взаємодіючими незалежними об'єктами. Інтерфейс визначає параметри, процедури та характеристики взаємодії об'єктів.

Чим простіше, тим краще. Це не означає, що має бути лише текст і контактна інформація, але сайт не повинен містити марної інформації, шрифт повинен бути зручним для читання. Графіка повинна бути чіткою, виразною та швидкою для завантаження. Неприпустимо використовувати анімацію та звук на веб-сайті, що займає багато часу для завантаження та відволікання. Людське око сканує сторінки сайту зверху вниз.

Чим простіше верх сторінки, тим легше запам'ятати назву сайту та саму компанію. Типовою помилкою багатьох сайтів є накопичення вгорі сторінки сайту авангардної та складної композиції багатьох малюнків, серед яких назва компанії та назва сайту ледь помітні. Як результат, людське око, скануючи цю частину сторінки сайту, не встигає розпізнати і запам'ятати необхідну інформацію. Відбувається перенасичення. Замість того, щоб збільшити запам'ятовуваність, на таких сайтах зменшується основних підписів (фотографій) сайту, назви компанії та назви сайту.

Структура сайту - система взаємного розташування і взаємозв'язків файлів (сторінок) сайту. Часто під структурою сайту мають на увазі блок-схему сайту.

Розрізняють такі види структур сайтів:

1. Лінійна структура. Сторінки сайту розташовуються строго одна за одною. Зручна для створення невеликого за кількістю сторінок сайту з малою кількістю гіперпосилань і послідовним викладом матеріалів.

2. Структура у вигляді решітки. Заснована на побудові системи навігації сайту, коли між вертикальними і горизонтальними елементами (сторінками) є взаємозв'язок та можливість швидкого переходу з однієї сторінки на іншу без необхідності відвідування проміжних сторінок. Подібна структура призводить до зайвого збільшення гіперпосилань і застосування її обмежено для великих сайтів, тобто сайтів з великою кількістю сторінок.

Оптимальною структурою сайту вважається структура, яка задовольняє таким критеріям:

1. Існує головна сторінка, під якою розташовуються основні розділи сайту.
2. При натисканні на одну з гіперпосилань відвідувач переходить в один з основних розділів сайту, в якому, аналогічно, можуть бути розміщені гіперпосилання на інші станиці даного розділу.

При проектуванні структури динамічного сайту слід враховувати, що з часом кількість наданої відвідувачам інформації буде рости і щоб уникнути подальших витрат на редизайн, структура сайту буде оптимізована.

Система навігації сайту - сукупність текстових і графічних гіперпосилань, призначених для пошуку інформації на сайті. Найкраща пошукова система не зможе замінити коректну побудовану систему навігації сайту.

Яким би цікавим не був контент сайту, відвідувач не зможе з ним ознайомитися, якщо не зможе знайти його на сайті. Це завдання має вирішувати система навігації сайту.



Розрізняють вертикальну, горизонтальну і комбіновану навігацію сайту. Динамічний сайт «InBlock» матиме комбіновану навігацію, основне горизонтальне меню буде складатися з пунктів:

1. Старт
2. Про нас
3. Відео
4. Новини
5. Зареєструватися

Далі в залежності від обраного пункту користувач буде переправлений на сторінку.

## 2.6 Логотип

Логотип - важливий елемент відмінності компанії, який відіграє велику роль при проведенні рекламної кампанії і просуванні бренду. Цілком природно, що і розробляється логотип відповідно до певних правил.

Так, виділяють чотири основні етапи розробки логотипу:

1. підготовка, збір інформації, обробка інформації;
2. концепція;
3. внесення змін до концепції;
4. впровадження.

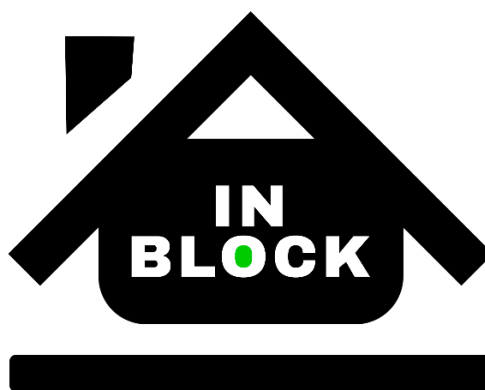


Рисунок 2.6. – Логотип сайту InBlock

У своєму повсякденному житті ми постійно стикаємось з тисячами логотипів. Вони містяться майже в кожному продукті (харчовому, промисловому, від автомобілів до паперу). На кожній упаковці ви можете знайти логотип виробника. Більшість друкованих засобів масової інформації (журнали, газети) мають власний унікальний логотип, що робить цю публікацію відмінною від інших[16].

У наш час логотип - це візитна картка компанії (основний засіб ідентифікації компанії)

Зазвичай логотип можна розділити на дві основні частини:

1. яскраво виражений - товарний знак;
2. невимовне - колір, шрифт, спеціальне зображення.

Особливості логотипу:

1. визначити відмінні риси компанії чи товару;
2. дати назву товару чи компанії;
3. сприяти визнанню рекламного об'єкта та вказувати джерело його походження;
4. інформує споживача про товар і стимулює бажання його придбати;
5. символізувати якісні характеристики продукту.

## **2.7 Реєстраційна форма**

Реєстраційна форма призначена для введення особистих даних користувачів. В майбутньому ця інформація буде використовуватися для їх ідентифікації між робочими сесіями. Ця інформація може зберігатися як на сервері, так і на клієнті.

Реєстрація необхідна для отримання можливостей, які доступні користувачам в особистому кабінеті. Для того, щоб створити обліковий запис, вам необхідно:

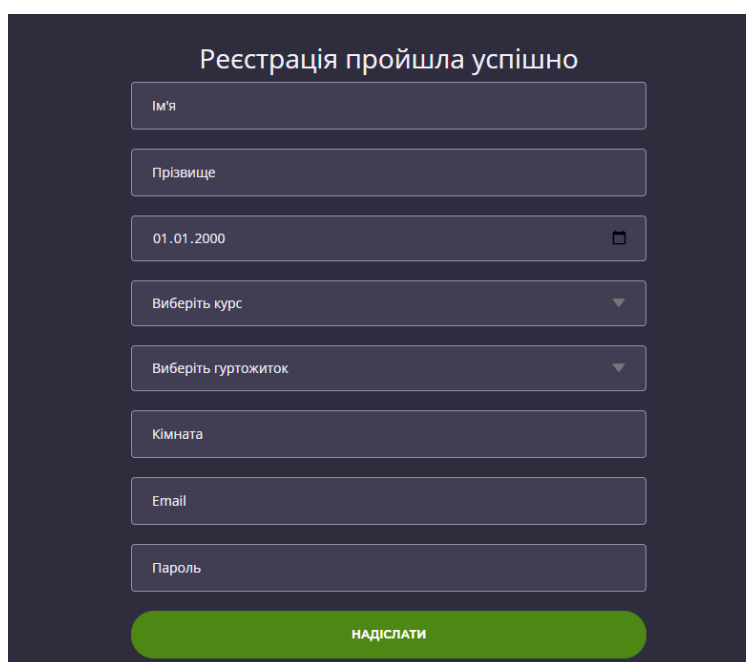
- Знайти кнопку «Реєстрація».

– Заповнити запропоновані поля. Буде потрібно надати максимально достовірну інформацію. Особливу увагу приділіть логіну та паролю - вони повинні бути досить складними, але в той же час запам'ятовуються для вас.

– Підтвердити вибір.

– Ви можете увійти, використовуючи логін і парольний фразу, введені в момент реєстрації.

На цьому створення облікового запису вважається завершеним - ви маєте повний доступ до всіх можливостей і переваг особистого кабінету[17].



Реєстрація пройшла успішно

Ім'я

Прізвище

01.01.2000

Виберіть курс

Виберіть гуртожиток

Кімната

Email

Пароль

НАДІСЛАТИ

Рисунок 2.7. – Реєстраційна форма

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ САЙТУ

### 3.1 Обґрунтування вибору інструментальних засобів

Після визначення об'єктів та встановлення зв'язків між ними вони переходять до етапу «Вибір інструментів для розробки баз даних». Цей етап є одним з найважливіших моментів у розробці проекту бази даних, оскільки він принципово впливає на весь процес проектування бази даних та впровадження інформаційної системи. Теоретично при виборі бази даних слід враховувати десятки факторів. Але на практиці розробники керуються лише власною інтуїцією та кількома найважливішими критеріями, які зокрема включають:

- тип моделі даних, яку підтримує дана СУБД, її адекватність потребам предметної області;
- робочі характеристики системи;
- запас функціональних можливостей для подальшого розвитку системи;
- наскільки система оснащена інструментами для персоналу, що займається адмініструванням даних;
- Зручність і надійність роботи СУБД;
- вартість СУБД та додаткового програмного забезпечення.

СУБД - це програмний пакет, призначений для виконання операцій обробки даних з метою надання інформації користувачам.

Сьогодні однією з найпопулярніших настільних баз даних є Ms Access. Це пояснюється тим, що Access має дуже широкий спектр інструментів для введення, аналізу та представлення даних. Ці інструменти не тільки прості та зручні, але й високопродуктивні, що забезпечує високу швидкість розробки додатків.

У цьому дипломному проєкті база даних Ms Access була використана як інструмент розробки бази даних, оскільки одним із головних критеріїв вибору СУБД є модель даних (використовується для представлення даних), а Ms Access підтримує реляційну модель даних.

Реляційна модель даних представлена у вигляді двовимірної таблиці, інформаційними одиницями якої є поля, домени та кортежі. Для опису інформаційних об'єктів використовується відповідна мова опису інформаційних документів.

Крім того, СУБД MS Access була обрана на основі наступних функціональних можливостей:

- забезпечує ефективну обробку великих обсягів інформації;
- забезпечує доступ до всіх типів даних;
- існує можливість одночасного використання декількох таблиць з базами даних;
- незважаючи на те, що Access є потужною та складною системою, її використання не є складним для непрофесійних користувачів;
- простота створення таблиць за допомогою одного з трьох режимів: за допомогою конструктора, за допомогою майстра та шляхом введення даних;
- простота створення посилань між таблицями за допомогою "Редактора посилань";
- можливість створювати форми за допомогою конструктора та майстра форм;
- існування вбудованого механізму перевірки цілісності даних;
- СУБД доступ дозволяє створювати різні звіти на основі даних з таблиць та інших об'єктів бази даних.

VS Code його функціонал дозволяє розробляти програми з графічним інтерфейсом, а також консольні програми, включаючи програми, що підтримують технологію Windows Forms, а також веб-сайти, веб-програми, веб-служби як у власному, так і в керованому коді для всіх платформ.

Редактор містить вбудований налагоджувач, інструменти для роботи з Git та інструменти для рефакторингу, навігації кодом, автоматичного завершення типових конструкцій та контекстуальних підказок.

Продукт підтримує розробку платформ ASP.NET і Node.js і вважається легким рішенням, яке позбавляє потреби в повному IDE.

Великим плюсом редактора є підтримка великої кількості мов, таких як C++, C#, Python, PHP, JavaScript та інших.

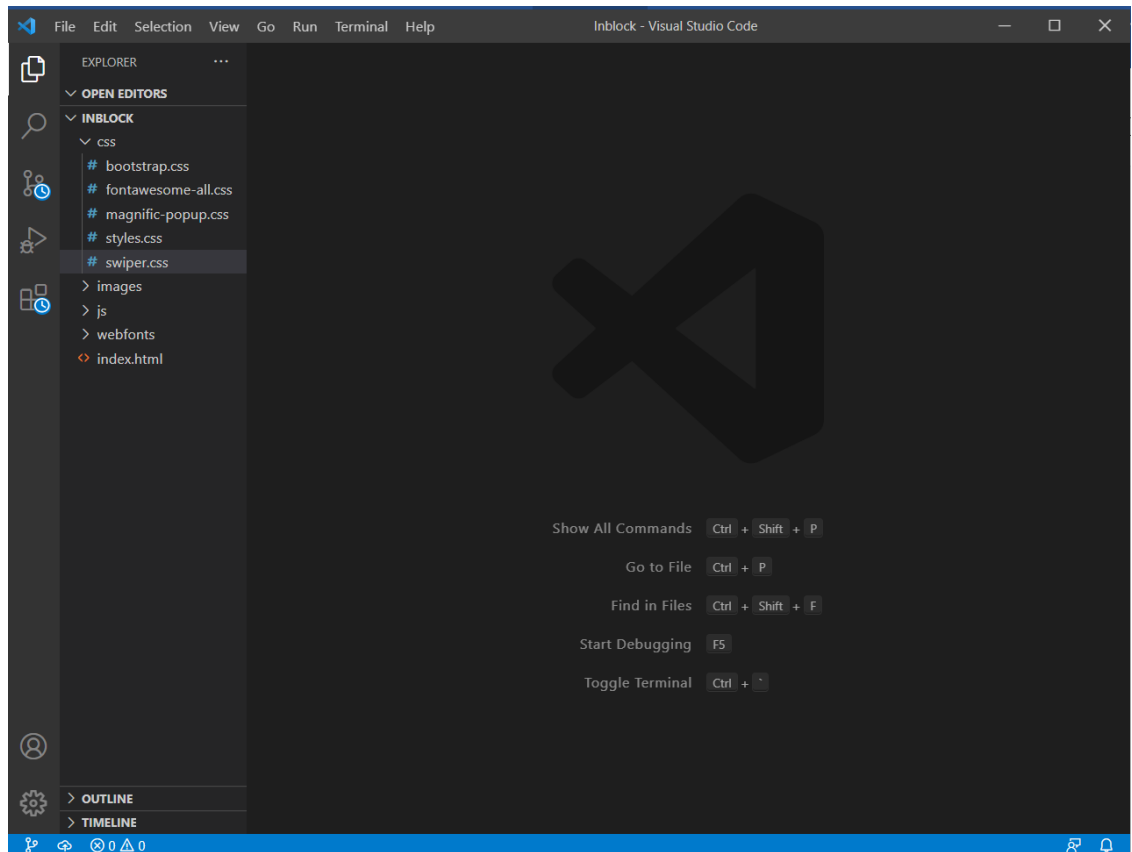


Рисунок 3.1. – Інтерфейс середовища розробки Visual Studio Code

#### Функції коду Visual Studio:

- вбудовані інструменти для інтеграції з GitHub, GIT та Visual Studio Team Services для швидкого тестування, складання, упаковки та реалізації різних типів програм;
- простота роботи з проектами;
- працювати з Mono та Node.js за допомогою вбудованого налагоджувача;
- підтримка TypeScript та JavaScript;
- публікація програм, створених у Microsoft Azure, за допомогою Visual Studio Team Services;
- підтримка майже всіх мов програмування;

- написання коду для конкретного завдання з подальшою його інтеграцією в проект (з доповненням або безпосередньо);
- велика бібліотека шаблонів, готових фрагментів коду та фрагментів з можливістю додавання власних елементів;
- одночасна робота з декількома проектами (у декількох вікнах);
- інтерфейс можна розділити на дві панелі порівняння коду;
- функція налагодження.

Переваги коду Visual Studio:

- безліч налаштувань (як всієї програми, так і інтерфейсу);
- розширювана бібліотека доповнень та готових рішень;
- універсальність (редактор підтримує майже всі мови, що використовуються для створення додатків);
- простота та гнучкість.

### 3.2 Опис призначеного для користувача інтерфейсу

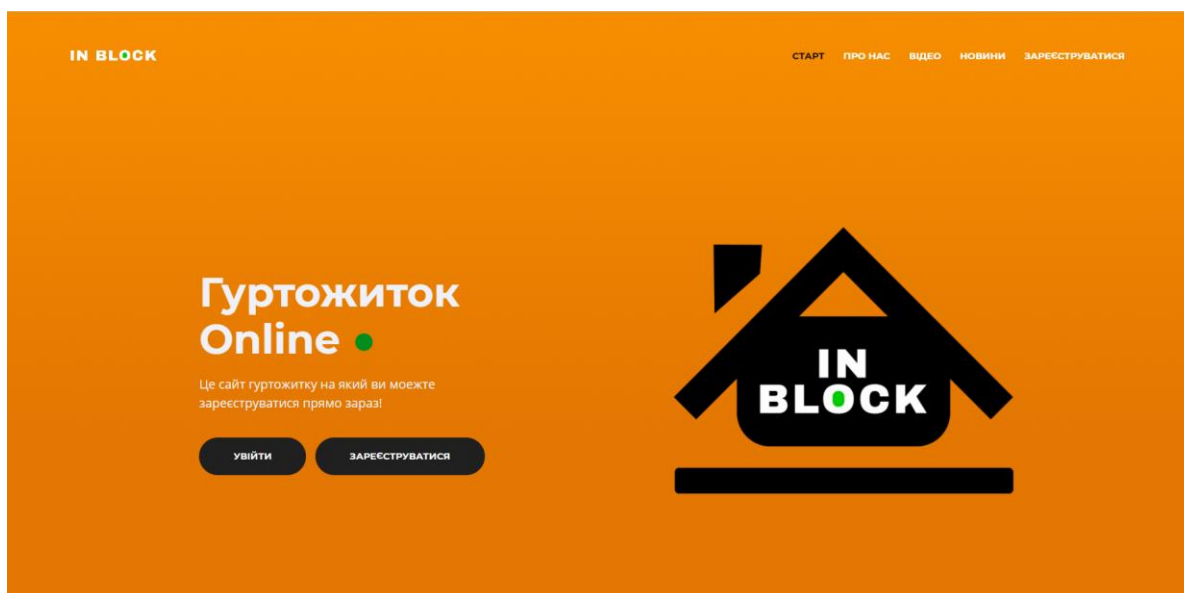
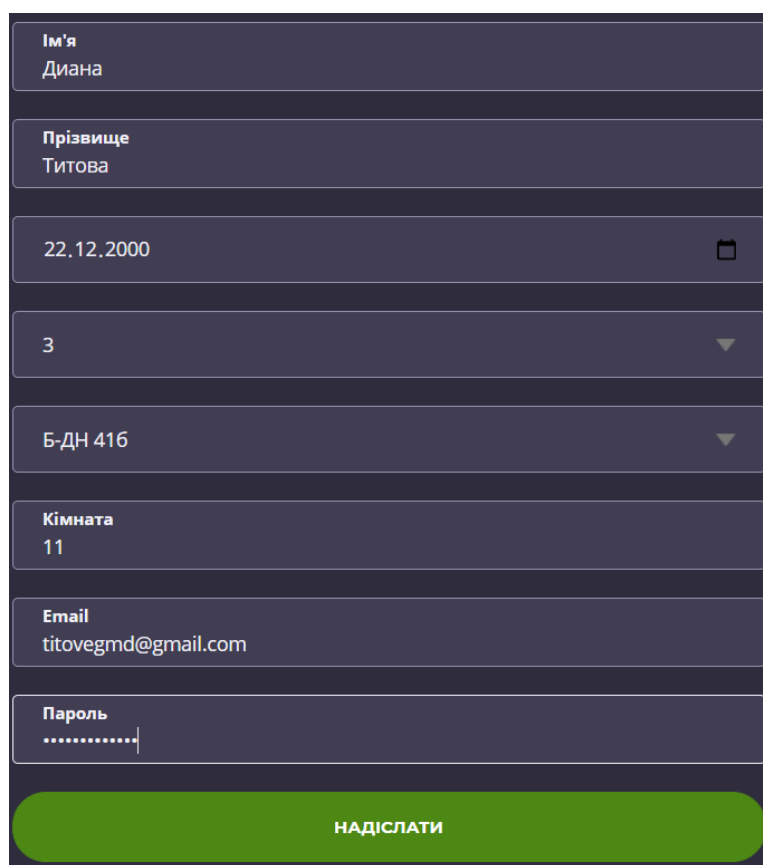


Рисунок 3.2. – Головна сторінка сайту

Перше, що бачить користувач під час заходу на сайт - це головна сторінка сайту. Одна з головних завдань головної сторінки - зустріти користувача і зробити його постійним відвідувачем. Щоб він затримався на сайті якомога довше.

На головній сторінці сайту міститься основна інформація, доступна користувачу при переході на адресу сайту. Саме з головної сторінки і починається знайомство відвідувача з сайтом. Відразу розташовуються 2 функціональні кнопки (реєстрація, і вхід) при натисканні на які виконується одна з дій. Наприклад при натисканні на кнопку реєстрації користувача перенаправляє в низ сайту, де розміщена форма реєстрації користувачів. Також на головній сторінки присутній меню сайту і відмінні чорти у вигляді логотипу.



The image shows a registration form with the following fields and values:

- Ім'я: Диана
- Прізвище: Титова
- 22.12.2000 (with a calendar icon)
- 3 (with a dropdown arrow)
- Б-ДН 416 (with a dropdown arrow)
- Кімната: 11
- Email: titovegmd@gmail.com
- Пароль: .....

At the bottom of the form is a green button labeled "НАДІСЛАТИ".

Рисунок 3.2. – Заповнена реєстраційна форма

На сторінці реєстраційної форми користувач може зареєструватися в платформу INBLOCK. Щоб почати реєстрацію потрібно натиснути на напис "Реєстрація" і заповнити відповідні поля. В імені і паролі можуть бути присутніми будь-які символи українського і англійського алфавітів, а також цифри, знаки пунктуації, дужки, підкреслення. Для завершення реєстрації потрібно натиснути на



напис "Відправити" Ім'я і пароль будуть відправлені на сервер, де відбудеться їх перевірка.

Після позитивного завершення перевірки імені та пароля Ви потрапите на сайт. У розділі "Профіль" можна ввести додаткову інформацію про себе, додати фотографію і виконати налаштування.



Рисунок 3.3. – Головна сторінка особистого кабінету

На зображенні 3.3 зображено головну сторінку сайту. Після реєстрації і входу користувач потрапляє в свій особистий кабінет, в якому потрапляє на головну сторінку. З лівого боку розташування функціонального меню сайту з доступними категоріями. На ній розташована актуальна інформація така як кількість користувачів, кількість новин і також розташований остання актуальна новина.

**IN BLOCK**

Микита Рудак

Головна

Профіль

Новини

Інформація

Зв'язок

© 2021 - 2021 INBLOCK

Ім'я Прізвище  
Микита Рудак

Рік народження  
2000.24.07

Курс  
4

Спеціальність  
Інженерія програмного забезпечення

Гуртожиток  
№ 1

Номер кімнати  
№ 16

Студент #001

Редагувати

Рисунок 3.4 –Сторінка «Профіль»

Друга сторінка в особистому кабінеті це сторінка профілю, при переході на яку користувач бачить інформацію, яку вводив при реєстрації на сайті. На цій сторінці можна підправити дані користувача, при помилковому заповненні. Також доступна завантаження зображення.

**IN BLOCK**

Микита Рудак

Головна

Профіль

Новини

Інформація

Зв'язок

© 2021 - 2021 INBLOCK

**Для іноземних студентів СНУ ім. В. Даля побудовано гуртожиток**

У Східноукраїнському національному університеті імені Володимира Даля здано в експлуатацію новий гуртожиток. Він був збудований спеціально для іноземних студентів, які навчаються в університеті. Тут проживатимуть 150 чол.

Відкрити

**Східноукраїнський національний університет імені Володимира Даля на 100% готовий до зими**

Східноукраїнський національний університет імені Володимира Даля має всі необхідні умови для початку опалювального сезону. Цього року у Далівському університеті капітально відремонтовано теплотрасу до гуртожитку №6.

Відкрити

**Дійсні реквізити СНУ ім. В.Даля для сплати за навчання (з 01.01.2021 року)**

Увага! Лійсні реквізити СНУ ім. В. Даля для сплати за навчання

Рисунок 3.5. –Сторінка «Новини»

Ця сторінка сайту на якій користувач може подивитися актуальні новини гуртожитку.

**IN BLOCK**

**Зворотній зв'язок**

Якщо у вас виникли якісь запитання чи пропозиції щодо розробки, то можете залишити своє питання  
 Bulvar Druzhby narodiv 41b, k16, +380994436308, blockin@gmail.com

Ім'я

Email

Ваше повідомлення

**НАДІСЛАТИ**

© 2021 - 2021 INBLOCK

Рисунок 3.6. –Сторінка зворотнього зв'язку

Остання сторінка особистого кабінету. Можна потрапити на форму зворотного зв'язку, де заповнивши невелику форму відправити свої побажання, питання, чи технічні проблеми.

### 3.3 Реалізація клієнтської частини

Незважаючи на безліч труднощів, пов'язаних із впровадженням програмні системи в середовищі World WideWeb, створення рішення веб-платформ за останні десять років сформували найбільше перспективний і динамічно розвивається напрямок сучасного індустрія розробки додатків.

Для розміщення сайту на хостингу було прийнято рішення використання VDS хостинг (<https://ru.000webhost.com/>). При використанні VDS ми отримуємо ряд переваг Сюди можна віднести підтримку PHP, MySQL, Curl, GD2, Zend, VML, fopen (), PHP-сокетів, завантажувача Ioncube і файлів .htaccess., Також цей хостинг безкоштовний.

### 3.3.1 Реалізація реєстрації користувача

Першим кроком при створенні реєстрації є створення бази даних MySQL. При роботі з базами даних MySQL використовуються найпопулярніші оператори мови запитів SQL. Таким чином для створення бази даних в СУБД MySQL використовуються два різні способи. Перший спосіб - це створення бази даних у консолі MySQL за допомогою оператора SQL CREATE DATABASE.

Далі представлена створення бази даних "Inblock", що складається з дев'яти таблиць: "id", "name", "surname", "data", "course", "hostel", "room", "email" і "password".

SQL запит для створення таблиці users.

```
CREATE TABLE `users` (
  `id` int NOT NULL,
  `name` varchar(20) NOT NULL,
  `surname` varchar(20) NOT NULL,
  `data` varchar(20) NOT NULL,
  `course` varchar(20) NOT NULL,
  `hostel` varchar(20) NOT NULL,
  `room` varchar(20) NOT NULL,
  `email` varchar(50) NOT NULL
)
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	<b>id</b>			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2	<b>name</b>	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	3	<b>surname</b>	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	4	<b>data</b>	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	5	<b>course</b>	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	6	<b>hostel</b>	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	7	<b>room</b>	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	8	<b>email</b>	utf8mb4_0900_ai_ci		Нет	Нет			
<input type="checkbox"/>	9	<b>password</b>	utf8mb4_0900_ai_ci		Нет	Нет			

Рисунок 3.7. – Структура таблиці «user»

Необхідно відзначити, що кожна з перерахованих таблиць бази даних MySQL створюється окремим оператором CREATE TABLE. Кожен стовпець містить ім'я, за яким слід опис типу даних і атрибути даного стовпчика. Деякі стовбці містять і інші специфікатори:

- NOT NULL означає, що всі рядки таблиці повинні мати значення в цьому атрибуті. Якщо NOT NULL не вказано, поле може бути порожнім;
- AUTO\_INCREMENT є спеціальною можливістю MySQL, яку можна задіяти в числових стовпцях. Якщо при вставці рядків в таблицю залишати таке поле порожнім, MySQL автоматично генерує унікальне значення ідентифікатора. Це значення буде на одиницю більше максимального значення, вже існуючого в стовпці. У кожній таблиці може бути не більше одного такого поля;
- PRIMARY KEY визначає, що цей стовпець є первинним ключем для таблиці і використовується після імені стовпчика. Дані в цьому стовпці повинні бути унікальними. MySQL автоматично індексує цей стовпець;
- UNSIGNED використовується після цілочисельного типу і означає, що його значення може бути або позитивним, або нульовим.

Структура sql-запитів які можуть використовуватися в БД для сортування та вибору інформації:

- SELECT ('стовпці або \* для вибору всіх стовпців; обов'язково')
- FROM ('таблиця; обов'язково')
- WHERE ('умова / фільтрація, наприклад, city ='Kiev'; необов'язково')
- GROUP BY ('стовпець, по якому хочемо згрупувати дані; необов'язково')
- HAVING ('умова / фільтрація на рівні згрупованих даних; необов'язково')
- ORDER BY ('стовпець, по якому хочемо впорядкувати висновок; необов'язково')

```

<?php
$server = 'localhost'; // Имя или адрес сервера
$user = 'mysql'; // Имя пользователя БД
$password = 'mysql'; // Пароль пользователя
$db = 'inblock'; // Название БД

$link = mysqli_connect($server, $user, $password, $db); // Подключение

// Проверка на подключение
if (!$link) {
    // Если проверку не прошло, то выводится надпись ошибки и заканчивается работа скрипта
    echo "Не удается подключиться к серверу базы данных!";
    exit;
}

```

Рисунок 3.8. – Файл для підключення до бази даних (db.php)

Основним методом, використовуваним в цьому скрипті (Рисунок 3.8.), є `mysqli_connect()`. Це внутрішня функція PHP для встановлення нового з'єднання з сервером MySQL[11].

На початку коду ми бачимо кілька оголошень змінних і значень, присвоєних цим змінним. Зазвичай це чотири змінні, щоб встановити правильне з'єднання з базою даних: `$server`, `$db`, `$user` і `$password`. У коді ми вказуємо наші дані для доступу до БД як значення для цих змінних, щоб їх можна було передати в функцію. Якщо спроба з'єднання була невдалою, виконується `if (!$link)`, що видає повідомлення про помилку підключення, яке ми прописали.

```

<?php
require_once ('db.php');

// Проверяем нажата ли кнопка отправки формы
if (isset($_REQUEST['doGo'])) {
    $name = $_REQUEST['name'];
    $surname = $_REQUEST['surname'];
    $yearbirth = $_REQUEST['yearbirth'];
    $course = $_REQUEST['course'];
    $hostel = $_REQUEST['hostel'];
    $room = $_REQUEST['room'];
    $email = $_REQUEST['email'];
    $password = $_REQUEST['password'];

    // Добавление пользователя
    mysqli_query($link, "INSERT INTO `users` (`id`, `name`, `surname`, `data`, `course`, `hostel`, `room`, `email`, `password`)
VALUES (NULL, '" . $name . "', '" . $surname . "', '" . $yearbirth . "', '" . $course . "', '" . $hostel . "',
'" . $room . "', '" . $email . "', '" . $password . "')");
}
?>

```

Рисунок 3.9. – Код реалізації реєстрації

На рисунку 3.9., представлений код реалізації реєстрації на сайті. Даний код виконаний на мові програмування php. І з першого рядка коду ми підключаємо наш файл db.php, який забезпечує підключення до бази даних і зберігає настройки підключення до неї. Далі ми перевіряємо натискання кнопки і передаємо в змінну значення, які ввів користувач. І останній етап це додавання даних в БД, за допомогою SQL запиту "INSERT INTO" ми вибираємо в які таблиці ми додаємо інформацію і після чого вказуємо змінні в яких і передавалася інформація з форми реєстрації.

```
<div class="form-message">
  <div class="h3 text-center hidden">
    <?php
      if (isset($_REQUEST['doGo'])) {
        echo 'Реєстрація пройшла успішно';
      }
    ?>
  </div>
</div>
```

Рисунок 3.10. – Код реалізації реєстрації

Наступний етап після натискання на відправки форми, виконується код, який перевіряє натиснення кнопки і після перевірки виводиться повідомлення "Реєстрація пройшла успішно", що дає зрозуміти користувачеві, що він успішно зареєстрований.

id	name	surname	data	course	hostel	room	email	password
13	Градов Тихон	Тихон	1999-06-22	4	Б-ДН 416	12	l1om1m@amazingmail.xyz	l1om1m@amazingmail.xyz
14	Смирнова	Лия	1999-06-10	4	Б-ДН 416	11	uptvya6@bestmail.monster	uptvya6@bestmail.monster
15	Аверьян	Щербakov	2003-11-23	2	Б-ДН 416	1	t7r0ybv@amazingmail.xyz	t7r0ybv@amazingmail.xyz
16	Харитина	Новикова	2004-04-05	1	Б-ДН 416	23	01keq0@amazingmail.xyz	01keq0@amazingmail.xyz
17	Диана	Титова	2000-12-02	4	М 24	23	426zdb@amazingmail	426zdb@amazingmail

Рисунок 3.11. – БД «user»

На рисунку 3.11. представлена заповнена таблиця користувачами, які зареєструвалися на сайті. Таким чином показана працездатність форми реєстрації та БД. Подальші дані можуть бути використані в доопрацюванні сайту і введення статистики.

Таким чином був розроблений повністю функціональна база даних з коректно заповненими таблицями і вмістом. Дану базу даних можливо вдосконалити до різних завдань і умов. Дані з БД можуть бути експортовані для подальшої обробки та введення статистики. Також є можливість коригування даних з самого інтерфейсу phpMyAdmin.



## ВИСНОВКИ

В процесі розробки дипломної роботи було проаналізовано інформаційну систему гуртожитку. Було зроблено створення веб-сайту, заповнення розділів веб-сайту навчального закладу гуртожитку "Східноукраїнській національний університету імені Володимира Даля".

На підставі аналізу предметної області був розроблений функціонал сайту.

Також, крім того розроблена і спроектована структура бази даних, була з врахуванням всіх необхідних умов. Проаналізувавши різні засоби СУБД, середовища програмування, була здійснена вибірка поміж наявних варіантів, і на основі першої розробленої структури, створена база даних.

Розроблений сайт відповідає всім нормам та вимоги, які були поставлені на етапі постановки завдання. При розробці веб-сайту були використані готові модулі аутентифікації.

Дані модулі були доопрацьовані з урахуванням специфіки web-сайту і успішно впроваджені в його структуру.

Для подальшого вдосконалення web-сайту представляється можливим розробка модулів доступу до систем дистанційної освіти. Так само можлива доробка з метою подальшого підвищення інформативності, та зручності інтерфейсу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хомоненко А.Д., Циганков В.М., Мальцев М.Г. Базы данных: Пособие для высших учебных заведений / За ред. проф. А.Д. Хомоненко. - СПб: КОРОНА-принт, 2000..
2. Т. Тиори, Дж. Фрай. Проектирование структур баз данных, т. 1. - М.: Мир. Тисячі дев'яност вісімдесят п'ять.
3. Ульман Дж. Основы систем баз данных. - М.: Финансы и статистика, 1983.
4. Коннолли Т., Бегг К., Страчан Л. Базы данных: проектирование, реализация и сопровождение. Теория и практика. 2-е издание.: Пер. с англ. - М.: Издательский дом «Вильямс», 2001.
5. Дейт К.Дж. Введение в системы баз данных.: Пер. с англ. - К.: Диалектика, 1998..
6. Діго С.М. Проектирование и использование баз данных. - VI: Финансы и статистика, 1995.
7. Наго М. Катаяма Т., Уемура С. Структуры и базы данных: Пер. с япон. - М.: Мир, 1986.
8. Що таке конструктор веб-сайтів? - [websitetooltester.com](http://www.websitetooltester.com) [Електронний ресурс] - Режим доступу: - <https://www.websitetooltester.com>
9. HTML & CSS - [www.w3.org](http://www.w3.org) [Електронний ресурс] - Режим доступу: <https://www.w3.org/standards/webdesign/htmlcss.html>
10. 15 Основних вимог до веб сайту - [studfile.net](http://studfile.net) [Електронний ресурс] - <https://timeweb.com> <https://studfile.net/preview/6438407/page:3/>
11. Що таке MySQL? - [atlantic.net](http://atlantic.net) [Електронний ресурс] - Режим доступу: <https://www.atlantic.net/what-is-mysql/>
12. Переваги побудови динамічних сайтів - [hackernoon.com](http://hackernoon.com) [Електронний ресурс] - Режим доступу: <https://hackernoon.com/the-advantages-ofbuilding-dynamic-websites-bc914071a155>

13. Introduction. What is Apollo Server and what does it do? [Електронний ресурс]. - Режим доступу: <https://www.apollographql.com/docs/apollo-server/>. - Дата звернення: 30.05.2019 р.
14. Server-side web frameworks [Електронний ресурс]. - Режим доступу: <https://developer.mozilla.org/en-US/docs/Learn/Server->
15. Модель «сутність — зв'язок» [Електронний ресурс]. - Режим доступу: [https://uk.wikipedia.org/wiki/Модель\\_«сутність\\_—\\_зв'язок»](https://uk.wikipedia.org/wiki/Модель_«сутність_—_зв'язок»). - Дата звернення: 20.05.2019 р.
16. Як створити логотип [Електронний ресурс]. - Режим доступу: <https://webkitchen.kiev.ua/ua/blog-ua/kak-sozdat-logotip-5-osnovnyh-printsipov>. - Дата звернення: 21.05.2021 р.
17. Як зробити - Форму реєстрації [Електронний ресурс]. - Режим доступу: [https://html5css.ru/howto/howto\\_css\\_register\\_form.php](https://html5css.ru/howto/howto_css_register_form.php). - Дата звернення: 21.05.2021 р.
18. Запит SQL [Електронний ресурс]. - Режим доступу: <https://mykonspekts.ru/2-3242.html>. - Дата звернення: 21.05.2021 р.
19. Про затвердження Примірного положення про студентський гуртожиток вищого навчального закладу [Електронний ресурс]. - Режим доступу: [https://osvita.ua/legislation/Vishya\\_osvita/3142/](https://osvita.ua/legislation/Vishya_osvita/3142/). - Дата звернення: 21.05.2021 р.

## **ДОДАТКИ**

## ДОДАТОК А. ЗМІСТ ФАЙЛУ SCRIPTS.JS

```
(function($) {  
    "use strict";  
  
    /* Preloader */  
    $(window).on('load', function() {  
        var preloaderFadeOutTime = 500;  
        function hidePreloader() {  
            var preloader = $('.spinner-wrapper');  
            setTimeout(function() {  
                preloader.fadeOut(preloaderFadeOutTime);  
            }, 500);  
        }  
        hidePreloader();  
    });  
  
    /* Navbar Scripts */  
    // jQuery to collapse the navbar on scroll  
    $(window).on('scroll load', function() {  
        if ($(".navbar").offset().top > 20) {  
            $(".fixed-top").addClass("top-nav-collapse");  
        } else {  
            $(".fixed-top").removeClass("top-nav-collapse");  
        }  
    });  
  
    // jQuery for page scrolling feature - requires jQuery Easing plugin  
    $(function() {  
        $(document).on('click', 'a.page-scroll', function(event) {
```

```

    var $anchor = $(this);
    $('html, body').stop().animate({
        scrollTop: $($anchor.attr('href')).offset().top
    }, 600, 'easeInOutExpo');
    event.preventDefault();
});
});

```

```

// closes the responsive menu on menu item click
$(".navbar-nav li a").on("click", function(event) {
    if (!$(this).parent().hasClass('dropdown'))
        $(".navbar-collapse").collapse('hide');
});

```

```

/* Rotating Text - Morphtext */
$("#js-rotating").Morphtext({
    // The [in] animation type. Refer to Animate.css for a list of available
    animations.
    animation: "fadeIn",
    // An array of phrases to rotate are created based on this separator. Change
    it if you wish to separate the phrases differently (e.g. So Simple | Very Doge | Much Wow
    | Such Cool).
    separator: "|",
    // The delay between the changing of each phrase in milliseconds.
    speed: 2000,
    complete: function () {
        // Called after the entrance animation is executed.
    }
}

```

```
});

/* Card Slider - Swiper */
var cardSlider = new Swiper('.card-slider', {
  autoplay: {
    delay: 4000,
    disableOnInteraction: false
  },
  loop: true,
  navigation: {
    nextEl: '.swiper-button-next',
    prevEl: '.swiper-button-prev'
  },
  slidesPerView: 3,
  spaceBetween: 20,
  breakpoints: {
    // when window is <= 992px
    992: {
      slidesPerView: 2
    },
    // when window is <= 768px
    768: {
      slidesPerView: 1
    }
  }
});
```

```
/* Image Slider - Swiper */  
var imageSlider = new Swiper('.image-slider', {  
  autoplay: {  
    delay: 2000,  
    disableOnInteraction: false  
  },  
  loop: false,  
  navigation: {  
    nextEl: '.swiper-button-next',  
    prevEl: '.swiper-button-prev',  
  },  
  spaceBetween: 30,  
  slidesPerView: 5,  
  breakpoints: {  
    // when window is <= 380px  
    380: {  
      slidesPerView: 1,  
      spaceBetween: 10  
    },  
    // when window is <= 516px  
    516: {  
      slidesPerView: 2,  
      spaceBetween: 10  
    },  
    // when window is <= 768px  
    768: {  
      slidesPerView: 3,  
      spaceBetween: 20  
    },  
  },  
}
```



```

// when window is <= 992px
992: {
    slidesPerView: 4,
    spaceBetween: 30
},
// when window is <= 1200px
1200: {
    slidesPerView: 5,
    spaceBetween: 30
},
}
});

```

```

/* Image Slider - Magnific Popup */
$('.popup-link').magnificPopup({
    removalDelay: 300,
    type: 'image',
    callbacks: {
        beforeOpen: function() {
            this.st.image.markup = this.st.image.markup.replace('mfp-
figure', 'mfp-figure ' + this.st.el.attr('data-effect'));
        },
        beforeClose: function() {
            $('.mfp-figure').addClass('fadeOut');
        }
    },
    gallery: {
        enabled: true //enable gallery mode
    }
});

```

```

    }
  });

  /* Video Lightbox - Magnific Popup */
  $('.popup-youtube, .popup-vimeo').magnificPopup({
    disableOn: 700,
    type: 'iframe',
    mainClass: 'mfp-fade',
    removalDelay: 160,
    preloader: false,
    fixedContentPos: false,
    iframe: {
      patterns: {
        youtube: {
          index: 'youtube.com/',
          id: function(url) {
            var m = url.match(/[\?&]v=([^\?&]+)/);
            if ( !m || !m[1] ) return null;
            return m[1];
          },
          src: 'https://www.youtube.com/embed/%id%?autoplay=1'
        },
        vimeo: {
          index: 'vimeo.com/',
          id: function(url) {
            var m = url.match(/(https?:\:\/\/)?(www.)?(player.)?vimeo.com\/([a-z]*\/)*([0-9]{6,11})[?].*/);
            if ( !m || !m[5] ) return null;

```

```

        return m[5];
    },
    src: 'https://player.vimeo.com/video/%id%?autoplay=1'
}
}
}
});

```

```

/* Lightbox - Magnific Popup */
$('.popup-with-move-anim').magnificPopup({
    type: 'inline',
    fixedContentPos: false, /* keep it false to avoid html tag shift with margin-
right: 17px */
    fixedBgPos: true,
    overflowY: 'auto',
    closeBtnInside: true,
    preloader: false,
    midClick: true,
    removalDelay: 300,
    mainClass: 'my-mfp-slide-bottom'
});

```

```

/* Counter - CountTo */
var a = 0;
$(window).scroll(function() {
    if ($('#counter').length) { // checking if CountTo section exists in the page,
if not it will not run the script and avoid errors

```

```

var oTop = $('#counter').offset().top - window.innerHeight;
if (a == 0 && $(window).scrollTop() > oTop) {
    $('.counter-value').each(function() {
        var $this = $(this),
            countTo = $this.attr('data-count');
        $({
            countNum: $this.text()
        }).animate({
            countNum: countTo
        },
        {
            duration: 2000,
            easing: 'swing',
            step: function() {
                $this.text(Math.floor(this.countNum));
            },
            complete: function() {
                $this.text(this.countNum);
                //alert('finished');
            }
        });
    });
    a = 1;
}
});

```

*/\* Move Form Fields Label When User Types \*/*

```
// for input and textarea fields
$("input, textarea").keyup(function(){
    if ($(this).val() != "") {
        $(this).addClass('notEmpty');
    } else {
        $(this).removeClass('notEmpty');
    }
});

/* Contact Form */
$("#contactForm").validator().on("submit", function(event) {
    if (event.isDefaultPrevented()) {
        // handle the invalid form...
        cformError();
        csubmitMSG(false, "Please fill all fields!");
    } else {
        // everything looks good!
        event.preventDefault();
        csubmitForm();
    }
});

function csubmitForm() {
    // initiate variables with form content
    var name = $("#cname").val();
    var email = $("#cemail").val();
    var message = $("#cmessage").val();
    var terms = $("#cterms").val();
```

```

$.ajax({
  type: "POST",
  url: "php/contactform-process.php",
  data: "name=" + name + "&email=" + email + "&message=" + message +
"&terms=" + terms,
  success: function(text) {
    if (text == "success") {
      cformSuccess();
    } else {
      cformError();
      csubmitMSG(false, text);
    }
  }
});
}

```

```

function cformSuccess() {
  $("#contactForm")[0].reset();
  csubmitMSG(true, "Message Submitted!");
  $("input").removeClass('notEmpty'); // resets the field label after submission
  $("textarea").removeClass('notEmpty'); // resets the field label after
submission
}

```

```

function cformError() {
  $("#contactForm").removeClass().addClass('shake
animated').one('webkitAnimationEnd      mozAnimationEnd      MSAnimationEnd
oanimationend animationend', function() {
    $(this).removeClass();

```

```
});
}
```

```
function csubmitMSG(valid, msg) {
  if (valid) {
    var msgClasses = "h3 text-center tada animated";
  } else {
    var msgClasses = "h3 text-center";
  }
  $("#cmsgSubmit").removeClass().addClass(msgClasses).text(msg);
}
```

```
/* Privacy Form */
$("#privacyForm").validator().on("submit", function(event) {
  if (event.isDefaultPrevented()) {
    // handle the invalid form...
    pformError();
    psubmitMSG(false, "Please fill all fields!");
  } else {
    // everything looks good!
    event.preventDefault();
    psubmitForm();
  }
});
```

```
function psubmitForm() {
  // initiate variables with form content
  var name = $("#pname").val();
```

```

        var email = $("#pemail").val();
var select = $("#pselect").val();
var terms = $("#pterms").val();

$.ajax({
    type: "POST",
    url: "php/privacyform-process.php",
    data: "name=" + name + "&email=" + email + "&select=" + select +
"&terms=" + terms,
    success: function(text) {
        if (text == "success") {
            pformSuccess();
        } else {
            pformError();
            psubmitMSG(false, text);
        }
    }
});
}

```

```

function pformSuccess() {
    $("#privacyForm")[0].reset();
    psubmitMSG(true, "Request Submitted!");
    $("input").removeClass('notEmpty'); // resets the field label after submission
}

```

```

function pformError() {

```



```

    $("#privacyForm").removeClass().addClass('shake
animated').one('webkitAnimationEnd      mozAnimationEnd      MSAnimationEnd
oanimationend animationend', function() {
        $(this).removeClass();
    });
}

```

```

function psubmitMSG(valid, msg) {
    if (valid) {
        var msgClasses = "h3 text-center tada animated";
    } else {
        var msgClasses = "h3 text-center";
    }
    $("#pmsgSubmit").removeClass().addClass(msgClasses).text(msg);
}

```

```

/* Back To Top Button */
// create the back to top button
$('body').prepend('<a href="body" class="back-to-top page-scroll">Back to
Top</a>');
var amountScrolled = 700;
$(window).scroll(function() {
    if ($(window).scrollTop() > amountScrolled) {
        $('a.back-to-top').fadeIn('500');
    } else {
        $('a.back-to-top').fadeOut('500');
    }
});

```

```
/* Removes Long Focus On Buttons */  
$(".button, a, button").mouseup(function() {  
    $(this).blur();  
});  
  
})(jQuery);
```