

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської роботи  
на ступінь вищої освіти бакалавр

на тему: **«СТВОРЕННЯ ВЕБ-СЕРВІСУ ДЛЯ ПЛАНУВАННЯ ВІЛЬНОГО ЧАСУ З  
ВИКОРИСТАННЯМ PHP/JS»**

Виконав: студент 4 курсу, групи ПД-41  
спеціальності: 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)

Немчин С.В.  
(прізвище та ініціали)

Керівник Дібрівний О.А.  
(прізвище та ініціали)

Рецензент \_\_\_\_\_  
(прізвище та ініціали)

Нормоконтроль \_\_\_\_\_  
(прізвище та ініціали)

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 «Інженерія програмного забезпечення»

### ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ О.В. Негоденко

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 року

## ЗАВДАННЯ

### НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Створення веб-сервісу для планування вільного часу з використанням PHP/JS»

Керівник роботи ст. викладач Дібрівний О.А.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “12” березня року №65.

2. Строк подання студентом роботи 01.06.2021.

3. Вхідні дані до роботи:

3.1. Положення розробки веб-сервісів;

3.2. Методи побудови веб-сервісів;

3.3. Розробка моделі веб-сервісу для планування вільного часу;

3.4. Науково-технічна література.

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

4.1. Загальні положення розробки веб-сервісу для планування вільного часу;

4.2. Аналіз технології і методів створення веб-сервісу;

4.3. Розробка веб-сервісу для планування вільного часу з використанням PHP та JS;

4.4. Висновки

5. Перелік графічного матеріалу.

5.1. Основні характеристики роботи;

5.2. Актуальність задачі;

5.3. Дослідження поняття веб-сервісу;

5.4. Аналіз інструментів розробки;

- 5.5. Відбір необхідних інструментів для проекту;
- 5.6. Формування технічного завдання;
- 5.7. Розробка дизайну веб-сервісу;
- 5.8. Створення проекту веб-сервісу;
- 5.10. Порівняння якостей отриманих результатів;
- 5.11. Дослідження швидкості функціонування веб-сервісу;
- 5.13. Висновки.

6. Дата видачі завдання 19.04.2021

### КАЛЕНДАРНИЙ ПЛАН

№ з / п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз науково-технічної літератури	19.04.21	Виконано
2	Дослідження основних аспектів веб-розробки	20.04.21-21.05.21	Виконано
3	Аналіз методів створення веб-сервісів	23.04.21-25.04.21	Виконано
4	Розробка веб-сервісу для планування вільного часу	25.04.21-20.05.21	Виконано
5	Висновки, оформлення роботи	20.05.21-23.05.21	Виконано
6	Розробка демонстраційних матеріалів	24.05.21	Виконано
7	Здача роботи	25.05.21	

Студент \_\_\_\_\_ Немчин С.В.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Дібрівний О.А.  
підпис ) (прізвище та ініціали)





## РЕФЕРАТ

Текстова частина магістерської роботи 41 с., 8 рис., 20 джерел.

ВЕБ-РОЗРОБКА, МОВА ПРОГРАМУВАННЯ, ВЕБ-СЕРВІС,  
АЛГОРИТМ, МЕТОД, РОЗРОБКА, ДИЗАЙН.

*Об'єктом дослідження* - є підвищення ефективності планування вільного часу з використанням PHP/JS.

*Мета роботи* – розробка веб-сервісу для планування вільного часу з використанням PHP/JS, призначеного для вибору потенційними клієнтами з спектру послуг з вибору місць, де користувач зможе провести вільний час.

*Предмет дослідження* – технологія розробки веб-сервісу для планування вільного часу.

*Методи дослідження.* У науковій роботі було використано як загальнонаукові, так і спеціальні методи дослідження, до яких можна віднести: методи аналізу та синтезу, індукції та порівняння під час дослідження основ веб-розробки, метод єдності історичного та логічного було використано для відображення головних етапів створення веб-сервісів, а також, методи та алгоритми побудови для розробки проекту, а також графічні методи – для представлення результатів дослідження у графічному вигляді.

Ключові слова: PHP, JS, веб-сервіс

## Зміст

<b>ВСТУП</b> .....	8
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ WEB-СЕРВІСІВ</b> .....	10
<b>1.1 Основні поняття розробки веб-сервісів</b> .....	10
<b>1.2 Формалізований опис технічного завдання проекту</b> .....	17
<b>РОЗДІЛ 2. ОПИС ПРОГРАМНИХ ЗАСОБІВ</b> .....	23
<b>2.1 Засоби розробки веб-сервісів</b> .....	23
<b>2.2 PHP</b> .....	25
<b>2.3 JavaScript</b> .....	27
<b>2.4 Мова розмітки HTML</b> .....	28
<b>2.5 Інші програмні засоби</b> .....	29
<b>РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА</b> .....	32
<b>3.1 Модельне подання об'єкта розробки</b> .....	32
<b>3.2 Опис продукту та результатів розробки</b> .....	37
<b>ВИСНОВОК</b> .....	48
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	50
<b>ДОДАТКИ</b> .....	53

## ВСТУП

Наш час називають «інформаційним століттям». Ця назва виникла тому, що найважливішим, цінним і необхідним ресурсом є інформація. Володіння інформацією і вміння своєчасно, швидко і чітко донести необхідну інформацію до клієнтів - ключ до успіху в реаліях сучасного бізнесу. На даний момент саме Інтернет, здатний оперативно і масово передавати текст, звук, зображення і навіть відео-ролики, тобто практично будь-яку інформацію.

С точки зору користувача, Інтернет - це величезний інформаційний ресурс, в якому можна знайти все, що завгодно: від прогнозу погоди до особистих переваг голлівудських кінозірок. Аудиторія мережі Інтернет зростає з кожним роком, Інтернет - саме швидко розвивається засіб передачі інформації за все історію людства. Користувачі мережі інтернет є метою особливого інтересу для рекламодавців, тому що факт залишається фактом - серед користувачів мережі купа потенційних клієнтів.

Сфера послуг є однією з найперспективніших сфер економіки і економічної діяльності. Вона охоплює широкий спектр економічної діяльності і за своїм функціональним призначенням не є єдиним комплексом. У процесі комерціалізації української економіки і при українських реаліях: великі відстані, гостра нестача часу; особливого значення набуває така важлива форма реалізації товарів і послуг, як торгівля і реклама через мережу Інтернет.

Існує величезна кількість способів комерційного підходу до мережі інтернет. У мережі можна рекламувати послуги, продавати товари або представляти споживачеві і те й інше. У сучасному бізнесі багато що залежить від самопрезентації компанії, її позиціонування на ринку послуг, що надаються і здатності шукати нових клієнтів і ринки збуту. Одним з інструментів, як іміджевих, так і маркетингових, є наявність свого сайту в мережі Інтернет. З точки зору бізнесу. Інтернет - це сучасна рекламна майданчик, що дозволяє забезпечити приплив клієнтів. Web-сайт перетворює



компанію в сучасний бізнес. Web-сайт компанії є найважливішим джерелом інформації для потенційних клієнтів і людей, чия думка є суспільно значущим.

*Актуальність* даної роботи обумовлена затребуваністю якісного веб-сервісу для планування вільного часу з використанням PHP/JS. Спосіб залучення клієнтів за допомогою web-сервісу відрізняється відносно низькими витратами і великою кількістю цільової аудиторії.

*Об'єктом дослідження* - є підвищення ефективності планування вільного часу з використанням PHP/JS.

*Предмет дослідження* – є технологія розробки веб-сервісу для планування вільного часу.

*Мета дослідження:* розробка веб-сервісу для планування вільного часу з використанням PHP/JS, призначеного для вибору потенційними клієнтами з спектру послуг з вибору місць, де користувач зможе провести вільний час.

Відповідно до поставленої мети в роботі визначені наступні завдання:

- вивчити технології по розробки сервісів;
- визначити цілі, ідеї, потреби web-сервісів;
- розробити технічну концепцію сервісу (структуру);
- розробити технічне завдання на створення web-сервісу.

Результатом роботи є веб-сервіс для планування вільного часу з використанням PHP/JS. Сервіс знаходиться в стадії

## РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ WEB-СЕРВІСІВ

### 1.1 Основні поняття розробки веб-сервісів

Веб-сервіси представляються як нова різновид веб-додатків для розміщення різних служб у різних мережевих місцях, за допомогою яких бізнес взаємодіє за допомогою спеціального підходу, який використовує переваги базової Інтернет-інфраструктури. Найважливішою особливістю Веб-сервісів є те, що комунікація серед них не залежить від платформи, на якій вони розроблені (використовуються операційна система та мова програмування). Оскільки вона створює спосіб обміну та розповсюдження інформації в глобальному масштабі. Веб-служба виникає як систематична та розширювана структура взаємодії додатків із додатками, заснована на існуючих веб-протоколах та відкритому XML-стандарті. [13] Після того, як веб-служба розгорнута, її інтерфейс та прив'язка можуть бути ідентифіковані, описано, виявлений, а потім викликаний іншими WebApplications.

Веб-сервіс забезпечує кращий зв'язок між кількома обчислювальними машинами, платформами, програмами та діловими партнерами. Ділова діяльність тепер може здійснюватися незалежно від збирання, а вирощені веб-сервіси дозволяють досягти автоматичної та динамічної взаємодії між бізнес-системами для дуже ефективного виконання бізнес-завдань. Поширення ділових послуг через Інтернет дозволяє різноманітним комунікаційним пристроям отримувати до них доступ і створює глобальну платформу для приватних осіб, малих та середніх підприємств та організацій для здійснення комерційної діяльності та надання послуг із доданою вартістю.

Веб-сервіси лежать в тренді, щоб організації дотримувались того, що називається архітектурою, орієнтованою на послуги, що є принципом організації технічної інфраструктури організації для забезпечення потреб користувачів програмного забезпечення організації. Зокрема у великих

організаціях SOA, як його часто називають, став домінуючим підходом до організації інформаційно-технологічної (IT) інфраструктури. Загалом, SOA включає в себе організований масив незалежних програмних компонентів, кожен з яких забезпечує чітко визначену, самостійну одиницю функціональності. SOA базується на давніх тенденціях до розподілених обчислень та забезпечує будівельні блоки сучасного середовища, придатного для обчислювальної епохи, в якій панує Інтернет.

Веб-сервіси забезпечують засоби для реалізації цієї архітектури, орієнтованої на послуги. Хоча теоретично для побудови SOA можуть бути використані інші технології, веб-служби виступають домінуючим підходом і включають набір протоколів та стандартів, заснованих на міжнародно встановлених сумісних протоколах, які забезпечують технологічну структуру, що відповідає SOA.

Однак не всі реалізації веб-сервісів піднімаються до рівня SOA. Організації зі складними обчислювальними середовищами можуть розробляти стратегії, які організовують свою технічну інфраструктуру в модулі та компоненти, які взаємодіють та працюють через веб-служби. Враховуючи той факт, що багато організацій (деякі з яких мають більшу здатність підтримувати веб-служби, ніж інші) працюють із різноманітними програмними додатками, SOA, швидше за все, є цілком більше, ніж готовим продуктом.

Веб-сервіси також можуть бути розгорнуті для виконання певних функцій, не обов'язково вимагаючи, щоб вся навколишня інфраструктура базувалася на SOA. За потребою може бути реалізований окремий набір веб-служб, і з часом інфраструктура організації може перерости у більш повноцінний SOA.

Вираз «веб-сервіс» може призвести до певної плутанини, тому необхідно розглянути кілька концепцій, які легко сприйняти як веб-сервіс, але насправді не входять до сфери веб-сервісів.

Наприклад, під час обговорення веб-сервісів ті, хто авторитетно обговорює концепцію, говорять не лише про системи з веб-інтерфейсом. Веб-інтерфейси вирішують проблеми інтерфейсу відображення інформації та взаємодії з користувачами через веб-браузер, який став домінуючим інструментом доступу до інформації та експлуатації програмних додатків. Більшість користувачів не тільки отримують доступ до основних веб-сторінок через веб-браузери, але і все більша кількість програмних програм також покладається на веб-інтерфейс (а не на графічні програми або вікна, такі як ті, які будуть використовуватися для роботи в настільних програмах на комп'ютерах, на яких працює або операційна система Windows або Macintosh). Завдяки таким технологіям, як AJAX (асинхронний JavaScript та XML), веб-інтерфейси можуть надавати майже всі функції інтерфейсу користувача, які раніше були доступні лише через настільні програми. Але веб-інтерфейси не є веб-службами.

Інтернет-інформаційні ресурси також не є обов'язково веб-службами. Те, що вміст може бути доставлений через Інтернет, не робить його веб-службою. Інформація, представлена через веб-браузер - через статичні веб-сторінки чи доставлена динамічно із базової бази даних або сховища, не є веб-службою.

Користувачі комп'ютерів дедалі частіше отримують професійні послуги - такі як Інтернет-банкінг, бронювання подорожей, послуги з картографування та напрямків подорожей, послуги підтримки клієнтів та Інтернет-магазини, лише згадавши декілька - через веб-системи.

Крім того, веб-сервіси не слід плутати з поняттями «програмне забезпечення як послуга» або «постачальник послуг додатків». Ці все більш популярні моделі для розгортання програмного забезпечення включають розміщення програми на віддаленому сервері та надання своїм користувачам доступу через Інтернет. Одна з ключових переваг цієї моделі програмного забезпечення полягає в тому, що клієнтам не потрібно встановлювати та

підтримувати програмне забезпечення на локальних серверах або настільних системах. Проте програмне забезпечення як послуга - це не те саме, що веб-служби.

Нарешті, у багатьох бібліотеках працевлаштована людина, яка займає посаду бібліотекаря веб-сервісу. Як правило, ця посада відповідає за розробку та обслуговування веб-сервісу бібліотеки, що включає доставку інформаційних ресурсів та бібліотечні послуги, що базуються на Інтернеті. Однак у більшості випадків бібліотекар веб-сервісів, швидше за все, не буде задіяний у технологіях веб-сервісів, про які йдеться у цій роботі.

Кожне із згаданих раніше понять стосується інтерфейсу Інтернету, Інтернету, який "бачить" користувач, наприклад сторінки Google із результатами пошуку. Веб-сервіси, скоріше, є закулісною технологією; вони забезпечують частину водопроводу, необхідного для підключення комп'ютерних систем. Ці системи можуть часто мати веб-інтерфейс, але вони не обов'язково повинні мати його. Веб-служби працюють на одних і тих же протоколах нижчого рівня, таких як TCP / P та HTTP, пов'язаних з Інтернетом та Всесвітньою павутиною.

Не всі засоби зв'язку між комп'ютером і кваліфікуються як веб-служби. Незважаючи на те, що вони мають певну гнучкість, веб-служби дотримуються певної архітектурної бази та передбачають використання певного набору протоколів та стандартів. Наприклад, EDI (електронний обмін даними) - це взаємодія між комп'ютером, яка слідує за протоколами, які не вважаються веб-послугами.

Не всі засоби зв'язку між комп'ютером і кваліфікуються як веб-служби. Незважаючи на те, що вони мають певну гнучкість, веб-служби дотримуються певної архітектурної бази та передбачають використання певного набору протоколів та стандартів. Наприклад, EDI (електронний обмін даними) - це

взаємодія між комп'ютером, яка слідує за протоколами, які не вважаються веб-послугами.

Структура веб-служб забезпечує деякі основні компоненти електронної комерції. Для комерційних операцій часто потрібна функція вилучення інформації з кількох джерел, наприклад, коли користувач укладає транзакцію електронної комерції через веб-сайт певної організації. У процесі транзакції програмі на оригінальному сервері може знадобитися зв'язуватися з іншими програмами, або в межах своєї інфраструктури, або із зовнішніх джерел. Для того, щоб завершити свою роботу, програмі може знадобитися знайти інші програми, які містять необхідну інформацію, надіслати запит, чекати та отримати відповідь. Можуть мати місце кілька послідовностей запит-відповідь, і можуть бути задіяні кілька постачальників. Усі ці закулісні транзакції веб-сервісу відбуваються невідомо для користувача. Одна з красень веб-служб полягає в їх здатності об'єднувати ресурси багатьох комп'ютерних систем для виконання складних завдань без втручання користувача.

W3C, група, відповідальна за розробку веб-служб, визначає веб-послуги таким чином:

Веб-сервіс - це програмна система, призначена для підтримки взаємодії міжмережевої взаємодії через мережу. Він має інтерфейс, описаний у машинно-обробному форматі (зокрема, WSDL). Інші системи взаємодіють з веб-службою способом, передбаченим її описом, із використанням повідомлень SOAP, як правило, переданих за допомогою HTTP із серіалізацією XML разом із іншими веб-стандартами.

Однією з ключових тенденцій у технології є розподілені обчислення. Замість того щоб створювати монолітні, автономні програми, розробники часто воліють створювати декілька менших систем, кожна з яких має спеціалізовану функцію. Кожен компонент виконує спеціалізоване, чітко визначене завдання. Часто простіше розробити групу невеликих

спеціалізованих компонентів, ніж єдину складну систему. Ці невеликі компоненти часто можна використовувати в багатьох додатках, заощаджуючи значні ресурси для розвитку в довгостроковій перспективі. Крім того, ці невеликі, автономні компоненти можна вважати веб-службами, коли вони реалізують робочі процеси та стандарти, пов'язані з цим середовищем.

Веб-служби також можуть вдихнути нове життя у старіші програми. Старі системи, часто побудовані з використанням інструментів та технологій, які сьогодні вважаються застарілими, можуть бути прийняті для участі в сервісно-орієнтованому середовищі.

Модель веб-сервісів розроблена таким чином, щоб байдуже ставитися до того, як виконується будь-яка обчислювальна задача. Програмна програма може сприйматись як “обгорнута” веб-службою; іншими словами, на рівень веб-сервісу не впливає те, як виконується певне завдання чи яке апаратне чи програмне забезпечення працює. Робота веб-служби полягає в передачі запиту на виконання певного завдання, а потім у поверненні результатів.

Хоча нова програма, ймовірно, буде спеціально розроблена для роботи як веб-сервіс, застарілі програми можна перетворити на веб-служби, просто створивши рівні комунікації, які дозволяють протоколи веб-сервісу. Розробляючи шлюзи веб-сервісів, організації можуть допомогти зберегти інвестиції, спрямовані на розробку програм попереднього покоління програм. Архітектура внутрішньої бази даних та вміст та бізнес-логіка застарілої системи можуть продовжувати працювати як і раніше, вхід і вихід програми передаються через рівні веб-сервісу.

Веб-сервіси приховують базове апаратне та програмне забезпечення від очей, а модель веб-служб полегшує взаємодію, нейтралізуючи відмінності в апаратних платформах, операційних системах та засобах розробки. Після того, як ці компоненти загорнуті в протоколи веб-сервісу, усі компоненти програмного забезпечення здаються однаковими для користувачів у

«зовнішньому світі». Крім того, веб-сервіси можуть бути реалізовані на всіх апаратних та програмних платформах - Linux, Unix, Mac або Windows. Незважаючи на те, що можливо запустити веб-служби на настільних системах, більшість з них працюють на апаратних та операційних системах класу сервера.

Веб-сервіси можуть працювати в режимі без стану, або їх можна запрограмувати на роботу в межах певних, визначених сеансів. Багато простих веб-служб працюють незалежно від будь-якого конкретного контексту або складної серії подій. Цей тип веб-сервісу складається з простої, дискретної транзакції запит-відповідь. Однак за деяких обставин важливо підтримувати сеанс між учасниками операції Веб-сервісів, щоб можна було організувати кілька запитів Веб-сервісу, тобто тих, що з'єднані в складній транзакції.

У розподіленому обчислювальному середовищі жоден комп'ютерний додаток, ймовірно, не буде повністю незалежним. Часто потрібно, щоб один комп'ютерний компонент покладався на інші для отримання конкретних елементів інформації або для обчислення або перетворення. Коли програмному додатку потрібна інформація для виконання своїх функцій, веб-служби забезпечують механізм запиту цієї інформації, спосіб переміщення цієї інформації та спосіб структурування цієї інформації для транзиту.

Веб-сервіси забезпечують рівень комунікаційних технологій, що визнає взаємозв'язок організацій та пов'язаних із ними сфер знань, інформації та послуг. Це технологія, яка дозволяє організаціям краще виконувати свої ролі постачальників послуг та надає можливості для створення послуг із доданою вартістю на основі покращення послуг, отриманих від інших.

Веб-сервіси можна вважати зрілою та усталеною обчислювальною моделлю. Як технологія веб-сервіси з'явилися наприкінці 90-х років минулого століття і набирають популярності з того часу. Відповідні стандарти чітко визначені та широко прийняті, і існує безліч інструментів для розробки веб-



сервісу. Це не технологія очікування; В даний час веб-послуги є загальнодоступними, і їх можна вважати безпечною інвестицією для бібліотек.

## **1.2 Формалізований опис технічного завдання проекту**

Розробка web-сервіс для планування вільного часу виконана в рамках випускний кваліфікаційної роботи (назва навчального закладу)

Назва продукту розробки (проектування). Web-сервіс для планування вільного часу.

Призначення продукту. Web-сервіс для для планування вільного часу призначений для надання інформації про місця та послуги для відпочинку і для взаємодії клієнтів: замовників і виконавців.

Web-сервіс повинен містити наступні функціональні можливості:

- реєстрація користувачів;
- налаштування облікових записів користувача;
- перегляд списку послуг та місць;
- вибір послуги або місця;
- замовлення послуги;
- наявність середовища для віддаленого управління web-сервісом і його адмініструванням, і обслуговуванням.

Характеристика області застосування продукту.

Процеси і структури, в яких передбачається використання продукту розробки. Web-сервіс буде розташовуватися в загальному доступі для всіх користувачів мережі Інтернет.

Характеристика персоналу (кількість, кваліфікація, ступінь готовності).  
Адміністратор - всі заявки на затвердження послуг проходять через адміністратора даного web-сервісу, нові користувачі повинні пройти перевірку адміністратора, щоб новий обліковий запис вступив в роботу її повинен активувати адміністратор. Адміністратор стежить за роботою решти персоналу.

Необхідні професійні навички:

- Знання HTML, CSS, PHP, JavaScript;
- Уміння складати технічні завдання на розробку інтернет-сервісів;
- Володіння методами пошукової оптимізації;
- Знання основ забезпечення комплексної безпеки веб-сайтів;
- Базові знання PHP;
- Система управління контентом, CMS.

Модератори - модератори займаються оформленням web-сервісу і оновленням інформації в web-сервісі. Кількість модераторів - 2 людини. Модератор повинен володіти такими якостями:

- Комунікбельність. Модератору часто доводиться спілкуватися з користувачами сервісу, писати їм листи, розмовляти по телефону;
- Відповідальність. Важливо пам'ятати, що від модератора залежить загальна картина роботи сервісу, завдяки йому у користувачів складається враження про інтернет-ресурсі, тому ступінь відповідальності за результати роботи у нього повинна бути високою;
- Уважність. Модератор не може пропустити жодної помилки в оголошенні, матеріалі або коментарі;
- Посидючість. Необхідно мати терпіння, щоб уважно проглянути всі матеріали, повідомлення та коментарі;

- Об'єктивність.

Модератор повинен вміти усуватися від свого особистого ставлення до теми обговорення, бути об'єктивним.

Маркетолог - маркетолог займається рекламуванням і просуванням web-сервісу. Кількість маркетологів - 2 людини. Потрібна вища або середньо-спеціальну освіту маркетолога.

Вимоги до продукту розробки.

Вимоги до продукту в цілому. При реалізації і використанні web-сервісу повинні бути враховані вимоги до функціональних характеристик, параметрів технічних засобів, інформаційної та програмної сумісності.

Вимоги до апаратного забезпечення. Мінімальні вимоги до програмних і апаратних засобів для нормального функціонування програми: Процесор:

- AMD або Intel з частотою 2,3 GHz і вище;
- ОЗУ: 2 Gb і вище;
- ОС: Windows 7 і вище;
- Монітор: SVGA монітор;
- Ємність ЖД: вільного місця не менше 10 Gb;
- Інші вимоги: мережева карта, клавіатура, маніпулятор миша. У кожного працівника повинно бути персональне цифровий пристрій.

Вказівка системного програмного забезпечення (операційні системи, браузері, програмні платформи і т.п.). Програмна система функціонує в середовищі Windows 7 і вище з будь-яким браузером.

Вказівка програмного забезпечення, використовуваного для реалізації. Denwer 3 PHP 5.3.13, MySQL 5.1, PostgreSQL 8.4 etc - локальний сервер; Laravel – фреймворк на мові PHP; Google Chrome - браузер; Photoshop CS6 - графічний редактор.

Для мережевих систем - особливості реалізації серверної і клієнтської частин.

Серверна частина: Web-сервіс може бути розташований на хостингу з певним доменом. Існують різні тарифи на хостинг з різними цінами. Для того щоб web-сервіс функціонував повноцінно потрібно вибрати для нього домен, вибрати тариф для хостингу, зареєструвати домен, оплатити замовлення, отримати доступ до вибраного домену. Після чого можна переносити web-сервіс на хостинг. Клієнтська частина: Web-браузер є клієнтською частиною, за допомогою браузера здійснюється доступ до web-сервісу.

Формати вхідних та вихідних даних

У форматі вхідних даних виступає:

- З боку замовника - формування заявки для замовлення конкретної послуги;
- З боку виконавця - формування заявки на виконання конкретної послуги;

У форматі вихідних даних виступає;

- З боку замовника - виконана послуга;
- З боку виконавця - отримання оплати за виконану послугу;
- Джерела даних та порядок їх введення в систему (програму), порядок виводу, зберігання.

Джерелами даних є користувачі (клієнти). Користувач реєструється в системі, після чого отримує доступ до сервісу. Всі заявки проходять через адміністратора web-сервісу. Всі дані, вся інформація зберігаються на сервері.

Порядок взаємодії з іншими системами, можливості обміну інформацією. Взаємодією з іншими системами є тільки реклама web-сервісу.

Заходи захисту інформації. Захист логіна і пароля. Для того щоб увійти в середу сервісу потрібно ввести пароль від облікового запису.

Вимоги до призначеного для користувача інтерфейсу.

Загальна характеристика призначеного для користувача інтерфейсу. В роботі web-сервісу виділені три основні завдання: перегляд переліку послуг та місць для відпочинку, перегляд карти, замовлення послуги для її виконання. Відповідно до цього в інтерфейсі передбачена панель з кнопками, за допомогою яких можна переміщатися з однієї сторінки на іншу

Особливості введення інформації користувачем, уявлення вихідних даних. Для замовлення послуги замовнику потрібно пройти реєстрацію і заповнити форму заявки:

- вибрати послугу зі списку послуг;
- позначити суму оплати за виконання послуги;
- вибрати конкретний період для виконання послуги;
- заповнити додаткову інформацію (адреса, доп. Інформацію про послугу, вимоги і т.п.), пройти просту перевірку на анти-бот (наприклад, 5 + 10 =?), Натиснути кнопку «Відправити».

Для виконання послуги виконавцю пройти реєстрацію, вибрати в формі зі списку конкретну послугу і натиснути кнопку «Відправити».

Процес розробки включає в себе наступні етапи:

- Вибір платформи/фреймворку для веб-сервіса;
- Завантаження та встановлення фреймворку LARAVEL;
- Налаштування міграцій та бази даних веб-сервісу;
- Створення та верстка дизайн/макет веб-сервіса;
- Підключення API від Google;
- Написання коду для виведення локацій відпочинку/розваг;

- Створення персонального аккаунт та реєстрації користувачів;
- Створення фільтру радіуса від місця положення користувача;
- Створення сторінки «улюблені».

Фунціонал веб-сервісу має надавати наступні можливості:

- Виводити рейтинг;
- Виводити Тип закладу(кафе, ресторан, парк, тощо...);
- Виводити відгуки;
- Виводити цінову політику;
- Виводити час роботи;
- Виводити адресу;
- Виводити місце положення на карті.

## РОЗДІЛ 2. ОПИС ПРОГРАМНИХ ЗАСОБІВ

### 2.1 Засоби розробки веб-сервісів

В епоху обчислювальної техніки, в якій панує Всесвітня павутина, технологія, яку називають веб-послугами, є ключовою для того, щоб комп'ютери могли зв'язувати машину з машиною, програмувати до програми. Подібно до того, як поява Мережі перетворило спосіб спілкування людей між собою та збору інформації, користувачі веб-сервісів отримують великі вигоди. Наприклад, веб-служби спрощують підключення всіх типів комп'ютерних програм між собою. Як ви побачите у цьому звіті, веб-служби створюють основу взаємодії, вкрай необхідну у світі, де комп'ютерні послуги та цифрова інформація існують у різних формах та варіантах.

Якщо в майбутньому бібліотеки хочуть бути ізольованими островами в океані змісту та інформації, вони можуть ігнорувати веб-служби. Але через те, що більша частина того, що роблять бібліотеки, зосереджується на наданні інформації бібліотечній клієнтері, а також тому, що інформація стає все більш електронною - що змушує бібліотеки збігатися з багатьма іншими організаціями в інформаційній сфері - бібліотекам необхідно співпрацювати та взаємодіяти з широким набором інших організацій та їх технічна інфраструктура.

Веб-сервіси забезпечують механізми, що дозволяють бібліотекам розширювати свої послуги багатьма важливими способами. Наприклад, вони дозволяють бібліотекам надавати послуги меценатам через небібліотечні інтерфейси; уможливити транзакції між підприємствами з постачальниками бібліотек; та підтримка кулуарного пошуку та пошуку у віддалених ресурсах для покращення сервісу. Цей звіт надасть багато прикладів поточних та потенційних можливостей, що стали можливими завдяки веб-службам.

Архітектура web-сервісів в даний час є однією з найбільш швидко розвиваються IT-сфер, оскільки вже стала фактично стандартом для взаємодії різних додатків, що функціонують на різних платформах. Ця архітектура підтримує розробку та впровадження відкритих систем, в яких розробка та інтеграція різних компонентів може виконуватися як на етапі розробки системи в цілому, так і під час її функціонування. Індивідуальні компоненти (web-сервіси) представляють свої послуги через реєстр (зазвичай розроблений з використанням UDDI-стандарту).

Послуги описуються XML-подібним мовою, який носить назву WSDL (Web Services Definition Language). Web-сервіси в даний час широко використовуються в розробці численних критичних додатків, серед яких слід особливо відзначити Інтернет-банкінг, онлайн-магазини, системи резервування і продажу туристичних послуг, системи електронного бізнесу та електронної науки. Саме тому дослідження по підвищенню надійності подібного роду систем є перспективним напрямком розвитку передових інформаційних технологій.

Створення Web-сервісу складається з наступних основних етапів:

- розробка проекту. На даному етапі визначаються призначення майбутнього проекту, які завдання він повинен вирішувати, і способи їх реалізації;
- створення дизайну і навігації. Дизайн відповідає за зовнішнє сприйняття даного web-сервісу, і перше на що звертає увагу користувач - це зовнішнє оформлення програми. Даний етап може включати в себе: створення ескізів сторінок, розробку логотипу, оформлення інтерфейсу користувача і їх розташування на сторінці і інше;
- верстка. Опис структури елементів проекту на мові розмітки відповідно до заданої специфікації;



- програмування. Створення «активного» додатку, що дозволяє не тільки виводити потрібну користувачу інформацію, а й отримувати будь-які дані від нього;
- встановлення. Копіювання проекту на робочий сервер і інсталяція. Налаштування конфігурації сервера для працездатності сервісів;
- тестування і налаштування. Тестування даного проекту на наявність помилок, для посвідчення справності роботи програми або усунення знайдених неполадок.

На даний момент сайти є вже практично у всіх досить великих компаній. А ті, у кого сайту немає, мріють його створити. І, останнім часом, більшість користувачів почали розуміти, що створення сайту - не така вже й легка справа. Існують певні технології створення сайтів, якими потрібно добре володіти, щоб створити хороший, працює сайт. Адже сайт є не тільки візитною карткою компанії, але і одним (а найчастіше основним) з рекламних, а також інформаційних інструментів. Тому постає питання, яким чином можна отримати хороший сайт, який виправдає вкладені в нього кошти.

## 2.2 PHP

Це скриптова мова програмування, створена для генерації HTML-сторінок на веб-сервері і роботи з базами даних. На даний момент він підтримується практично всіма представниками хостингу, входить в «стандартний» набір для створення сайтів (LAMP - Linux, Apache, MySQL, PHP).

Завдяки своїй простоті, швидкості виконання, багатій функціональності, розповсюдженню початкових кодів на основі ліцензії PHP, ця мова є чи не найпопулярнішим в області технологій створення сайтів. Відрізняється наявністю ядра і модулів, «розширень»: для роботи з базами даних, сокетамі,

динамічною графікою, криптографічними бібліотеками, документами формату PDF і т.п. Є можливість розробити, а також підключити додаткове розширення.

Можливості PHP дуже великі. Головним чином, PHP застосовується при написанні скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI (наприклад, обробляти даних форм, генерувати динамічні сторінки, відсилати і приймати cookies). Але PHP дає можливість виконувати також безліч інших завдань.

Існують три основні області, де використовується PHP:

- створення скриптів для виконання на стороні сервера;
- створення скриптів для виконання в командному рядку;
- створення додатків GUI, що виконуються на стороні клієнта.

Крім цього, PHP:

- доступний для більшості операційних систем, включаючи Linux, багато модифікації Unix (такі, як HP-UX, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS, і багатьох інших;
- включає підтримку більшої частини веб-серверів (для більшості серверів PHP поставляється в якості модуля, для інших, що підтримують стандарт CGI, PHP може функціонувати в якості процесора CGI);
- підтримує широке коло баз даних;
- підтримує DBX для роботи на абстрактному рівні (таким чином можна працювати з будь-якою базою даних, що використовують DBX); ODBC (тобто ви можете працювати з будь-якою базою даних, що підтримує цей стандарт);

- підтримує "спілкування" з іншими сервісами з використанням різних протоколів: LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows), а також багатьох інших;
- підтримує стандарт обміну складними структурами даних WDDX;
- підтримує об'єкти Java, дає можливість використовувати їх в якості об'єктів PHP;
- дає можливість формувати зображення, файли PDF, ролики Flash, створювані "на льоту"; здатний видавати будь-які текстові дані (XHTML, інші XML-файли); автоматично генерувати і зберігати в файлової системі вашого сервера;
- включає засоби обробки текстової інформації, починаючи з регулярних виразів Perl або POSIX Extended і закінчуючи парсером документів XML;
- підтримує багато інших розширення (функції пошукової машини mnoGoSearch, функції IRC Gateway, функції для роботи зі стиснутими файлами (gzip, bz2), функції календарних обчислень, функції перекладу та ін.).

### **2.3 JavaScript**

Це поки ще відносно молодий мову програмування, але вже дуже популярний в області технологій створення сайтів. На даний момент, робота над ним ще не закінчена. Він постійно допрацьовується і удосконалюється. Технічний комітет працює над істотними розширеннями, включаючи механізми для сценаріїв, які будуть створені для застосування в Internet, а також більш жорсткою координацією з іншими основними стандартами груп World Wide Web Консорціум і Wireless Application Protocol Форум. Java Script вже зіграв дуже важливу роль у розвитку технологій створення сайтів

В даний час використовується третє видання ECMA-262, що включає потужні регулярні вирази, кращу обробку рядків, нові інструкції контролю, управління, перехоплення і обробку виняткових ситуацій, більш жорстке визначення помилок, форматування для числового виведення і незначні зміни в очікуванні введення засобів багатомовності та майбутнього розвитку мови.

## 2.4 Мова розмітки HTML

Ця мова є базовим в області технологій створення сайтів, так як відносно легкий в освоєнні. Але надмірна простота є і його недоліком. HTML (від англійського Hyper Text Markup Language - мова розмітки гіпертексту) прекрасно відповідав вимогам раннього періоду розвитку технологій створення сайтів, але з подальшим його розвитком виникли істотні проблеми. HTML надає наступні можливості:

- видавати мережеві документи з заголовками, текстом, таблицями, списками, фотографіями і т.п. ;

- отримувати інформацію з Мережі через посилання гіпертексту при натисканні кнопки;

- створювати форми для посилки запитів на віддалені комп'ютери, щоб проводити пошук інформації, здійснювати бронювання, замовляти товари та т.п.;

- включати електронні таблиці, відео кліпи, аудіо кліпи, і інші програмні додатки безпосередньо в їх документи.

Історія розробки HTML досить тривала. У кожній його версії розробники намагалися домогтися того, щоб HTML-сторінки читалися усіма браузерами, на всіх комп'ютерних платформах.

Мова HTML призначений для формування та оформлення сторінок тексту, який призначений для перегляду в мережі Інтернет за допомогою спеціальних програм - браузерів (основні з них Netscape Navigator (фірми Netscape) і Internet Explorer (фірми Microsoft)).

В даному випадку, під сторінкою мається на увазі окремий файл зі спеціальними позначками, які вказують браузеру, як саме повинен бути відображений на екрані і в роздруківці текст, що міститься в цьому файлі. Крім цього, можуть ще бути присутнім посилання на файли з графічними зображеннями, які будуть відображатися разом з текстом при перегляді.

Для формування файлу HTML немає необхідності у використанні потужного текстового процесора, більш зручне використання, наприклад, редактора Notepad (Блокнот). Вміст HTML-сторінок представлено в гіпертекстовому вигляді, що передбачає наявність в документі, який відображається на екрані, виділених певним чином місць, клацнувши мишею по яким, можна переміститися в іншу частину цієї ж сторінки або на іншу сторінку.

Коли мова заходить про розробку веб-сервісів, тут багато гнучкості. Усі основні мови програмування та середовища розробки забезпечують певний спосіб створення веб-служб. Далі подано лише зразок деяких засобів розробки та середовищ, які зазвичай використовуються для створення веб-служб.

## **2.5 Інші програмні засоби**

Microsoft .NET. Корпорація Microsoft прийняла веб-послуги як ключову стратегічну технологію, і вона брала активну участь у розробці протоколів, стандартів та специфікацій, пов'язаних з веб-службами. Компанія називає свою стратегію веб-сервісу .NET і додає цю назву до багатьох своїх продуктів та технологій.

Microsoft Visual Studio .NET є інтегрованим середовищем розробки для створення програмних додатків і має вбудовану підтримку веб-служб. Існують версії цього середовища для декількох різних мов програмування, включаючи Visual Basic, Visual C ++, Visual C # та J # (реалізація Java).

Корпорація Майкрософт працює над випуском своєї основи для розробки наступного покоління Indigo, яка була спеціально розроблена для створення сервісно-орієнтованих бізнес-додатків.

Apache. Програмне забезпечення для веб-серверів Apache керує дуже великою частиною веб-серверів у світі. Apache - це реалізація з відкритим кодом, створена всесвітньою командою розробників, координованою Фондом програмного забезпечення Apache ([www.apache.org](http://www.apache.org)).

Axis - це реалізація SOAP для веб-серверів Apache. За наявності Axis розробка веб-служби стає простішим завданням, оскільки вона піклується про рівень обміну повідомленнями SOAP. Axis працює з веб-службами, запрограмованими на Java. Apache Tomcat, "контейнер сервлетів", забезпечує середовище програми, в якій виконуються програми Java. У більшості випадків комбінація веб-сервера Apache, Apache Axis і Tomcat буде працювати разом, щоб забезпечити середовище для веб-служб на основі SOAP на основі програмного забезпечення з відкритим кодом.

Apache Axis замінює попередню реалізацію Apache SOAP. Axis має набагато більше можливостей, працює краще та виправляє багато проблем, які були присутніми в SOAP Apache.

SOAP:Lite. Модуль SOAP:Lite Perl надає можливість створювати веб-служби за допомогою мови програмування Perl. Perl доступний майже на всіх основних комп'ютерних платформах і особливо популярний для написання веб-програм. Perl - це програма з відкритим кодом, і її можна використовувати безкоштовно. Perl широко використовується на варіаціях Unix та Linux. На арені Windows компанія ActiveState пропонує ActivePerl, що включає

двійковий дистрибутив для Windows. Perl із SOAP:Lite буде використано для деяких прикладів у цьому звіті.

## РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

### 3.1 Модельне подання об'єкта розробки

Виходячи з основних завдань системи, вона повинна надавати такі можливості:

При першому відвідуванні сервісу 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 товарів, можливість ознайомитися з асортиментом магазину.

Якщо користувач не захоче користуватися послугами сервісу, він може покинути його, не залишивши записів про себе в базі даних. Для цього незареєстрований користувач не повинен мати доступ до форми замовлення і гостьовій книзі.

Якщо користувач захоче скористатися послугами, йому необхідна реєстрація, щоб отримати доступ до всіх функцій сервісу.

Для перегляду відомостей про послуги і записи в гостьовій книзі необхідно кілька сторінок, на яких буде розташовуватися каталог товарів. Так як різні категорії користувачів цікавляться різними видами послуг, буде доцільно надати до перегляду як повний каталог, так і тематичні підкаталоги. Записи на дисках можна розділити на три основні категорії: мобільні пристрої, комп'ютери та периферія.

Незареєстрований користувач повинен мати можливість переглядати існуючі записи в гостьовій книзі. З їх допомогою можна ознайомитися з переважаючою тематикою сервісу, очікувані послуги і думки клієнтів про роботу сервісу.



Реєстрація може здійснюватися кількома способами. Перший - заповнення користувачем заявки, відправлення її на перевірку та затвердження адміністраторам сайту. Переваги такого методу в тому, що ймовірність появи в базі даних сайту не до кінця заповнених, порожніх або заповнених неправильно профілів користувачів зменшується. Очевидні недоліки методу в тому, що робота адміністратора або декількох адміністраторів в такому випадку збільшується в кілька разів і в разі помилки адміністратора вина за неправильно розглянуту заявку лягає на нього. До того ж на таку перевірку потрібно як правило не менше години, а користувач бажає отримати доступ до сайту відразу ж після реєстрації.

Другий спосіб - заповнення користувачем форми і безпосередня реєстрація його в базі даних. У цьому випадку поява потенційно невірних записів збільшується, але вина в цьому випадку лягає безпосередньо за користувача, а в обов'язки адміністратора з цього питання входить періодично очищати базу даних від таких профілів.

Для сервісу було обрао другий спосіб. Він знижує навантаження на адміністрацію сервісу, яка, зважаючи на постійне оновлення послуг і зміни асортименту і так не дуже низька. До того ж відразу після реєстрації користувач може обрати послугу або залишити побажання в гостьовій книзі.

Способів зв'язку клієнта з адміністрацією також два.

Перший - це безпосередній зв'язок за допомогою прихованого повідомлення або електронного листа. Такий спосіб, на мій погляд, прийнятний більше для особистого спілкування.

Другий - це доступні як користувачу, так і адміністратору коментарі. Сайт, орієнтований на користувача повинен містити окрему сторінку, де кожен відкрито може висловити свою думку. До того ж, можливий випадок, коли побажання одного клієнта може бути підтримано іншими.

Зміст мого сервісу має на увазі наявність гостьової книги. Це місце, де кожен зареєстрований користувач може залишити коментар, видимий всім, в тому числі і незареєстрованим користувачам.

В першу чергу, розглянемо дизайн створюваного сервісу. Дизайн включає в себе кольорове оформлення, наявність логотипу.

Цільовою аудиторією сайту будуть потенційні клієнти які зазвичай вже користувалися тими чи іншими послугами і знають приблизні характеристики бажаних послуг тому вирішили обрати їх його в мережі.

Дизайн сервісу грає важливу роль в створенні самого сервісу.

Оформлення має бути тематичним, що б відвідувач потрапивши на сервіс відразу зрозумів чим займається дана компанія. Бажано менше різної банерної реклами провідної на сторонні ресурси. Менше непотрібних відволікаючих матеріалів, через які гість може забути навіщо зайшов.

Проаналізуємо можливі колірні рішення сервісу.

Перше рішення можна назвати «діловим». Воно відрізняється холодними кольорами - темно-зелений, синій різних відтінків, а також коричневий. Як правило, тут можна зустріти чорний текст на білому тлі, сині посилання і суворе оформлення. Таке рішення зазвичай використовують для своїх сервісів великі компанії, що спеціалізуються на ІТ-технологіях, промислові, будівельні підприємства. Таке поєднання кольорів говорить потенційному клієнту або замовнику про надійність і серйозності підприємства.

Друге рішення характеризується яскравістю, яскравістю кольорів, поєднаннями рожевого, червоного, салатого, жовтого кольорів. Такий вибір роблять як правило творці розважальних сервісів, в тому числі деяких інтернет-магазинів. Їх метою є незвичайне оформлення і залучення людей, як

правило молодих, для спілкування, обговорення спільних тем різної спрямованості.

Третє рішення - «спокійне». Його характеризує переважання неясних, пастельних тонів. Його часто можна спостерігати у сервісах навчальної спрямованості, так як воно не відволікає учня від процесу засвоєння знань; також таке рішення можна зустріти на інформаційних сайтах, особливо створених для людей різного віку.

Для оформлення мого сервісу я вибрав рішення першої так як воно найбільше підходить для мого проекту складається з нейтральних елементів. Основний фон сторінок сайту я зробив сірим, текст і посилання пофарбовані в чорний. Для виділення частини тексту або проведення розділової смуги використовується світло-сірий колір. Таке поєднання не дратує людину, кольори не змащуються, текст і посилання добре видно на сірому тлі.

У верхній частині сторінки поміщається логотип - зображення, що додає сайту індивідуальність. Логотип виконаний в тих же тонах, що й сервіс.

Зображення складається з трьох основних елементів:

- Логотип у верхній частині сторінки;
- Меню посередині сторінки;
- Основний зміст сторінки.

Логотип знаходиться на кожній сторінці і не змінюється в залежності від місцезнаходження на сторінках.

Меню призначається для навігації по сайту, в ньому відображаються посилання на всі сторінки сервісу. Меню також статично і виглядає однаково для всіх сторінок.

Основний зміст сторінки змінюється в залежності від місцезнаходження користувача. Нижче представлені кілька сторінок які будуть на в проекті і коротка інформація про зміст.

Головна сторінка. Ця сторінка відображається при першому вході до сервісу, її оформлення і зміст надає користувачеві перше уявлення про магазин. У верхній частині сторінки поміщається інформація, що повідомляє користувачеві профіль, призначення і контингент сервісу. Далі я вважав за потрібне помістити список нових послуг з різних категорій. Зареєстрованим користувачам буде цікаво дізнатися про нові надходження, а незареєстрованим - скласти уявлення про рід послуг.

Сторінка пошуку. У будь-якому сервісі, тим більше що містить безліч однорідної інформації, організація пошуку необхідна. Це зручно для користувача, якому потрібна конкретна послуга. Інша необхідність полягає в тому, щоб впорядкувати список товарів за алфавітом, але це вимагає додаткових ресурсів сервера.

На сторінці знаходиться форма пошуку, що складається з одного поля введення і кнопки «знайти». Користувачеві пропонується ввести в форму назву послуги. Якщо послуга з такою назвою присутня в списку, про неї виводиться вся доступна інформація, якщо ж відсутня - текст «Ваш пошук не дав результатів»

Сторінка реєстрації. Ця сторінка призначена для реєстрації користувача. Вона містить форму реєстрації з полями, необхідними для відправлення людині поштової посилки. Відповідальність за неправильне введення даних лежить на користувачеві і його потрібно додатково попередити. Форма містить захист від повторюваних логінів. Це вкрай необхідно, так як нік в системі - основний ідентифікатор користувача і у різних користувачів не може бути однакових логінів. Якщо користувач при реєстрації намагається ввести вже наявний в системі нік, профіль не реєструється і нижче форми з'являється

повідомлення «Цей логін вже використовується в системі! Введіть інший логін!» Якщо реєстрація пройшла успішно, користувач перенаправляється на сторінку, яка повідомляє йому про успішну реєстрацію.

Зворотній зв'язок. Ця сторінка служить для зв'язку користувачів і адміністрації сервісу. Вона дозволяє залишати коментарі, відгуки, побажання користувачів. На сторінці містяться останні 5 відгуків, нижче знаходиться форма для відправки коментарів. Форма перевіряє, чи міститься даний нік в системі. Якщо дані вірні, сторінка оновлюється і коментар додається. Коментарі доступні всім користувачам сервісу.

### **3.2 Опис продукту та результатів розробки**

Згідно з технічним завданням на розробку web-сервісу диплома, необхідно розробити сайт для планування вільного часу. Він призначений для розміщення поточної інформації, автоматизації пошуку даних і виведення результатів. Автоматизація процесу пошуку даних дозволяє вирішити багато завдань, пов'язані з пошуком необхідної інформації. На цьому сайті ви можете розміщувати інформацію і здійснювати пошук інформації за різними критеріями.

Основні особливості програми:

- вибір необхідної інформації;
- натискаючи на гіперпосилання і кнопки;
- простий і зручний інтерфейс для роботи користувача.

Вибір методу рішення

В даний час існує безліч способів створення програмного забезпечення, схожого на веб-сайт.

Для вирішення проблеми можна надати такі програмні засоби як:

- Мова гіпертекстової розмітки (HTML);
- об'єктне середовище візуального програмування Visual Studio Code;
- Текстовий редактор блокнота;
- Мова JavaScript;
- PHP;
- Фреймворк Laravel.

За допомогою кожного з цих інструментів можна забезпечити повне виконання завдання на проектування. Сервіс, розроблений в рамках дипломного проекту, спрощує і прискорює процес пошуку і відображення на екрані необхідної інформації. Сервіс може бути використаний як самостійний продукт.

#### Розробка структури сервісу

Перше місце в процесі створення сервісу займає турбота про правильність шляхів до файлів. Якщо ви заздалегідь подумаєте про іменування файлів і каталогів на локальному комп'ютері, то пізніше проблем при розміщенні сайту на веб-сторінці буде менше.

Незважаючи на те, що файлова структура сайту може відрізнятися, ви завжди повинні слідувати деяким правилам. Вам слід розглянути можливість поновлення сайту в майбутньому. Якщо вам потрібно замінити файли, то ви і всі посилання потрібно буде змінити, а це може зайняти багато часу і сил.

Відомі такі способи організації веб-сервісів:

Всі файли в одному каталозі. Невеликі сайти з декількома HTML-сторінками і декількома зображеннями можуть дозволити собі зберігати всі їх вміст в одній директорії. Одним з найбільших переваг цієї системи є те, що вам не потрібно вказувати шляхи для локальних зв'язків. Недоліком є те, що

така система перевантажується при зростанні сайту: в цій одній директорії, як і в будь-якому дампі, нічого не знайти. Оновити такий сайт непросто.

Функціональні каталоги. Один із способів організації складних веб-сайтів наступний. У кожній з директорій перебувають файли, що відносяться до однієї і тієї ж функціональної частини системи. Наприклад, на верхньому рівні відображаються тільки main.html і пов'язані з ним графіки. Каталоги наступного рівня містять файли, які стосуються кожному розділу сайту: «опис лікарні», «меню» і т.д. Кожен каталог може містити власні HTML-файли та пов'язані з ними графічні або мультимедійні файли.

Директорії за типами файлів. Деякі веб-дизайнери вважають за краще зберігати файли одного і того ж типу в кожній директорії, незалежно від того, до якої функціональної частини сайту вони належать. В основному каталозі сайту зберігається тільки index.html, в інших каталогах можуть міститися, наприклад, всі зображення сайту, HTML сторінки, завантажені файли та ін. Основною перевагою такого підходу є те, що при оновленні файли потрібно замінювати тільки один раз. Наприклад, якщо одне і те ж зображення використовується на декількох сторінках, досить замінити його тільки в одному місці, де зберігаються зображення. В цьому випадку всі посилання на всіх сторінках будуть вказувати на оновлений файл (звичайно, не забудьте назвати його точно так само, як і старий).

Після того, як ви визначили процедуру організації файлів, ви повинні створити всі необхідні директорії як на сервері, так і на вашому комп'ютері. Тобто, якщо використовуються образи каталогів, то вони будуть такими ж, як і на сервері. Як і на локальному комп'ютері, повинен бути відносний шлях, який містить цей каталог. Створення файлової структури має 2 функції. По-перше, він захищає від випадкових помилок при розміщенні файлів не в тих каталогах на сервері. По-друге, це полегшує процес створення відносних URL (гіпертекстових посилань).

Говорячи про інтерфейс, необхідно сказати, що термін «інтерфейс» часто використовується в тих областях, де людина займається обробкою інформації на комп'ютері. Перекладено з англійської, інтерфейс означає зовнішнє обличчя. У комп'ютерному світі існує безліч різних типів інтерфейсів: Інтерфейс, графічний інтерфейс, інтерфейс вводу / виводу, зовнішній або внутрішній інтерфейс, інтелектуальний інтерфейс, людино-машинний інтерфейс, програмний інтерфейс і т.д.

Інтерфейс - набір засобів і правил, що дозволяють взаємодіяти між пристроями, програмами та людьми.

Особливе значення має інтерфейс, що дозволяє користувачеві взаємодіяти з персональним комп'ютером, званий інтерфейсом користувача. Зручність цього інтерфейсу в чому визначає успіх нового програмного продукту на конкурентному ринку програмних засобів. Інтерфейс може бути символічним або графічним.

Символічний інтерфейс зазвичай використовується, коли відеосистема працює в текстовому режимі. Інформація відображається на екрані монітора в символічному вигляді. До Windows всі операційні системи, включаючи MS DOS і його оболонку Norton Commander, надавали користувачеві символічний інтерфейс. Вона досить ресурсо-ефективність і може зробити роботу користувача досить комфортною. Винятком є інтерфейс командного рядка MS DOS, де користувач повинен знати синтаксис команди.

Графічний інтерфейс з'являється, коли відеосистема може працювати в графічному режимі, тобто на екрані монітора може відображатися інформація по пунктам. Перехід на графічний користувальницький інтерфейс став можливий завдяки поліпшенню технічних характеристик персонального комп'ютера. Такий інтерфейс висуває підвищені вимоги до швидкості роботи відеосистеми, але в той же час досягається головна мета - створення комфортного середовища для користувача, адже людина з зображеннями



(картинками) більш природний і зручний у використанні. Графічний інтерфейс сприймається як більш зрозумілий і інтуїтивно зрозумілий в порівнянні з символічним інтерфейсом.

Графічний інтерфейс користувача - це інтерфейс, в якому графічні засоби використовуються для взаємодії людини і комп'ютера.

Показовим прикладом графічного інтерфейсу користувача є інтерфейс Windows. При розробці даної операційної системи фахівці широко використовували можливі графічні інструменти: креслення, спеціальні іконки, кольорове оформлення, різноманітні шрифти, екранне оформлення і т.д.

Графічний інтерфейс Windows дозволяє швидше ставити команди операційної системи, запускати програми, вибирати файли, кнопки і пункти меню.

Колір і фон сторінки вибираються за бажанням веб-майстра, але при цьому слід враховувати і те, що від цього залежить візуальне сприйняття всього сайту. Немає причин приховувати все занадто багато, текст легко читається, коли він добре видно. Не використовуйте чорний колір і не мотивуйте його для продовження терміну служби моніторів користувача. Згідно зі статистикою, кількість відвідувачів «чорних» сайтів не дуже велике.

При виборі кольору фону важливо враховувати, як він гармонує з усіма іншими елементами: Колір тексту, колір гіперпосилань. Вкрай важливо забезпечити достатній контраст кольорів і яскравих елементів між елементами навігації на сторінці і фоном. Якщо зображення використовується в якості фону, воно повинно бути ледь помітно. Це зображення повинно містити тільки нейтральні, що не конфліктують між собою кольору.

Технологія каскадних таблиць стилів (Cascading Style Sheets, CSS) являє собою набір правил дизайну і верстки, які можуть бути застосовані до різних елементів веб-сторінки. У стандартній мові HTML для присвоєння елементу певних властивостей, таких як колір, розмір і т.д., ви повинні описувати його

кожен раз при його використанні. Використовуючи технологію CSS, буде досить один раз описати властивості елементів і визначити цей опис як стиль.

Завжди необхідно зберігати прийнятий стиль. Якщо сайт має розгалужену структуру, необхідно надати його сторінках такий же дизайн, тобто витриманий в єдиному стилі. Всі сторінки повинні мати однакові елементи управління, однакові інтервали і однакові шрифти.

За допомогою текстових фільтрів можна добитися хорошого дизайну сайту навіть без використання «крутих» зображень. Крім того, на відміну від зображень, ці фільтри не ускладнюють завантаження сторінок.

Далі була розроблена інструкція для використання веб-сервісу (Рисунок 3.1).

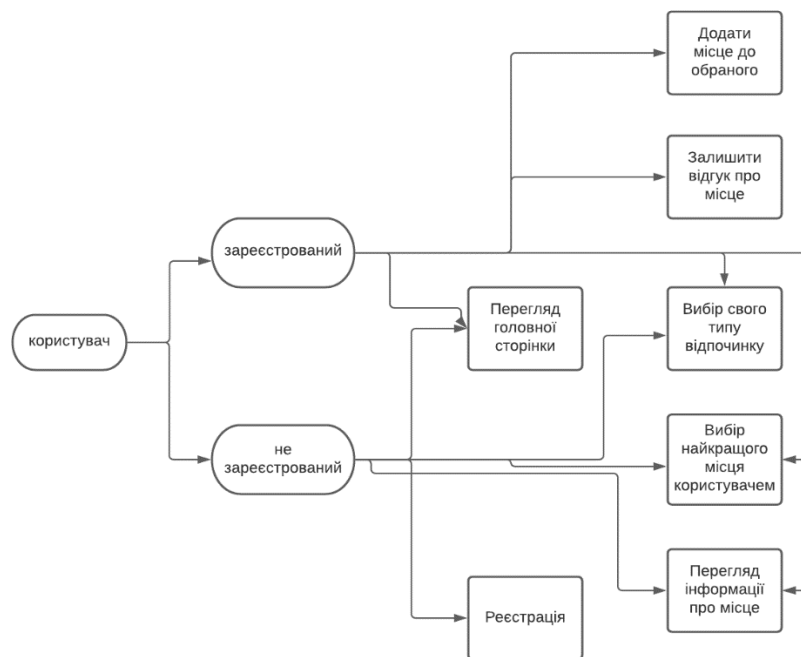


Рисунок 3.1 Схема використання веб-сервісу

При першому вході в web-сервіс необхідно зареєструватися. Потрібно заповнити форму реєстрації. Поля, відмічені зірочкою, є обов'язковими для заповнення (Рисунок 3.2).

Далі потрібно перевірити електронну пошту, яку вказано при реєстрації. На поштову скриньку прийде лист від адміністратора з підтвердженням «про реєстрацію облікового запису», в якому буде вказано діючий логін і пароль. Для входу потрібно ввести логін і пароль в форму входу в web-сервіс. З'явиться повідомлення з привітанням, тепер можна користуватися послугами сервісу.

Для того щоб редагувати інформацію в профілі треба зайти на вкладку «Мій профіль». Тут можна додати, виправити особисту інформацію, додати фотографію, змінити зовнішній вигляд профілю. Керувати списком друзів (Рисунок 3.3).

Для знайомства з послугами web-сервісу, необхідно перейти на вкладку головної сторінки, на цій сторінці надано перелік послуг та місць, які надає web-сервіс (Рисунок 3.4).

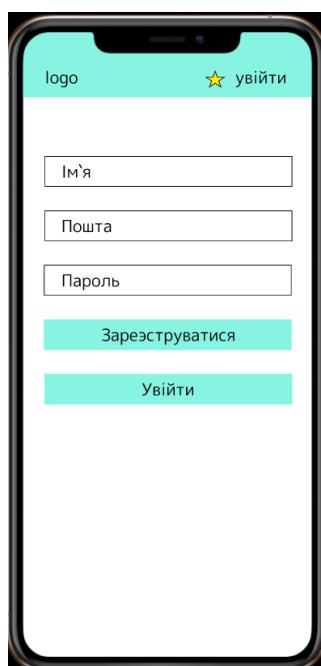


Рисунок 3.2 - Реєстрація в web-сервісі

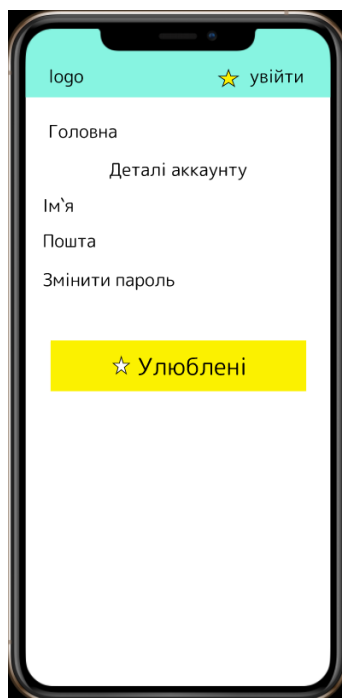


Рисунок 3.3 - Редагування профілю користувача

Для того щоб обрати місце та дізнатися детальну інформацію перейдіть на вкладку категорії зі списку (Рисунок 3.5).

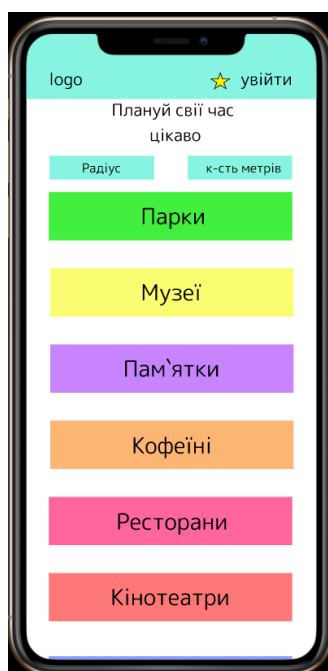


Рисунок 3.4 - Головна сторінка

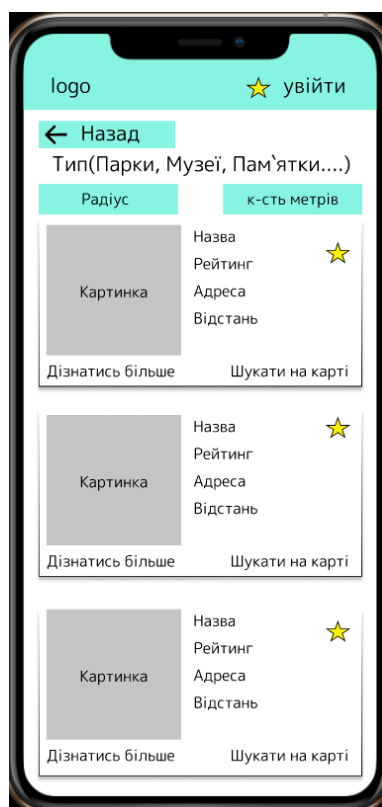


Рисунок 3.5 - Детальна інформація

Також користувач може переглянути інформацію певного місця натиснувши на його карточку (Рисунок 3.6).

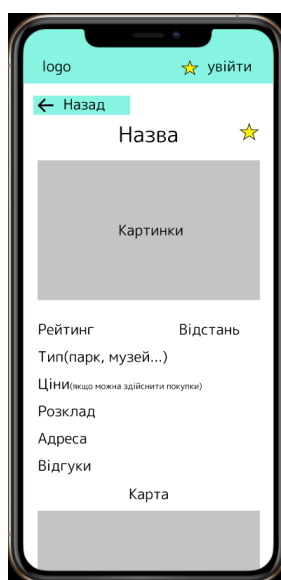


Рисунок 3.6 - Детальна інформація місця

Також користувач може ознайомитися с картою та визначити місцерозташування потрібної пам'ятки чи місця (Рисунок 3.7).

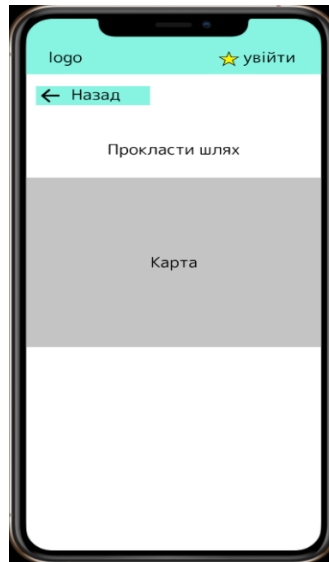


Рисунок 3.7 - Сторінка карти

Користувач може додавати та переглядати улюблені місця (Рисунок 3.8).

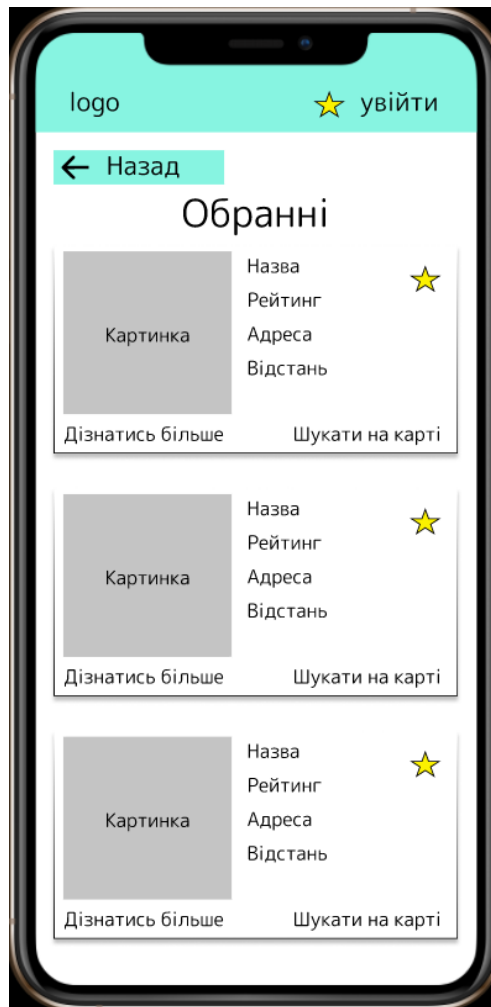


Рисунок 3.8 - Улюблені місця

## ВИСНОВОК

Використання послуг web-сервісу є однією з найбільш перспективних форм ведення бізнесу і організації роботи в сучасному суспільстві. Використання web-сервісу для надання різноманітних послуг вирішує проблему пошуку виконавців для виконання послуг.

Результатом даної роботи став готовий до роботи сервіс. В ході проведення роботи було виділено основні завдання сервісу, важливі особливості цільової аудиторії; проаналізовано різні шляхи вирішення і вибрано найбільш підходящі в даній ситуації. Оцінюючи виконану роботу можна зробити висновки:

- Сервіс надає користувачам з цільової групи найбільш потрібні їм послуги, а саме - можливість ознайомитися зі списком варіантів для планування вільного часу і висловити побажання адміністрації сервісу;
- Дизайн сервісу відповідає передбачуваним перевагам цільової групи, часу і цілям перебування потенційних клієнтів сервісу.
- Реалізовано сервіс пошуку по сайту, що я вважаю комфортним і навіть необхідним доповненням;
- Реалізована навігація за допомогою меню по сторінках сервісу. Посилання легко відрізнити від основного тексту, знаходяться в одному місці;
- Приділено особливу увагу безпеці користувачів, контролю над правильністю вхідних даних, захист від переповнення бази даних.

У даній роботі була поставлена наступна мета: розробити web-сервіс для надання послуг з планування вільного часу використовуючи PHP та Javascript.

У процесі досягнення поставленої мети були виконані наступні завдання:

- Вивчено основні поняття розробки web-сервісів;



- Вивчено обрана система для розробки web-сервісу для надання сервісу для надання послуг з планування вільного часу;
- Розроблено web-сервіс для надання сервісу для надання послуг з планування вільного часу;
- Розроблено інструкцію з використання web-сервісу.

Таким чином, слід вважати, що результати розробки відповідають всім вимогам технічного завдання, поставлена мета досягнута. Робота носить закінчений характер.

На мій погляд, при належному просуванні сервісу міг би мати успіх у певній категорії користувачів. Для цього необхідно перевести його на більш потужний веб-сервер і більш містку і оперативну базу даних. Також потрібно продумати рекламні акції, спосіб оповіщення потенційних клієнтів і інші комерційні аспекти.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дорофеева О. Удосконалення доступу до галузевого веб-ресурсу / Ольга Дорофеева // Бібліотечний вісник. – 2010. – № 1. – С. 3-9.
2. Інькова Н. А. Створення Web-сайтів : Навчально-методичний посібник (Електронний ресурс) / Інькова Н. А., Зайцева Е. А., Кузьміна Н. В., Толстих С. Г. – Режим доступу до електронного ресурсу: <http://club-edu.tambov.ru/methodic/fio/p5.doc> (дата звернення 30.05.2020).
3. Компанєєтс М.О. Принципи проектування ефективних вебсайтів / М.О. Компанєєтс // Молодий вчений. – 2015. – № 9. – Част. 2. – С. 106-108.
4. Крис Джамса. Эффективный самоучитель по креативному Web-дизайну / Крис Джамса, Конрад Кинг, Энди Андерсон. - ДиаСофтЮП, 2005 г. - 672с.
5. Мови та технології програмування - <https://www.metanit.com>
6. Розробка web-сервісу для надання побутових послуг - <https://core.ac.uk/download/pdf/156948657.pdf>
7. Серверный скриптовой язык Python [Електронний ресурс]. – Режим доступу: <http://bourabai.kz/alg/python.htm>
8. Основні етапи розробки веб-сайту (Електронний ресурс) / [укл. Юрчак І. Ю.] / Національний університет "Львівська політехніка". – Режим доступу до електронного ресурсу: <http://www.victoria.lviv.ua/html/wp/ssteps.html>
9. Паньчик Б. Розробка ефективних веб-додатків (текст на англ. мові) / Беата Паньчик // Актуальні проблеми економіки. – 2013. – № 11. – С. 443-251.
10. Тарнавська Т.В. Сутність інформаційних технологій в освіті / Т.В. Тарнавська // – К.: Кондор. – 2013. – 210с
11. Эллиот Э. JavaScript [Електронний ресурс] сайт: <http://chimera.labs.oreilly.com/books/1234000000262/index.html>

12. Budi Kurniawan . Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions / Budi Kurniawan : New Riders Publishing , April 12 2002 – 976 p.
13. Onishchenko. S. (2016). WEB-technologies. Berdyansk: BSPU. 500 (in Ukrainian). 2. Higher school pedagogy (2007).
14. Kurland, Z., Khmelyuk. R., Semenova. A. and others; 3rd ed., revised. and ext. Kyiv: Znannia. P. 495.
15. Charland, A. & LeRoux, B. Mobile application development: Web vs. native. Communications of the ACM. 2011. Vol. 54(5), P. 49-53.
16. PHP Fast & Easy Web Development, 2nd Edition. ISBN:193184187x; Premier Press © 2002
17. HTML - <https://www.w3.org/MarkUp/html-spec/html-essay.html>
18. What is JavaScript? - <https://www.infoworld.com/article/3441178>
19. JavaScript <https://www.123helpme.com/topics/javascript>
20. Essay On HTML - <https://www.bartleby.com/essay/Essay-On-HTML>



## ДОДАТКИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### «СТВОРЕННЯ ВЕБ-СЕРВІСУ ДЛЯ ПЛАНУВАННЯ ВІЛЬНОГО ЧАСУ З ВИКОРИСТАННЯМ PHP/JS»

Виконав студент 4 курсу  
 групи ПД-41  
**Немчин Станіслав Вячеславович**  
 Керівник  
**Дібрівний Олександр Андрійович**

Київ 2021

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- *Мета дослідження* - розробка веб-сервісу для планування вільного часу з використанням PHP/JS, призначеного для вибору потенційними клієнтами з спектру послуг з вибору місць, де користувач зможе провести вільний час.
- *Об'єктом дослідження* є підвищення ефективності планування вільного часу з використанням PHP/JS.
- *Предметом дослідження* є технологія розробки веб-сервісу для планування вільного часу.

## Актуальність роботи

В умовах сучасного інформаційного суспільства все актуальнішою стає потреба у новітніх технологіях, а саме використання сайтів, наприклад для покращення продажу різноманітних товарів, реклами фірм, навчання, перегляду фільмів, проведення вільного часу, перегляд новин чи прогнозу погоди. У період загальної комп'ютеризації встановлення зв'язку між бізнеспартнерами чи іншими особами стало набагато простіше та зручніше.



3

Сучасна епоха характеризується як епоха глобального інформаційного суспільства, зміст якої становить експоненціальне зростання інформаційних технологій. Одним з головних проявів є виникнення глобальної мережі Інтернет, стрімке і неухильне розширення її використання у всіх сферах життя суспільства. У міру розвитку Інтернету комунікативні можливості зростають, і все більш значна частина комунікації як ділового, так і особистого характеру здійснюється у віртуальному середовищі.



4

## Аналоги

Переваги	Недоліки
Мінімалістичний дизайн	Непрацююча реєстрація(на час перевірки)
Зрозумілий інтерфейс	Немає відгуків
	Немає рейтингу закладів
	Немає карти місця розташування закладів
	Немає можливості додати у обранні

**I-KIEV.COM**  
ПУТЕВОДИТЕЛЬ И ГИД ПО КИЕВУ

5

## Технічне завдання

Згідно з технічним завданням на розробку веб-сервіса диплома, необхідно розробити сервіс для планування вільного часу з використанням JavaScript та PHP. Він призначений для розміщення поточної інформації, автоматизації пошуку даних і виведення результатів. Автоматизація процесу пошуку даних дозволяє вирішити багато завдань, пов'язаних з пошуком необхідної інформації. На цьому сайті ви можете розміщувати інформацію і здійснювати пошук інформації за різними критеріями.



6

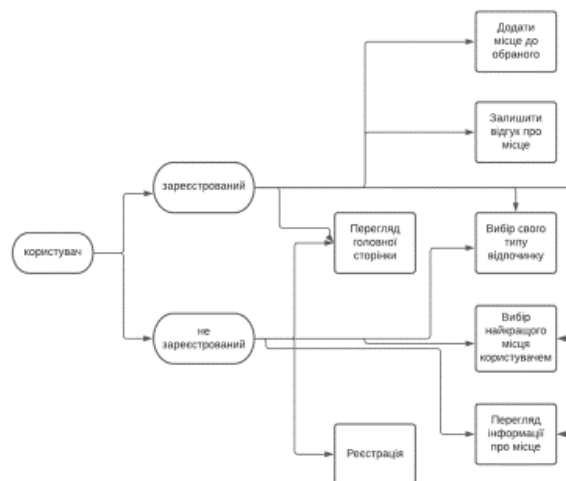
Основні особливості програми:

- вибір необхідної інформації;
- наявність гіперпосилань і кнопок;
- простий і зручний інтерфейс для роботи користувача.



7

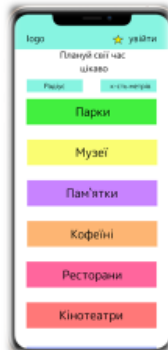
## Схема використання веб-сервісу



8



## Дизайн веб-сервісу представлений наступним чином



Головна сторінка

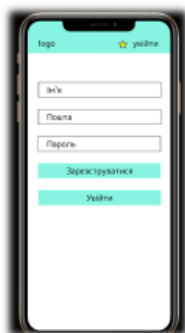


Перегляд карти/маршруту

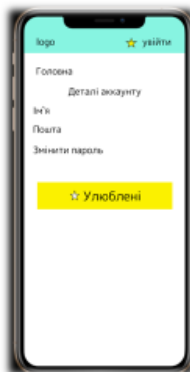


Сторінка з місцями для відпочинку

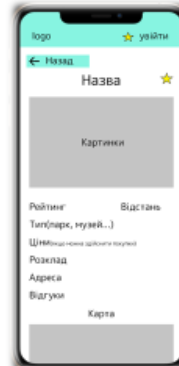
9



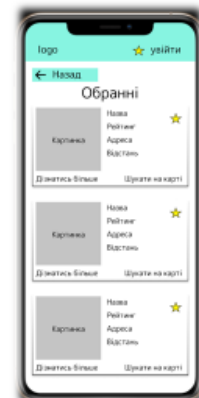
Сторінка реєстрації



Сторінка акаунту



Картка місця для відпочинку



Сторінка улюбленого

10

## ПРОГРАМНІ ЗАСОБИ ТА ІНСТРУМЕНТИ ДЛЯ РЕАЛІЗАЦІЇ

### Вибір методу рішення

В даний час існує безліч способів створення програмного забезпечення, схожого на веб-сайт.

Для вирішення проблеми можна надати такі програмні засоби як:

Мова гіпертекстової розмітки (HTML);

об'єктне середовище візуального програмування Visual Studio Code;

Текстовий редактор блокнота;

Мова JavaScript;

PHP.

За допомогою кожного з цих інструментів можна забезпечити повне виконання завдання на проектування. Сервіс, розроблений в рамках дипломного проекту, спрощує і прискорює процес пошуку і відображення на екрані необхідної інформації. Сервіс може бути використаний як самостійний продукт.

11

## Апробація результатів дослідження

Немчин С.В. науковий семінар факультету інформаційних технологій навчально-наукового інституту телекомунікацій "Перспективи інформаційних технологій найближчого майбутнього: Інтернет речей, роботизація, блокчейн", 11.04.2018р, ДУТ, м.Київ

12

## ВИСНОВКИ

Результатом даної роботи став готовий до роботи сервіс. В ході проведення роботи було виділено основні завдання сервісу, важливі особливості цільової аудиторії; проаналізовано різні шляхи вирішення і вибрано найбільш підходящі в даній ситуації. На мій погляд, при належному просуванні сервіс міг би мати успіх у певній категорії користувачів. Для цього необхідно перевести його на більш потужний веб-сервер і більш містку та оперативну базу даних. Також потрібно продумати рекламні акції, спосіб оповіщення потенційних клієнтів і інші комерційні аспекти.



Оцінюючи виконану роботу можна зробити висновки:

- Сервіс надає користувачам з цільової групи найбільш потрібні їм послуги, а саме - можливість ознайомитися зі списком варіантів для планування вільного часу і висловити побажання адміністрації сервісу.
- Дизайн сервісу відповідає передбачуваним перевагам цільової групи, часу і цілям перебування потенційних клієнтів сервісу.
  - Реалізовано сервіс пошуку місць для відпочинку;
- Реалізована навігація за допомогою меню по сторінках сервісу. Посилання легко відрізнити від основного тексту, знаходяться в одному місці.
- Приділено особливу увагу безпеці користувачів, контролю над правильністю вхідних даних, захист від переповнення бази даних.

**Дякую за увагу!**





