

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ВЕБСАЙТУ МОДЕЛЬНОГО АГЕНТСТВА
DEVELOPMENT OF A WEBSITE FOR A MODELLING AGENCY

Виконав(ла): студент(ка) 4 курсу, групи КНТ-110
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ПІКАЛОВ В.А.

(ПРІЗВИЩЕ та ініціали)

Керівник ГОЛУБ Т.В.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ПІКАЛОВА Владислава Анатолійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка вебсайту модельного агентства.
Development of a Website for a Modelling Agency.

керівник проєкту (роботи) к.т.н., доцент, ГОЛУБ Тетяна Василівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 24 ” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування вебсайту модельного агентства. 3. Розробка вебсайту модельного агентства. 4. Оптимізація вебсистеми та рекомендації по використанню.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ГОЛУБ Т.В., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 15 ” квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)


(підпис)

Владислав ПІКАЛОВ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)


(підпис)

Тетяна ГОЛУБ
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 74 с., 6 табл., 38 рис., 2 дод., 13 джерел.

ВЕБСАЙТ, МОДЕЛЬНЕ АГЕНТСТВО, HTML, CSS, JAVASCRIPT, PHP, MYSQL, ПОРТФОЛІО, ОПТИМІЗАЦІЯ.

Об'єкт дослідження – вебсайт модельного агентства та його компоненти.

Предмет дослідження – процес розробки вебсайту для модельного агентства з використанням сучасних вебтехнологій.

Мета роботи – розробка сучасного та функціонального вебсайту для модельного агентства, який дозволить підвищити ефективність діяльності компанії, розширити її присутність в онлайн-просторі та покращити взаємодію з цільовою аудиторією.

Матеріали, методи та технічні засоби: мови програмування HTML, CSS, JavaScript, PHP, система управління базами даних MySQL, фреймворк Laravel, методи оптимізації вебсайтів під пошукові системи, персональний комп'ютер з процесором Intel Core i5 під управлінням операційної системи Microsoft Windows 10.

Результати. Розроблено повнофункціональний вебсайт для модельного агентства з адаптивним дизайном, зручним інтерфейсом користувача та оптимізацією під пошукові системи.

Висновки. У результаті виконання кваліфікаційної роботи було створено сучасний та функціональний вебсайт для модельного агентства, який відповідає актуальним вимогам та трендам у сфері веброзробки.

Галузь використання – модельний бізнес, веброзробка, Інтернет-маркетинг.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 74 p., 6 tables, 38 figures, 2 appendix, 13 sources.

WEBSITE, MODELING AGENCY, HTML, CSS, JAVASCRIPT, PHP, MYSQL, PORTFOLIO, OPTIMIZATION.

The object of the research is a modeling agency website and its components.

The subject of the research is the process of developing a website for a modeling agency using modern web technologies.

The purpose of the work is to develop a modern and functional website for a modeling agency that will improve the company's efficiency, expand its online presence, and enhance interaction with the target audience.

Materials, methods, and technical tools: HTML, CSS, JavaScript, PHP programming languages, MySQL database management system, Laravel framework, website optimization methods for search engines, a personal computer with an Intel Core i5 processor running Microsoft Windows 10 operating system.

Results. A fully functional website for a modeling agency has been developed with an adaptive design, user-friendly interface, and search engine optimization.

Conclusions. As a result of the qualification work, a modern and functional website for a modeling agency was created, which meets the current requirements and trends in web development.

Field of application: modeling business, web development, Internet marketing.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Проектування сучасних вебсистем	10
1.2 Актуальні вимоги до вебсайтів в Україні.....	14
1.3 Порівняння сучасних вебсайтів модельних агентств.....	19
1.4 Висновок	24
2 Проектування вебсайту модельного агентства	26
2.1 Розробка структури вебсайту.....	26
2.2 Розробка бази даних.....	27
2.3 Розробка дизайну вебсайту	29
2.4 Вибір методів та мов розробки	32
2.5 Висновок	36
3 Розробка вебсайту модельного агентства.....	37
3.1 Програмна частина.....	37
3.2 Інтерфейс користувача	42
3.3 Висновок	49
4 Оптимізація вебсистеми та рекомендації по використанню	51
4.1 Семантичне ядро сайту.....	51
4.2 Карта сайту	52
4.3 Висновок	57
Висновки	58
Перелік джерел посилань	60
Додаток А Текст програми.....	62
Додаток Б Слайди презентації	68

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AJAX	- Asynchronous JavaScript and XML (;
API	- Application Programming Interface;
CSS	- Cascading Style Sheets;
DOM	- Document Object Model;
HTML	- HyperText Markup Language;
HTTP	- Hypertext Transfer Protocol;
IDE	- Integrated Development Environment
JSON	- JavaScript Object Notation;
MVC	- Model-View-Controller;
MySQL	- My Structured Query Language;
PHP	- PHP: Hypertext Preprocessor;
SEO	- Search Engine Optimization;
SQL	- Structured Query Language;
UI	- User Interface;
UML	- Unified Modeling Language;
URL	- Uniform Resource Locator;
IC	- Інформаційна система;
ООП	- Об'єктно-орієнтоване програмування;
ОС	- Операційна система;
ПЗ	- Програмне забезпечення;
СКВ	- Система керування вмістом;
СУБД	- Система управління базами даних.

ВСТУП

В умовах стрімкого розвитку інформаційних технологій та глобалізації бізнесу, все більше компаній переходять в онлайн-простір для розширення своєї присутності на ринку та залучення нових клієнтів. Не є винятком і індустрія моди та краси, зокрема модельний бізнес. Створення вебсайту для модельного агентства є важливим кроком для підвищення впізнаваності бренду, залучення нових моделей та клієнтів, а також для автоматизації бізнес-процесів компанії.

Сьогодні більшість успішних модельних агентств мають власні вебсайти, які служать потужним інструментом для просування своїх послуг, демонстрації портфоліо моделей та взаємодії з клієнтами. Сучасні вебтехнології дозволяють створювати привабливі та функціональні сайти, які відповідають високим стандартам індустрії моди.

Розробка вебсайтів для модельних агентств здійснюється з використанням різноманітних технологій, таких як HTML, CSS, JavaScript, PHP тощо. Для управління контентом часто використовуються системи керування вмістом (CMS), такі як WordPress або Drupal. Також популярними є фреймворки для веброботки, як от Laravel або Ruby on Rails.

Перспективними напрямками в цій сфері є використання технологій віртуальної та доповненої реальності для створення інтерактивних 3D-турів по портфоліо моделей, чат-ботів на основі штучного інтелекту для комунікації з відвідувачами сайту, а також інтеграція з соціальними мережами та месенджерами.

Метою цієї роботи є розробка сучасного та функціонального вебсайту для модельного агентства, який дозволить підвищити ефективність діяльності компанії, розширити її присутність в онлайн-просторі та покращити взаємодію з цільовою аудиторією.

Предметом дослідження є процес розробки вебсайту для модельного агентства з використанням сучасних вебтехнологій. Об'єктом дослідження є безпосередньо вебсайт модельного агентства та його компоненти.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- проаналізувати специфіку діяльності модельних агентств та їх потреби в онлайн-присутності;
- вивчити сучасні тенденції в вебдизайні та технологіях розробки сайтів;
- визначити функціональні та нефункціональні вимоги до вебсайту модельного агентства;
- розробити структуру та дизайн вебсайту відповідно до вимог;
- реалізувати вебсайт з використанням обраного стеку технологій;
- протестувати вебсайт та забезпечити його розгортання на хостингу.

Виконання цих задач дозволить створити повноцінний та ефективний інструмент для онлайн-присутності модельного агентства, що сприятиме зростанню бізнесу компанії та покращенню її іміджу в індустрії моди.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проєктування сучасних вебсистем

У сучасну інформаційну епоху люди все частіше звертаються до програмного забезпечення та послуг інженерів для вирішення різноманітних завдань. Однак, незважаючи на швидкі та потужні комп'ютери і загалом надійні програми, ми стикаємося з проблемою незадоволених, роздратованих і непродуктивних користувачів [1].

Раніше складні механічні пристрої вимагали ґрунтовної підготовки операторів, а доступ до великих машин мали лише професіонали у спеціальному одязі. Інформаційна ера змінила це, і тепер очікується, що пересічні користувачі зможуть впоратися з набагато складнішими технологіями, ніж будь-коли раніше. Яскравим прикладом є Інтернет, де від відвідувача сайту очікується вміння користуватися ним без попереднього навчання.

Сучасні інженери такі ж талановиті, як і завжди, тому, ймовірно, вони зіткнулися з якісно новою проблемою, відмінною від проблем індустріальної епохи. Інакше їхні традиційні інструменти працювали б так само добре. Цю нову проблему назвали "когнітивним опором" - це труднощі, з якими стикається людський інтелект при спробі розібратися у складній системі правил, що динамічно змінюються. Взаємодія з програмами має високий рівень когнітивного опору, тоді як взаємодія з фізичними пристроями, навіть складними, зазвичай викликає менший опір, оскільки механічні пристрої мають обмежену кількість станів порівняно з кількістю зовнішніх дій [1].

Комп'ютерні інтерфейси переповнені когнітивним опором. Навіть такі прості інтерфейси, як вебсторінки, змушують користувачів напружувати мозок більше, ніж будь-який механізм. Адже кожне гіперпосилання веде до іншого місця в Мережі, і ви можете лише клацнути на нього, а кінцева точка може змінюватися незалежно від вказівника без попередження. Когнітивний опір існує саме через гіпервластивості [2].

Більшість людей реагують на когнітивний опір однаково - вони беруть лише необхідний мінімум, а все інше ігнорують. Кожен користувач освоює найменший набір функцій, достатній для вирішення робочих завдань, відмовляючись від решти. Наприклад, в MS Word пересічний користувач не використовує 95% функцій.

Якщо програма дратує користувачів і є складною у використанні, люди почнуть уникати роботи з нею. Це може здаватися не надто значущим фактом, але варто усвідомити, що робота багатьох людей пов'язана із застосуванням програм. Корпоративні витрати на використання таких програм важко порахувати, але вони цілком відчутні і виражаються не стільки в грошах, скільки в інших, більш критичних валютах, таких як час, рівень безладу, репутація та лояльність клієнтів [1].

Справжня вигода від добре спроектованого продукту чи послуги - це висока лояльність, яку він викликає у ваших клієнтів.

Весь процес створення програмного забезпечення пронизаний проєктуванням - від вибору мови програмування до кольору вантажівки, що доставляє готовий продукт. Проєктування взаємодії здебільшого знаходиться в площині вибору поведінки, призначення, інформації та користувацького представлення цих аспектів.

У світі технологій завжди домінувала внутрішня архітектура програми, а проєктування зручної для користувача взаємодії завжди було другорядним завданням, яке вирішувалося у вільний час. Але зараз настав час зробити проєктування взаємодії головним аспектом при створенні програмних продуктів.

"Поведінкове проєктування" визначає, як елементи програми мають діяти і передавати інформацію. Наприклад, воно вказує, які інструменти можна застосовувати до таблиці звіту і як включати середні чи підсумкові показники. Проєктувальники взаємодії працюють від загального до конкретного, починаючи з цілей користувача, але враховуючи також ширші потреби бізнесу, технологічні обмеження та підпорядковані завдання [2].

Термін "проєктування взаємодії" є кращим за "проєктування інтерфейсу", оскільки слово "інтерфейс" передбачає, що код знаходиться в одному місці, люди - в іншому, а інтерфейс між ними дозволяє обмінюватися повідомленнями. Це означає, що саме інтерфейс відповідає за потреби користувачів. Наслідком ізоляції проєктування на рівні інтерфейсу є те, що програмісти роблять висновки на кшталт: "Я можу писати код, як мені заманеться, бо інтерфейс з'явиться після того, як я закінчу". Таким чином, проєктування відкладається до завершення програмування, коли вже надто пізно.

Дизайн інтерфейсу дозволяє надати певного вигляду вже існуючій поведінці системи. Наприклад, в генераторі звітів він прибирає непотрібні межі та інші візуальні перешкоди з таблиці з цифрами, виділяє кольором важливі поля, забезпечує якісний відгук на дії користувача тощо. Це краще, ніж нічого, але далеко не достатньо. Microsoft інвестує мільйони доларів у проєктування інтерфейсів, але продукти цієї компанії так і не здобули любов користувачів [3].

Процес проєктування сайту, як і будь-якої іншої програми, включає наступні кроки [4]:

- визначення мети сайту. Рамки проєкту задаються стратегією сайту. На стратегічному рівні визначається не лише те, що отримають власники сайту, а й те, що мають отримати його користувачі. Наприклад, для книжкового магазину стратегічні цілі очевидні: користувачі хочуть купити книги, а власники - їх продати. Але для інших сайтів цілі можуть бути не такими явними;

- створення словника термінів. Якщо розробляється великий сайт, для полегшення спілкування між розробниками та замовниками створюється словник термінів. У процесі проєктування, особливо під час мозкових штурмів, важливо мати детальний і точний словник. Технічні нюанси проєктування інтерактивних продуктів настільки важливі, що неправильно витлумачене слово може призвести до краху всього проєкту. Іноді різні

учасники команди замовника використовують такі поширені слова, як "кнопка" чи "діалог", для позначення зовсім різних речей. Буває, що десять високооплачуваних фахівців дві години запекло сперечаються через те, що користуються різними визначеннями одних і тих самих понять;

- визначення варіантів використання сайту (персонажі та дії). Це перелік та опис дій, які користувачі виконуватимуть на сайті. Для створення такого переліку потрібно мати модель користувача (персонажа);

- розробка сценаріїв. Це можливі послідовності дій для кожного варіанту використання, які часто називають бізнес-логікою або просто логікою сайту. Розрізняють сценарії для персонажа і сценарії для програмного забезпечення;

- створення інформаційної структури. Вона визначає, як різні функції та матеріали сайту співвідносяться між собою. Які саме функції та матеріали мають бути на сайті, визначається на попередніх етапах проектування. Наприклад, деякі книжкові інтернет-магазини мають функцію збереження адреси користувача для автоматичного заповнення при наступній покупці. Питання про включення цієї чи будь-якої іншої функції вирішується на етапі визначення варіантів використання сайту;

- розробка схеми сайту. Схема - це конкретне зображення більш абстрактної структури сайту. Вона визначає, наприклад, розташування елементів інтерфейсу на сторінці оплати, тоді як структура визначає, як користувачі потраплять на цю сторінку і куди зможуть перейти після здійснення оплати. Схема сайту, що лежить під його поверхнею, визначає розташування кнопки закладок, фотографій, текстових блоків тощо. Вона оптимізує взаємне розташування елементів сторінки для максимальної ефективності та привабливості, щоб користувач запам'ятав логотип сайту або швидко знайшов кнопку кошика, коли вона знадобиться;

- визначення зовнішнього вигляду сайту. На поверхні ви бачите набір вебсторінок, що складаються з графіки та тексту. На деяких зображеннях

можна клацнути і виконати певну дію, наприклад, переглянути вміст кошика. Інші зображення є лише ілюстраціями, як-от фото книги чи логотип сайту;

- підготовка текстового та іншого наповнення сайту. Будь-який сайт містить певну інформацію. На освітньому сайті зазвичай розміщують навчальні матеріали у вигляді текстів, таблиць, малюнків, презентацій, аудіо та відео файлів тощо. Найчастіше переважає текстова інформація, і розробники сайтів схильні вважати, що сприйняття навчального тексту з паперового підручника та з екрана монітора є однаковим. Але численні дослідження в галузі юзабіліті (науки про створення максимально зручних для користувача інтерфейсів програмного забезпечення) довели хибність цієї думки;

- кодування та програмування. Маючи проєкт сайту, програмісти та HTML-верстальники створюють програмний код і шаблони для зовнішнього вигляду вебсторінок.

1.2 Актуальні вимоги до вебсайтів в Україні

16 липня 2022 року вступають в силу зміни до Закону України «Про забезпечення функціонування української мови як державної», які регулюють застосування державної мови в Інтернеті та мови інтерфейсів комп'ютерних програм, встановлених на товарах. Відповідно, вебсайти мають бути приведені у відповідність до нових вимог.

На практиці на комерційних вебсайтах нерідко фіксуються випадки недотримання не лише вимог щодо державної мови, але й інших норм українського законодавства, пов'язані з неправомірним використанням різноманітного контенту. Це може призвести до накладання на власників таких вебсайтів значних штрафів та негативних репутаційних ризиків. Тому варто ретельно перевірити весь контент на вебсайті і, у разі виявлення порушень, наприклад, вимог конкурентного законодавства чи законодавства про рекламу, своєчасно їх усунути.

Вимоги до мови на вебсайтах [5].

Закон України «Про захист прав споживачів» передбачає право споживачів на необхідну, доступну, достовірну та своєчасну інформацію державною (українською) мовою про продукцію (товар, роботу чи послугу), її кількість, якість, асортимент, виробника (виконавця, продавця), якщо така інформація розміщується в Інтернеті на вебсайтах.

Починаючи з 16 липня 2022 року:

- будь-який вебсайт, власником якого є суб'єкт господарювання, зареєстрований в Україні, повинен мати україномовний інтерфейс та версію українською мовою;

- усі сторінки в соцмережах (на Youtube, Viber, Telegram тощо), які використовуються як інтернет-представництва українських суб'єктів господарювання, повинні мати основну україномовну версію, яка завантажується для користувачів в Україні за замовчуванням. Ці вимоги стосуються також інформаційних інтернет-ресурсів;

- україномовна версія інтернет-представництва іноземного суб'єкта господарювання, який продає продукцію в Україні, може містити менше інформації, ніж основна версія. Але обсяг такої інформації має бути достатнім для повного розуміння користувачем з України мети діяльності цього суб'єкта;

- мобільні додатки суб'єктів господарювання, що реалізують товари і послуги в Україні, повинні мати україномовну версію інтерфейсу користувача;

- суб'єкти господарювання, які обслуговують українських споживачів в Інтернеті (інтернет-магазини, інтернет-каталоги тощо), зобов'язані мати повноцінну україномовну версію, яка має бути основною, містити максимальний обсяг інформації і завантажуватися за замовчуванням.

Варто зазначити, що крім державної (української) мови, така інформація на вебсайтах, вебсторінках (Youtube, Viber, Telegram тощо) та мобільних додатках може дублюватися й іншими мовами.

Таким чином, відсутність україномовної версії комерційного вебсайту після 16 липня 2022 року може розцінюватися як порушення вимог Закону України «Про забезпечення функціонування української мови як державної» з боку його власника.

Вимоги до контенту на вебсайтах.

Окрім законодавчих вимог щодо використання української мови на вебсайтах, слід також згадати про інший контент, складові якого є об'єктами права інтелектуальної власності. Таким об'єктам необхідно приділити належну увагу і здійснити правовий аналіз їх комерційного використання.

Йдеться, зокрема, про такий контент:

- тексти, статті, оголошення рекламно-інформаційного характеру;
- інформацію про акції, конкурси, знижки;
- графічні зображення, дизайн;
- відеоролики, музичні банери;
- рекламні банери;
- зображення торгових марок чи слоганів інших правовласників;
- зображення фізичних та публічних осіб;
- персональні дані фізичних осіб тощо.

При перевірці контенту на вебсайті особливу увагу слід приділити інформації рекламно-інформаційного характеру, зокрема на предмет наявності в ній ознак недобросовісної конкуренції. Одним з поширених порушень українського законодавства про захист від недобросовісної конкуренції є поширення інформації, що вводить в оману споживачів, тобто відомостей, які містять неповні, неточні або неправдиві дані (про походження товару, виробника, продавця, спосіб виготовлення, ціну тощо).

Ретельної уваги та аналізу потребують розміщені на вебсайті графічні зображення, зображення фізичних осіб (відомих персон), оскільки згідно з вимогами українського законодавства для правомірного розміщення чужого зображення на комерційному вебсайті необхідно мати дозвіл від правовласника. А розміщення зображення фізичної особи потребує

відповідної письмової згоди від цієї особи. Важливо пам'ятати, що отримана згода фізичної особи може бути відкликана в будь-який момент (безпосередньо нею або її родичами) з вимогою подальшого відшкодування збитків, якщо розміщення зображення відбулось з порушеннями українського законодавства.

Непоодинокими є випадки добросовісного розміщення на вебсайтах, зокрема, роликів інформаційного характеру. Добросовісним може вважатися використання захищених авторським правом творів у новинах (розділ новин на вебсайті), а також з освітньою чи науково-дослідницькою метою. Використання таких роликів з комерційною (рекламною) метою не вважається добросовісним і потребує принаймні письмової згоди-дозволу правовласника цього чи подібного контенту.

Що стосується розміщених на вебсайтах зображень торгових марок, то таке розміщення має відповідати головному критерію – наявності відповідного дозволу на використання зображення торгової марки на вебсайті від її законного власника [4].

Зважаючи на зазначене, комерційний вебсайт та його контент повинні бути ретельно перевірені на відповідність вимогам Закону України «Про забезпечення функціонування української мови як державної», а також іншим нормам чинного законодавства України. Це дозволить вчасно виявити та усунути можливі порушення, інакше власнику комерційного вебсайту загрожує притягнення до відповідальності.

Відповідальність.

Розглядом скарг за порушення вимог законодавства щодо наявності української мови на вебсайті займатиметься Уповноважений Верховної Ради України із захисту державної мови. Цей регулятор відповідатиме за ініціювання розслідування порушень, складання протоколів і стягнення з порушників накладених штрафів.

З 16 липня 2022 року Уповноважений Верховної Ради України із захисту державної мови матиме право накладати штраф, зокрема, на

керівників та працівників підприємств, установ й організацій усіх форм власності, інших фізичних осіб, які порушуватимуть вимоги Закону. За перше порушення Уповноважений зможе накласти штраф у розмірі від 3 400 до 8 500 грн. За повторне порушення вимог Закону штраф становитиме від 8 500 до 11 900 грн.

Щодо неправомірного використання на вебсайті зображень чужих торгових марок, то Закон України «Про охорону прав на знаки для товарів і послуг» передбачає способи захисту порушених прав. Зокрема, передбачена кримінальна відповідальність за порушення авторського права і суміжних прав або незаконне використання торгової марки, комерційного (фірмового) найменування.

Законодавство України про захист від недобросовісної конкуренції передбачає значні санкції за відповідні порушення. Вчинення суб'єктами господарювання дій, визначених законодавством як недобросовісна конкуренція, тягне за собою накладення штрафу в розмірі до 5% доходу (виручки) від реалізації продукції (товарів, робіт, послуг) суб'єкта господарювання за останній звітний рік, що передував року накладення штрафу. Особи, яким завдано шкоди внаслідок вчинення дій, визначених як недобросовісна конкуренція, також можуть звернутися до суду з позовом про її відшкодування.

Органи АМКУ або особи, права яких порушено, також мають право звернутися до суду з позовом про вилучення з господарського обороту товарів з неправомірно використаним позначенням, копій виробів іншого суб'єкта господарювання.

Що стосується неправомірного використання на комерційному вебсайті зображень фізичних осіб без отримання відповідної згоди на таке використання, то власник комерційного вебсайту може отримати від правовласника такого зображення (автора, актора, зображеного на фото) майновий позов і зазнати значних штрафних санкцій через незаконне використання зображення з комерційною метою або для власної реклами.

За порушення рекламного законодавства України на власника комерційного вебсайту може бути накладено штраф у розмірі п'ятикратної вартості розповсюдженої на вебсайті інформації чи рекламного контенту.

1.3 Порівняння сучасних вебсайтів модельних агентств

Сучасні вебсайти модельних агентств відіграють важливу роль у просуванні послуг та пошуку нових талантів. Вони є віртуальною вітриною, яка демонструє потенційним клієнтам та моделям професіоналізм та можливості агентства. Сайт повинен мати привабливий дизайн, зручну навігацію та інформативний контент, щоб зацікавити відвідувачів та спонукати їх до співпраці.

Одним з ключових елементів сучасного вебсайту модельного агентства є портфоліо моделей. Воно має бути структурованим, з якісними фотографіями та детальною інформацією про кожну модель. Це дозволяє клієнтам швидко знайти потрібний типаж для своїх проєктів, а моделям - продемонструвати свої найкращі роботи.

Важливою функцією сайту є також можливість подачі заявки для потенційних моделей. Онлайн-форма з полями для введення основної інформації та завантаження фото спрощує процес відбору нових талантів. Крім того, сайт має містити чіткі інструкції та вимоги до кандидатів, щоб уникнути непорозумінь.

Для ефективної комунікації з клієнтами та моделями сайт повинен мати розділ контактів з адресою, телефоном, email та посиланнями на соціальні мережі агентства. Також корисно мати онлайн-чат або форму зворотнього зв'язку для швидкого вирішення питань відвідувачів.

Сучасні вебсайти модельних агентств часто включають блог або розділ новин, де публікуються актуальні події з життя агентства, успіхи моделей, поради для початківців тощо. Це підвищує довіру до бренду та залученість аудиторії.

Для зручності користувачів сайт має бути адаптивним, тобто коректно відображатися на різних пристроях - комп'ютерах, планшетах, смартфонах. Швидкість завантаження сторінок також є критичною, адже відвідувачі не будуть довго чекати на відкриття контенту.

З точки зору безпеки сайт модельного агентства повинен мати SSL-сертифікат для захисту даних користувачів та запобігання перехопленню інформації зловмисниками. Регулярні бекапи та оновлення системи керування контентом зменшують ризики втрати даних та злому.

Для покращення видимості сайту в пошукових системах необхідно проводити SEO-оптимізацію - використовувати ключові слова в текстах, мета-тегах, заголовках, альтернативних описах зображень. Також важливо отримувати якісні зворотні посилання з авторитетних ресурсів.

Інтеграція сайту з соціальними мережами дозволяє розширити охоплення аудиторії та залучити більше потенційних клієнтів та моделей. Розміщення віджетів соцмереж, кнопок "поділитися", автоматичний імпорт постів з Instagram та Facebook роблять сайт більш інтерактивним та динамічним.

Нарешті, сучасний вебсайт модельного агентства повинен регулярно оновлюватися та вдосконалюватися, щоб відповідати останнім трендам вебдизайну, юзабіліті та потребам користувачів. Тільки в такому випадку він буде ефективним інструментом для розвитку бізнесу в висококонкурентній сфері модельного бізнесу.

Модельне агентство "Ergo-Models" (рис. 1.1) [6].

Переваги:

- стильний та сучасний дизайн;
- зручний пошук моделей за параметрами;
- детальні портфоліо моделей з якісними фото;
- можливість онлайн-подачі заявки для моделей;
- адаптивність під різні пристрої.

Недоліки:

- відсутність блогу або розділу новин;
- не надто інформативний розділ "Про нас".

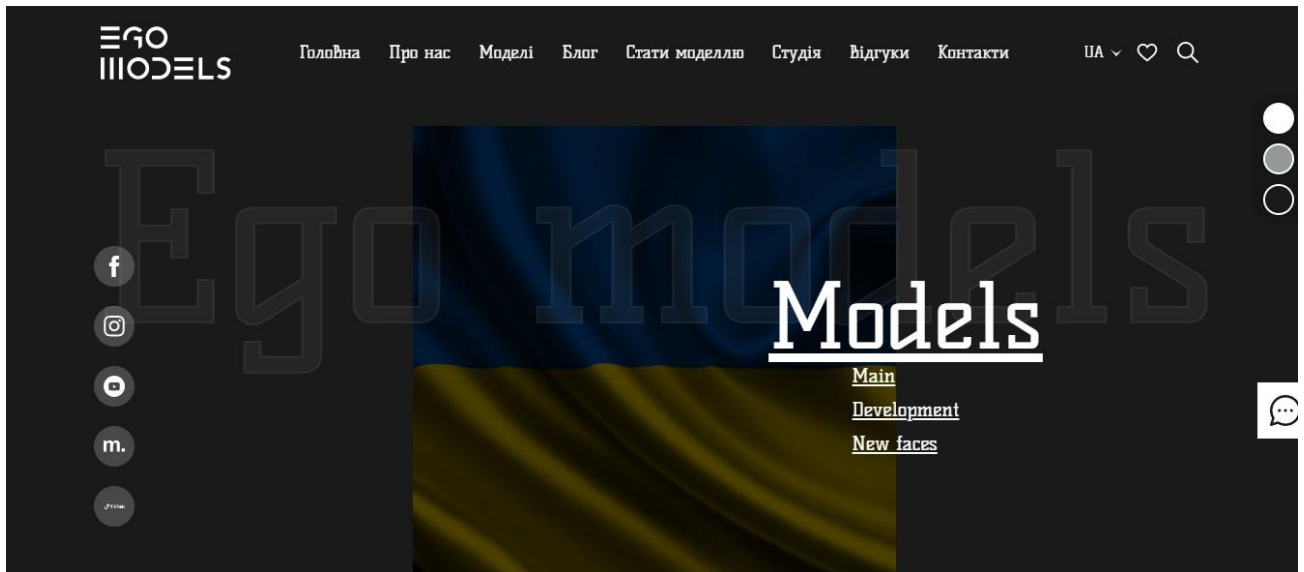


Рисунок 1.1 - Egro Models [6]

Модельне агентство "Браво" (рис. .1.2) [7].

Переваги:

- привабливий та яскравий дизайн;
- швидка загрузка сторінок;
- наявність відео-портфоліо моделей;
- розділ з детальною інформацією про послуги агентства;
- посилання на соцмережі.

Недоліки:

- відсутність онлайн-форми для подачі заявки моделями;
- не надто зручна навігація по сайту.

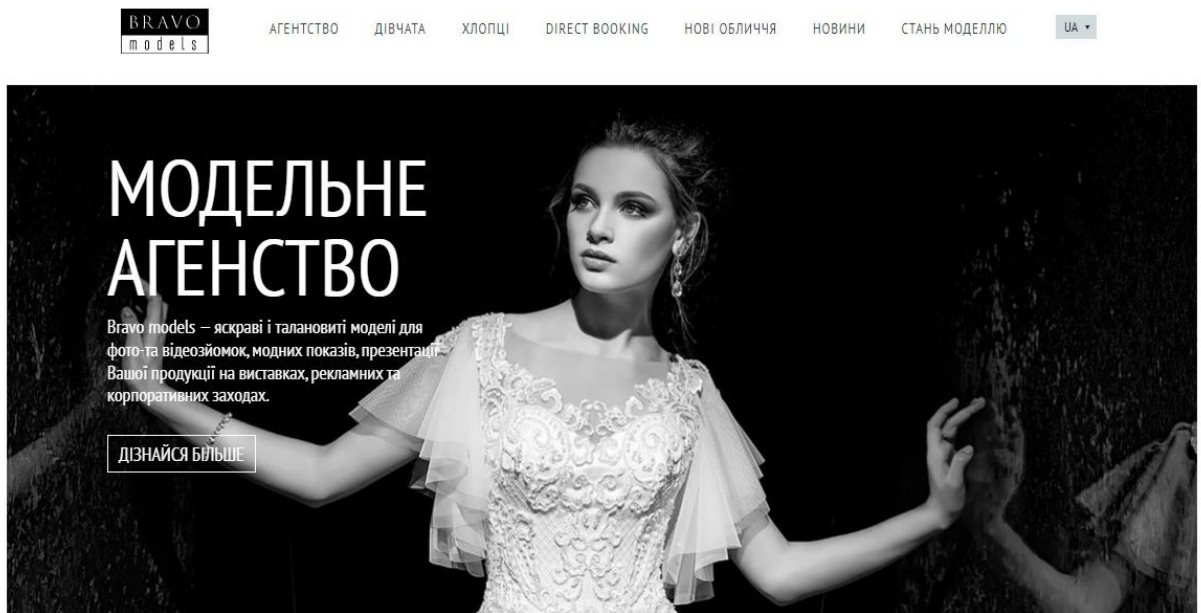


Рисунок 1.2 – Браво агенство [7]

Модельне агентство "Vea Models" (рис. 1.3) [8].

Переваги:

- лаконічний та стильний дизайн;
- зручний поділ моделей за категоріями;
- можливість онлайн-бронювання моделей;
- наявність блогу з корисними статтями;
- мультимовність (українська, англійська).

Недоліки:

- деякі сторінки довго завантажуються;
- не всі портфоліо моделей містять достатньо інформації.

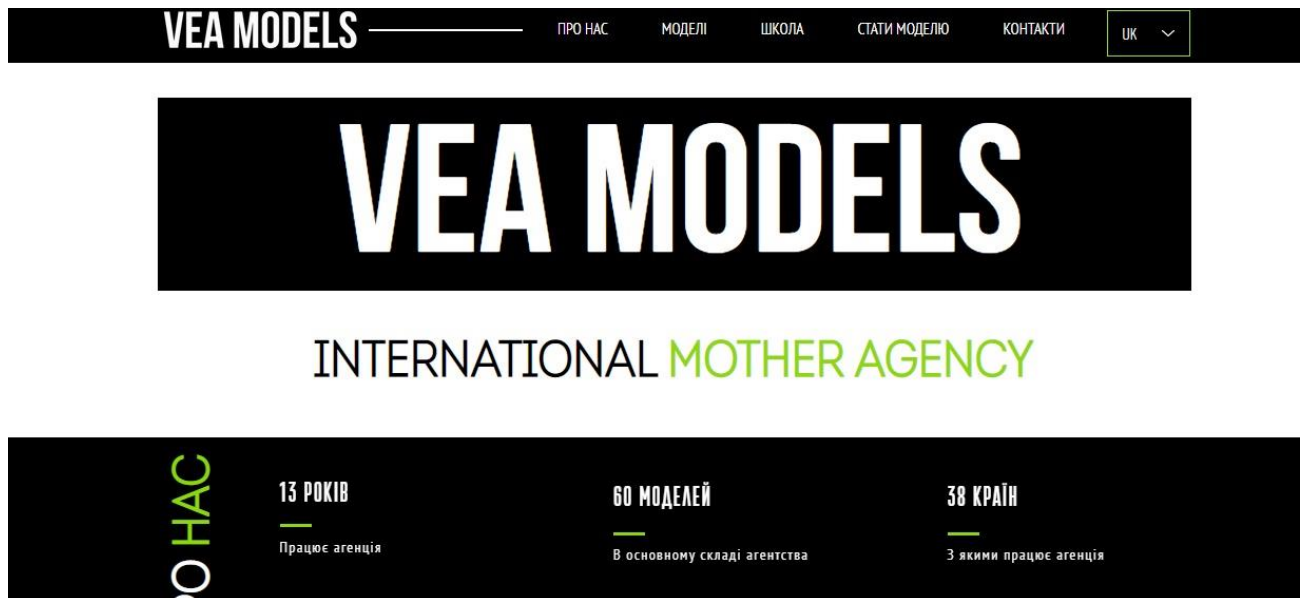


Рисунок 1.3 – Vea агентство [8]

При розробці нового вебсайту модельного агентства варто перейняти такі переваги:

- адаптивний та привабливий дизайн, який відповідає сучасним трендам та транслює імідж агентства;
- зручні інструменти для пошуку та фільтрації моделей за параметрами, категоріями тощо;
- детальні портфоліо моделей з якісними фото, відео та вичерпною інформацією;
- можливість онлайн-подачі заявки для нових моделей через форму на сайті;
- розділ з описом послуг агентства, цінами та умовами співпраці;
- блог або розділ новин для публікації корисного та цікавого контенту, який підвищує довіру та залученість аудиторії;
- інтеграцію з соціальними мережами для розширення охоплення та залучення нових клієнтів і моделей;
- швидку загрузку сторінок, зручну навігацію та інтуїтивно зрозумілу структуру сайту;
- мультимовність для охоплення міжнародної аудиторії;

- регулярне оновлення та вдосконалення сайту відповідно до потреб користувачів та змін на ринку.

Порівняння функцій обраних сучасних вебсайтів наведено в табл. 1.1.

Таблиця 1.1 – Порівняння вебсайтів

Критерії	Ergo	Bravo	VEA
Дизайн	Стильний та сучасний	Привабливий та яскравий	Лаконічний та стильний
Адаптивність	Адаптивний під різні пристрої	-	-
Швидкість завантаження	-	Швидка	Деякі сторінки довго завантажуються
Пошук моделей	Зручний пошук за параметрами	-	Зручний поділ за категоріями
Портфоліо моделей	Детальні з якісними фото	Наявність відео-портфоліо	Не всі містять достатньо інформації
Онлайн-подача заявки для моделей	Можливість онлайн-подачі	Відсутня	-
Інформація про послуги	-	Детальна інформація	-
Блог/новини	Відсутній	-	Наявний з корисними статтями
Розділ "Про нас"	Не надто інформативний	-	-
Соціальні мережі	-	Посилання на соцмережі	-
Додаткові можливості	-	-	Онлайн-бронювання моделей, мультимовність

1.4 Висновок

У першому розділі було розглянуто теоретичні аспекти проектування сучасних вебсистем, актуальні вимоги до вебсайтів в Україні та проведено порівняльний аналіз сучасних вебсайтів модельних агентств.

Проектування вебсайтів передбачає комплексний підхід, який враховує потреби користувачів, бізнес-цілі компанії та технологічні можливості. Ключовими етапами процесу проектування є визначення мети сайту, створення інформаційної архітектури, розробка сценаріїв взаємодії, створення прототипів та дизайну інтерфейсу.

Аналіз актуальних вимог до вебсайтів в Україні показав, що з 16 липня 2022 року всі комерційні сайти повинні мати україномовну версію інтерфейсу та контенту. Також важливо забезпечити дотримання вимог щодо використання об'єктів інтелектуальної власності, персональних даних та уникнення недобросовісної конкуренції. Недотримання цих норм може призвести до штрафних санкцій.

Порівняння сучасних вебсайтів модельних агентств дозволило виявити їх сильні та слабкі сторони, а також сформулювати рекомендації для розробки нового сайту. Серед ключових переваг успішних сайтів можна відзначити адаптивний та привабливий дизайн, зручні інструменти пошуку моделей, детальні портфоліо, онлайн-форми для подачі заявок, інтеграцію з соцмережами та наявність корисного контенту. Водночас, недоліками деяких сайтів є низька швидкість завантаження, неповнота інформації та недостатня зручність навігації.

Таким чином, проведений аналіз дозволяє сформулювати чітке розуміння принципів проектування сучасних вебсайтів, законодавчих вимог до їх функціонування в Україні та кращих практик у сфері створення сайтів для модельних агентств. Ці знання будуть використані при розробці нового вебсайту з урахуванням потреб цільової аудиторії та бізнес-цілей замовника.

2 ПРОЄКТУВАННЯ ВЕБСАЙТУ МОДЕЛЬНОГО АГЕНТСТВА

2.1 Розробка структури вебсайту

Вебсайт має певну структуру, яка включає в себе розподіл сторінок, категорій, підкатегорій і продуктів, відображаючи логічні зв'язки між ними. Навігація сайту з технічної сторони є послідовністю URL-адрес, тісно пов'язаних з семантичним ядром, яке визначає необхідний контент на сайті. Є різниця між зовнішньою та внутрішньою структурами сайту, де зовнішня вказує на розташування блоків на сторінці, а внутрішня - на відношення сторінок до категорій і матеріалів.

Залежно від цілей сайту, можна вибрати різні типи структур:

- алфавітна організація, що дозволяє розташувати інформацію за алфавітом;
- хронологічна для новин та прес-релізів, що дозволяє орієнтуватися по датах;
- деревоподібна структура, типова для інтернет-магазинів, де від головної сторінки відходять категорії і продукти;
- тематична, де вся інформація розташована за темами;
- лінійна структура у формі ланцюжка з визначеним маршрутом;
- гратчаста, де компоненти розміщені в окремих розділах;
- павутинна, що комбінує декілька типів, але застосовується рідко;
- гібридна, яка також комбінує декілька типів для полегшення пошуку.

Ієрархічна структура допомагає швидко знаходити необхідну інформацію, а приклад такої структури показаний на рисунку 2.1. Сайт включає в себе такі елементи:

- основне навігаційне меню (шапка);
- інформаційні сторінки модельного агентства;

Використання основного меню дозволяє переміщатися на потрібні сторінки сайту, такі як головна сторінка, про агентство, дівчата/хлопці, нові

обличчя, статі моделлю, контакти. У результаті вебсайт складається з шести сторінок (рис. 2.1).

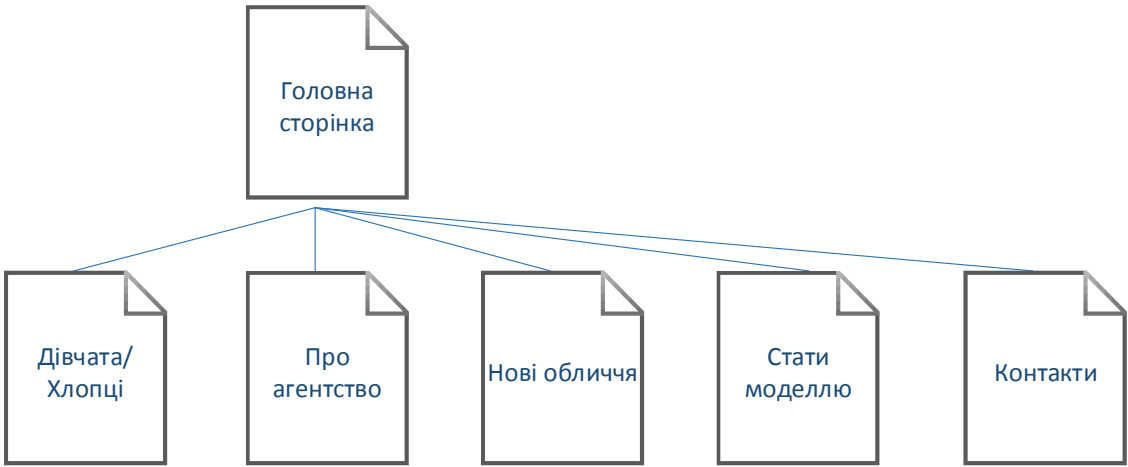


Рисунок 2.1 – Структура вебсайту

2.2 Розробка бази даних

База даних вебсайту модельного бюро містить дані про учасників (табл. 2.1), деталі учасників (табл. 2.2), портфоліо фотосесій (табл. 2.3), а також запити на послуги модельного бюро (табл. 2.4).

Таблиця 2.1 – Сутності таблиці «participant»

Сутність	Тип	Припустиме значення	Ключ
partId	числовий	Not null	+
uniqId	числовий	Not null	
townId	числовий	null	
partName	текстовий	—	
partBio	текстовий	null	
partStart	дата	—	
score	числовий	—	

Таблиця 2.2 – Сутності таблиці «participantInfo»

Сутність	Тип	Припустиме значення	Ключ
uniqId	числовий	Not null	+
firstName	текстовий	Not null	
lastName	текстовий	Not null	
phone	текстовий	Not null	
email	текстовий	Not null	

Таблиця 2.3 – Сутності таблиці «portfolio»

Сутність	Тип	Припустиме значення	Ключ
uniqId	числовий	Not null	+
photoSetName	текстовий	Not null	
photo	текстовий	Not null	
details	текстовий	Not null	

Таблиця 2.4 – Сутності таблиці «booking»

Сутність	Тип	Припустиме значення	Ключ
bookingId	числовий	Not null	+
clientFirstName	текстовий	Not null	
clientLastName	текстовий	Not null	
clientPhone	текстовий	Not null	
clientEmail	текстовий	Not null	
bookingDetails	текстовий	null	

Модель бази даних, яка була створена в phpMyAdmin, зображена на рисунку 2.2.

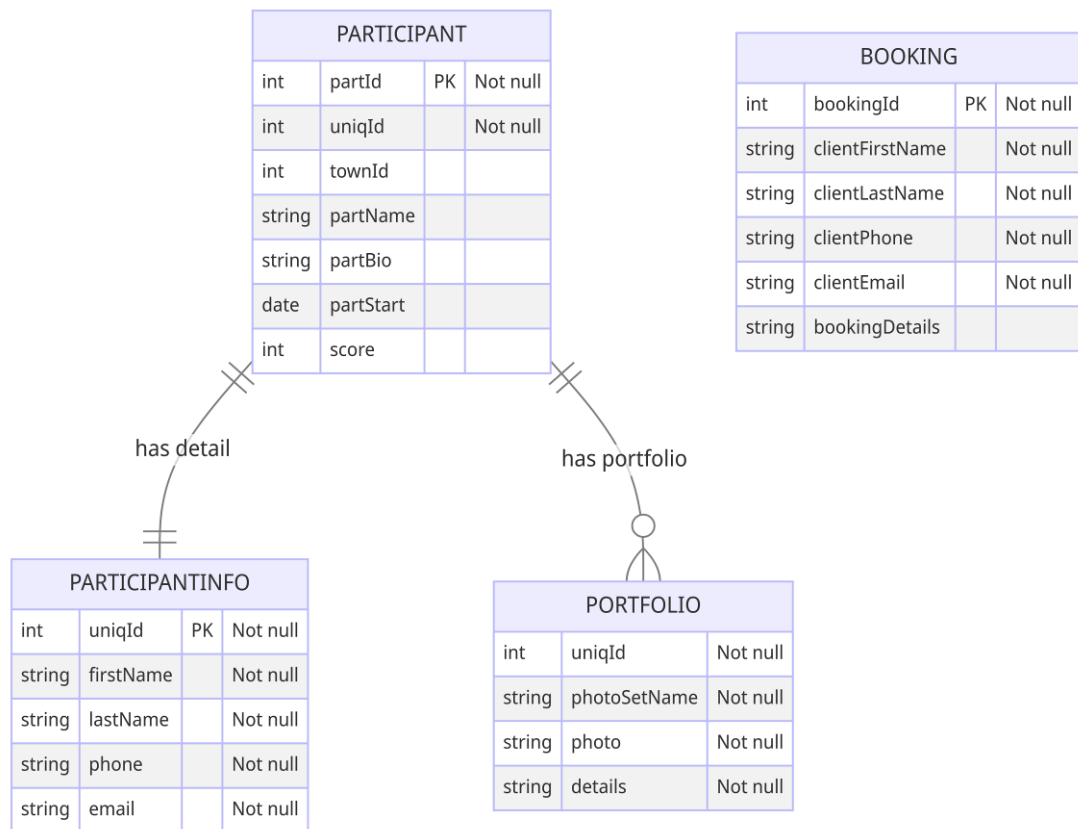


Рисунок 2.2 – Модель бази даних

2.3 Розробка дизайну вебсайту

Модельне агентство Астрал, яке вирізняється своїм широким спектром спеціалізацій, має велику базу даних, що включає фотомоделей, акторів, танцюристів та дитячих моделей: від моделей для подіуму, виставок і хостес до моделей для фото і відео реклами, а також акторів для кіно- та відеозйомок і проведення кастингів різної складності.

При створенні дизайну особлива увага приділялась рафінованій комбінації кольорів і текстур. Завдяки цьому відвідувачі сайту зосереджують свою увагу на ключових елементах, за допомогою яскравих кольорів: темно-сірого, чорного та білого. Відвідувачі та потенційні клієнти з легкістю знайдуть на сайті корисну для себе інформацію.

Головна сторінка містить важливу та актуальну інформацію про послуги модельного агентства, зокрема, відповідає на питання, чому слід вибрати Астрал, представляє основні послуги, відгуки клієнтів, посилання на соціальні мережі, контактну інформацію, а також заголовок і підвал сайту (рис. 2.3). Заголовок включає меню з посиланнями на різні сторінки сайту та назву агентства. Підвал сайту містить лінії з посиланнями на соціальні мережі компанії, телефонний номер та електронну пошту.

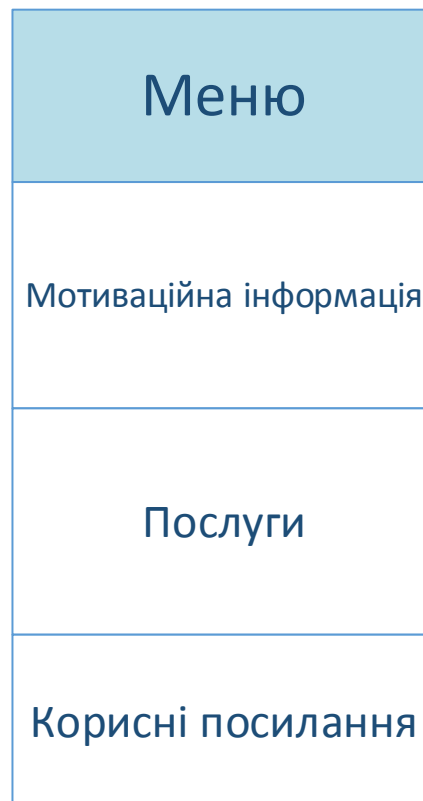


Рисунок 2.3 – Дизайн головної сторінки

Сторінка "Агентство" висвітлює історію створення компанії. Включає текстовий контент, інформацію про переваги співпраці з агентством, основних управлінців агентства, заголовок та підвал сайту (рис. 2.4).

Меню
Історія створення
Мотиваційна інформація
Команда
Корисні посилання

Рисунок 2.4 – Дизайн сторінки «Агентство»

Сторінка "Моделі" – одна з ключових секцій сайту з деталями про послуги модельного агентства та описом основних категорій моделей (рис. 2.5).

Меню
Послуги
Моделі
Корисні посилання

Рисунок 2.5 – Дизайн сторінки «Дівчата/Хлопці»

Сторінка "Нові обличчя" пропонує інформацію про новачків, які нещодавно приєдналися до агентства (рис. 2.6).



Рисунок 2.6 – Сторінка "Нові обличчя"

На кожній сторінці розміщено нижній заголовок з корисними посиланнями, переліком послуг агентства, контактною інформацією для замовлень та посиланнями на соціальні мережі.

2.4 Вибір методів та мов розробки

Вибір методів та мов розробки є ключовим етапом у процесі створення вебсайту модельного агентства. Від правильності вибору залежить не лише успішність реалізації проєкту, але й його масштабованість, ефективність та зручність подальшої підтримки. Сучасні вебтехнології пропонують широкий спектр інструментів для розробки, тому важливо зробити вибір, який найкраще відповідає вимогам та цілям проєкту.

Для розробки клієнтської частини вебсайту (front-end) було обрано мову розмітки HTML5 та каскадні таблиці стилів CSS3. HTML5 є стандартом для структурування та представлення контенту на вебсторінках. Він дозволяє створювати семантично значущу розмітку, що покращує доступність сайту та

спрощує його індексацію пошуковими системами. CSS3, в свою чергу, використовується для візуального оформлення HTML-елементів та створення адаптивного дизайну, який коректно відображається на різних пристроях [9].

З метою додання інтерактивності та динамічності сторінкам сайту було вирішено використовувати мову програмування JavaScript. JS дозволяє реалізувати такі функції як валідація форм, обробка подій, асинхронна комунікація з сервером тощо. Для прискорення та спрощення розробки доцільно використовувати один з популярних JS-фреймворків, таких як React, Angular або Vue.js. Вони надають готові компоненти та шаблони, а також забезпечують ефективне оновлення DOM-дерева та управління станом додатку [10].

Для розробки серверної частини (back-end) вебсайту було обрано мову програмування PHP та фреймворк Laravel. PHP є одним з найпопулярніших back-end рішень завдяки своїй простоті, гнучкості та широкій підтримці спільнотою розробників. Laravel, в свою чергу, є потужним та зручним PHP-фреймворком, який реалізує архітектурний шаблон MVC (Model-View-Controller) та надає безліч вбудованих функцій, як от ORM, шаблонізатор Blade, система маршрутизації, авторизація тощо.

Для зберігання даних вебсайту (інформації про моделі, користувачів, замовлення тощо) було обрано реляційну базу даних MySQL. MySQL є безкоштовною, надійною та швидкою СУБД, яка ідеально підходить для малих та середніх проєктів. Вона має зрозумілий синтаксис SQL та ефективні механізми індексації та кешування запитів. Для спрощення взаємодії з БД з коду додатку в Laravel використовується ORM Eloquent, який дозволяє працювати з таблицями як з об'єктами PHP [11].

Для забезпечення адаптивності та кросбраузерності верстки сайту було вирішено використовувати CSS-фреймворк Bootstrap. Він надає набір готових класів та компонентів (сітки, кнопки, форми, модальні вікна тощо), які автоматично адаптуються під розміри екрану та коректно відображаються

в різних браузерах. Це значно прискорює верстку та мінімізує потребу в написанні власних стилів.

З метою автоматизації рутинних задач розробки, як от компіляція та мініфікація ресурсів (CSS, JS), автоматичне оновлення сторінки при зміні файлів, транспіляція коду тощо, було вирішено використовувати збірник проєктів Webpack. Він дозволяє створити гнучкий конфігураційний файл з правилами обробки різних типів ресурсів та запускати їх збірку одною командою. Також Webpack має розвинену систему плагінів та ладерів, які розширюють його можливості [12].

Для контролю версій коду було обрано систему Git та хостинг репозиторіїв GitHub. Git дозволяє ефективно відслідковувати зміни в коді, створювати гілки для реалізації нових функцій та зручно об'єднувати їх. GitHub, окрім безпосереднього хостингу репозиторіїв, надає такі корисні функції як issue-трекер, pull-request, GitHub Actions для неперервної інтеграції тощо.

Для забезпечення якості коду було вирішено використовувати статичні аналізатори, такі як ESLint для JavaScript-коду та PHP_CodeSniffer для PHP-коду. Вони дозволяють виявляти потенційні помилки, невідповідності стандартам кодування та стилістичні огріхи ще до запуску коду. Також для тестування функціоналу сайту доцільно написати модульні (юніт) та інтеграційні тести за допомогою фреймворків PHPUnit та Jest [13].

Для розгортання сайту на реальному сервері було обрано використовувати Docker-контейнери. Docker дозволяє «запакувати» додаток з усіма його залежностями в ізольоване середовище (образ), яке потім можна розгорнути на будь-якому хості з встановленим Docker. Це значно спрощує процес розгортання та мінімізує ризики виникнення конфліктів між різними версіями ПЗ на сервері. Для оркестрації кількох контейнерів (вебсервер, БД, кеш тощо) зручно використовувати інструмент docker-compose.

Такий вибір технологій та інструментів дозволяє створити сучасний, ефективний та масштабований вебсайт, який відповідає найкращим

практикам розробки та здатний вирішувати поставлені бізнес-задачі. При цьому варто розуміти, що технології в світі веброботи дуже швидко розвиваються, тому важливо постійно слідкувати за трендами та бути готовим до впровадження нових інструментів у міру необхідності (табл. 2.5).

Таблиця 2.5 – Стек технологій

Область застосування	Обрані технології та інструменти	Обґрунтування вибору
Front-end	- HTML5 - CSS3 - JavaScript - Фреймворк (React, Angular, Vue.js)	- Семантична розмітка та структурування контенту - Візуальне оформлення та адаптивний дизайн - Інтерактивність та динамічність сторінок - Прискорення та спрощення розробки
Back-end	- PHP	- Простота, гнучкість, популярність - Архітектура MVC, вбудовані функції (ORM, шаблонізатор, маршрутизація)
База даних	- MySQL	- Безкоштовність, надійність, швидкість - Зрозумілий синтаксис SQL, ефективні механізми індексації та кешування
CSS-фреймворк	- Bootstrap	- Набір готових адаптивних компонентів - Кросбраузерність та прискорення верстки
Збірник проєктів	- Webpack	- Автоматизація рутинних задач розробки – Гнучкий конфігураційний файл та плагіни
Контроль версій	- Git	- Відслідковування змін, робота з гілками - Хостинг репозиторіїв та додаткові функції (issue-трекер, CI/CD)
Якість коду	- ESLint - PHP_CodeSniffer - PHPUnit, Jest	- Статичний аналіз та виявлення помилок - Перевірка на відповідність стандартам кодування - Модульне та інтеграційне тестування
Розгортання	- Docker	- Стандартизація середовища розробки та розгортання - Легкість використання та сумісність з CI/CD процесами

2.5 Висновок

У другому розділі було проведено проєктування вебсайту модельного агентства, яке включало такі етапи.

Розробка структури вебсайту. Було обрано ієрархічну структуру, яка складається з шести основних сторінок: головна, про агентство, дівчата/хлопці, нові обличчя, статі моделлю та контакти. Така структура дозволяє користувачам швидко знаходити потрібну інформацію та орієнтуватися на сайті.

Розробка бази даних. Було створено чотири таблиці для зберігання даних про учасників, деталі учасників, портфоліо фотосесій та запити на послуги модельного агентства. Правильно спроектована база даних є основою для ефективного функціонування динамічного вебсайту.

Розробка дизайну вебсайту. При створенні дизайну особлива увага приділялася рафінованій комбінації кольорів і текстур. Було розроблено макети основних сторінок сайту: головної, про агентство, моделі, нові обличчя. Привабливий та зручний дизайн є важливим фактором для залучення та утримання відвідувачів на сайті.

Вибір методів та мов розробки. Для розробки сайту було обрано сучасний стек вебтехнологій, який включає HTML5, CSS3, JavaScript, PHP, Laravel, MySQL, Bootstrap, Webpack, Git тощо. Такий вибір дозволяє створити ефективний, масштабований та зручний в підтримці вебсайт, що відповідає найкращим практикам розробки.

3 РОЗРОБКА ВЕБСАЙТУ МОДЕЛЬНОГО АГЕНТСТВА

3.1 Програмна частина

На рисунку 3.1 показана частина HTML код, яка структурує вебсторінку і включає заголовок та дві панелі навігації. На вебсторінці вказано назву "Ukrainian Models" в тегу <title>, і використовуються налаштування для адаптивного дизайну в тегу <meta>, щоб сторінка коректно відображалась на різних пристроях.

Розділ <header> містить логотип на верху сторінки, який веде на домашню сторінку, і дві панелі навігації (<nav>). Перша панель включає посилання на різні розділи сайту, такі як "Home", "About us", "Blog", "Model", "Show", і "Contact". Кожне посилання має відповідне гіперпосилання (<a href>). Друга панель, "search-bar", містить текстове поле для пошуку та кнопку для активації пошуку, проте функція JavaScript toggleSearch(), яка має бути прив'язана до кнопки, має бути визначена в іншій частині коду.

Кнопка з класом "profile-btn" передбачена для переключення профілю, і має атрибут onclick зі значенням "toggleProfile()".

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ukrainian Models</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <nav class="navbar">
      <div class="logo">
        <a href="index.html"></a>
      </div>
      <ul class="nav-links">
        <li><a href="index.html">Головна</a></li>
        <li><a href="about.html">Про агенцію</a></li>
        <li><a href="girls.html">Дівчата</a></li>
        <li><a href="boys.html">Хлопці</a></li>
        <li><a href="new-face.html">Нове обличчя</a></li>
        <li><a href="become-model.html">Стати моделлю</a></li>
      </ul>
      <div class="search-profile">
        <button class="search-btn" onclick="toggleSearch()">Q</button>
        <button class="profile-btn" onclick="toggleProfile()">👤</button>
        <div class="profile-dropdown" id="profileDropdown">
          <a href="profile.html">Профіль</a>
          <a href="settings.html">Налаштування</a>
          <a href="logout.html">Вийти</a>
        </div>
      </div>
    </nav>
    <div class="search-bar" id="searchBar">
      <input type="text" placeholder="Пошук...">
      <button>Шукати</button>
    </div>
  </header>
  <script src="script.js"></script>
```

Рисунок 3.1 – Меню вебсайту

На рисунку 3.2 частина коду JavaScript, що містить функції для управління відображенням елементів на вебсторінці. Функція `toggleSearch()` керує відображенням пошукової панелі: якщо панель пошуку є невидимою, вона стає видимою, і навпаки. Функція `toggleProfile()` управляє відображенням випадаючого меню профілю, перемикаючи його між станами видимості та невидимості. Обидві функції використовують стиль CSS `display` для зміни відображення елементів. Також є обробник подій, прикріплений до документа, який слухає кліки і перевіряє, чи містить відповідний елемент клас `profileDropDown`, активуючи функцію `toggleProfile()` при кліці на елементі, який містить цей клас. Це забезпечує динамічну інтерактивність елементів навігації на сторінці.

```
function toggleSearch() {
    const searchBar = document.getElementById('searchBar');
    if (searchBar.style.display === 'flex') {
        searchBar.style.display = 'none';
    } else {
        searchBar.style.display = 'flex';
    }
}

function toggleProfile() {
    const profileDropdown = document.getElementById('profileDropdown');
    if (profileDropdown.style.display === 'block') {
        profileDropdown.style.display = 'none';
    } else {
        profileDropdown.style.display = 'block';
    }
}

document.addEventListener('click', function(event) {
    const profileDropdown = document.getElementById('profileDropdown');
    const profileBtn = document.querySelector('.profile-btn');
    if (!profileBtn.contains(event.target)) {
        profileDropdown.style.display = 'none';
    }
});
```

Рисунок 3.2 – Елементи Java script

Рисунок 3.3 наводить частину SQL коду, який описує створення двох таблиць в базі даних для модельного агентства. Перша таблиця називається PARTICIPANT і містить поля participant_id (унікальний ідентифікатор учасника, який є первинним ключем), name (ім'я учасника), birthdate (дата народження), participation_date (дата участі у заході), score (бали, отримані

учасником). Друга таблиця називається PARTICIPATION і включає поля event_id (ідентифікатор події, первинний ключ), participant_id (ідентифікатор учасника), phone (телефонний номер), email (електронна адреса). Ці таблиці призначені для зберігання інформації про учасників та їх участь у різних заходах, дозволяючи легко управляти даними через SQL-запити.

```
CREATE DATABASE modeling_agency;

USE modeling_agency;

CREATE TABLE PARTICIPANT (
    partId INT AUTO_INCREMENT PRIMARY KEY,
    uniqId INT NOT NULL,
    townId INT,
    partName VARCHAR(255),
    partBio TEXT,
    partStart DATE,
    score INT
);

CREATE TABLE PARTICIPANTINFO (
    uniqId INT PRIMARY KEY,
    firstName VARCHAR(255) NOT NULL,
    lastName VARCHAR(255) NOT NULL,
    phone VARCHAR(15) NOT NULL,
    email VARCHAR(255) NOT NULL
);
```

Рисунок 3.3 – Елементи створення бази даних

Рисунок 3.4 показує код, який створює вебформу, яка дозволяє користувачеві вибрати метод сортування моделей на сторінці. Форма містить радіо кнопки для вибору типу сортування: за зростанням або спаданням. Користувач може вибрати один із трьох варіантів сортування: "новіші першими", "ім'я за зростанням", або "ім'я за спаданням". Після вибору опції, користувач може надіслати форму натисненням на кнопку "Submit". Обрані параметри сортування відправляються на сервер через PHP-скрипт fetch_models.php, який обробляє дані. Також, у формі є кнопка, що викликає

JavaScript функцію `fetchModels()`, яка ймовірно здійснює AJAX-запит для отримання даних без перезавантаження сторінки.

```

        <input type="text" name="eye_color" />
    </div>
    <div class="filter-group">
        <label>Колір Волосся:</label>
        <input type="text" name="hair_color">
    </div>
    <button type="submit">Фільтрувати</button>
</form>
</aside>
<main>
    <h2>Сортувати за</h2>
    <form method="GET" action="models.php">
        <label>
            <input type="radio" name="sort" value="newest" checked> Найновіші
        </label>
        <label>
            <input type="radio" name="sort" value="oldest"> Найстаріші першим
        </label>
        <label>
            <input type="radio" name="sort" value="name_asc"> А - Я
        </label>
        <label>
            <input type="radio" name="sort" value="name_desc"> Я - А
        </label>
        <button type="submit">Сортувати</button>
    </form>
    <div class="models-list">
        <?php include 'fetch_models.php'; ?>
    </div>
</main>
</div>
</body>
</html>

```

Рисунок 3.4 – Елемент форми сортування

Рисунок 3.5 це програмний код PHP створює фільтри для пошуку, використовуючи параметри, отримані через GET-запити. Цей фрагмент коду перевіряє, чи встановлені певні параметри у запиті (наприклад, вік, зріст, вага), і якщо вони присутні, додає відповідні критерії до змінної `$filters`, яка здається використовується для фільтрації результатів пошуку. Код обробляє різні атрибути, такі як зріст (`height`), вік (`age`), розмір плаття (`dress_size`), розмір панталонів (`pant_size`), розмір взуття (`shoe_size`), колір очей (`eye_color`) та колір волосся (`hair_color`). Для кожного параметру, якщо він не порожній, відповідне значення додається до змінної `$filters` для використання у запиті до бази даних або іншому контексті обробки даних.

```

/* фільтри
$filters = [];
.f (!empty($_GET['height'])) {
    $filters[] = "partBio LIKE '%Height: {"$_GET['height']}%'";

.f (!empty($_GET['age'])) {
    $filters[] = "partBio LIKE '%Age: {"$_GET['age']}%'";

.f (!empty($_GET['dress_size'])) {
    $filters[] = "partBio LIKE '%Dress Size: {"$_GET['dress_size']}%'";

.f (!empty($_GET['pant_size'])) {
    $filters[] = "partBio LIKE '%Pant Size: {"$_GET['pant_size']}%'";

.f (!empty($_GET['shoe_size'])) {
    $filters[] = "partBio LIKE '%Shoe Size: {"$_GET['shoe_size']}%'";

.f (!empty($_GET['top_size'])) {
    $filters[] = "partBio LIKE '%Top Size: {"$_GET['top_size']}%'";

.f (!empty($_GET['eye_color'])) {
    $filters[] = "partBio LIKE '%Eye Color: {"$_GET['eye_color']}%'";

.f (!empty($_GET['hair_color'])) {
    $filters[] = "partBio LIKE '%Hair Color: {"$_GET['hair_color']}%'";

```

Рисунок 3.5 – Елементи фільтру

Рисунок 3.6 це PHP скрипт виконує запит до бази даних для отримання інформації про учасників і виводить їх дані на вебсторінці. Скрипт розпочинається із створення SQL-запиту, який вибирає всі записи з таблиці `PARTICIPANT`. Якщо змінна `\$filters`, що містить фільтри з попереднього коду, не порожня, до запиту додається умова `WHERE` з цими фільтрами. Також можливе сортування даних за якоюсь характеристикою через змінну `\$sort`.

Після формування запиту, скрипт виконує його через об'єкт з'єднання `\$conn` і перевіряє, чи є результати. Якщо результати існують, вони виводяться в циклі `while`, де для кожного рядка створюється HTML-блок з класом `model-item`, що відображає зображення моделі, ім'я, специфічні дані якісь спеціальні характеристики (зміст цих полів не уточнений). Якщо результатів немає, виводиться повідомлення "Немає результатів.". Наприкінці скрипт закриває з'єднання з базою даних.

```

// Побудова запиту
$query = "SELECT * FROM PARTICIPANT";
if (!empty($filters)) {
    $query .= " WHERE " . implode(" AND ", $filters);
}
$query .= " ORDER BY $sort";

$result = $conn->query($query);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<div class='model-item'>";
        echo "<img src='path_to_image/{ $row['partId'] }.jpg' alt='{ $row['partName'] }'>";
        echo "<div class='details'>";
        echo "<strong>{ $row['partName'] }</strong>";
        echo "<span>Вік: { $row['partBio'] }</span>";
        echo "<span>Зріст: { $row['partBio'] }</span>";
        echo "<span>Розмір (Сукня): { $row['partBio'] }</span>";
        echo "</div>";
        echo "</div>";
    }
} else {
    echo "Немає результатів.";
}

$conn->close();
?>

```

Рисунок 3.6 – Запит до бази даних

3.2 Інтерфейс користувача

На рисунку 3.7 показано меню навігації сайту модельного агентства Ukrainian Models. Меню дозволяє переходити між основними розділами сайту - головною, інформацією про агентство, портфолію дівчат та хлопців, новими обличчями та сторінкою для подачі заявки в моделі. Таке меню є зручним та інтуїтивно зрозумілим для користувачів, які шукають моделей або хочуть співпрацювати з агентством. Меню вдало організовує контент сайту та полегшує навігацію по ньому.



Рисунок 3.7 – Меню вебсайту

На рисунку 3.8 можна побачити головну частину вебсайту модельного агентства, яка включає інформаційний заголовок та пошуковий рядок. Головний заголовок вітає відвідувачів зі словами "Ласкаво просимо в

Модельне агентство" і пропонує нижче короткий опис послуг агентства, яке спеціалізується на пошуку та представленні моделей. Вказується, що агентство займається останніми тенденціями моди, краси, стилю життя та культури.

Функціональність пошуку реалізована через введення тексту у поле з підказкою "Пошук моделей, талантів, подій, новин та більше" та кнопкою "Пошук", що дозволяє відвідувачам швидко знайти відповідну інформацію на сайті. Цей елемент вебсайту, ймовірно, інтегрований із базою даних агентства, дозволяючи користувачам переглядати, фільтрувати, і сортувати дані про моделей та асоційовані події або новини.

Загальний дизайн і використання нейтральних тонів у візуальному оформленні сприяють зосередженості користувачів на тексті та функціональних елементах сторінки.

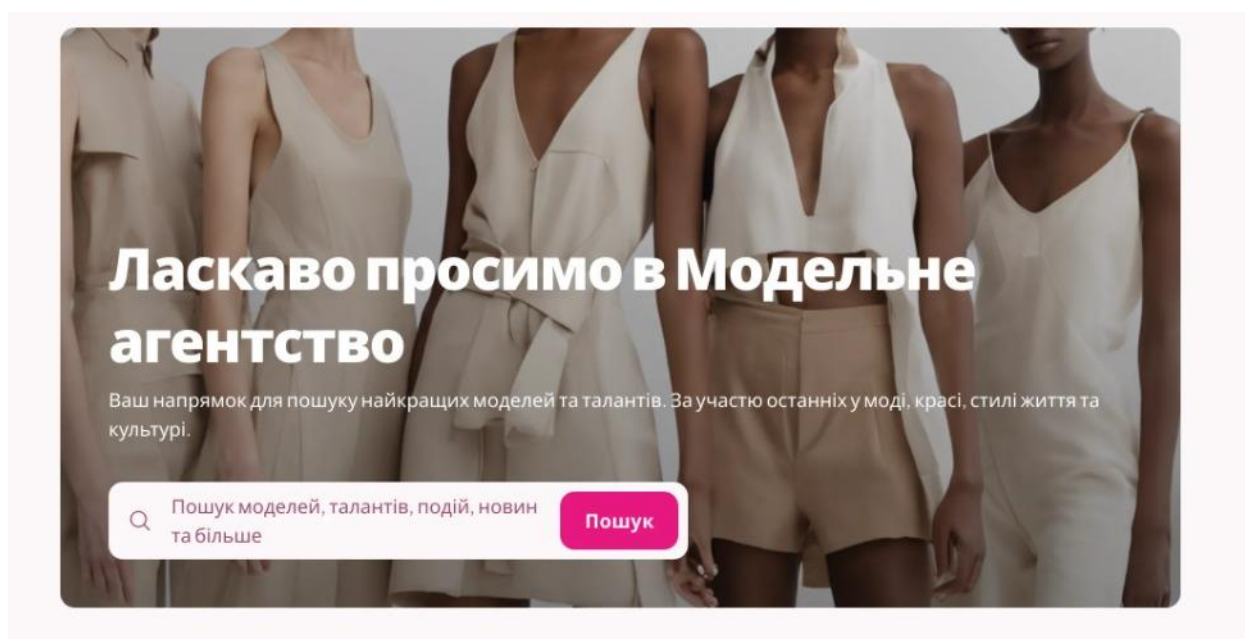


Рисунок 3.8 – Головна сторінка

На рисунку 3.9 відображено частину вебсайту модельного агентства, присвячену інформації "Про агентство". Текст розповідає про успіхи моделей агентства на міжнародному рівні, зокрема участь у великих модних показах і появи в таких виданнях як Vogue. Описано місію агентства: надання

персоналізованого сервісу, сприяння розвитку кар'єр моделей і забезпечення клієнтів найкращими моделями. Зміст передає професійний підхід агентства та його орієнтованість на якість.

Внизу є чотири візуальні елементи з короткими описами: "Моделі", "Обкладинки Vogue", "Подіумні покази", та "Найбільший бренд", кожен з яких вказує на різні аспекти успіхів агентства. Ці елементи ймовірно ведуть до додаткових сторінок або галерей, що демонструють роботу моделей, їхні успіхи в модних журналах, на подіумах, а також співпрацю з відомими брендами.

Такий дизайн спрямований на залучення нових клієнтів і моделей, показуючи високий професійний рівень агентства та його досягнення в індустрії моди.

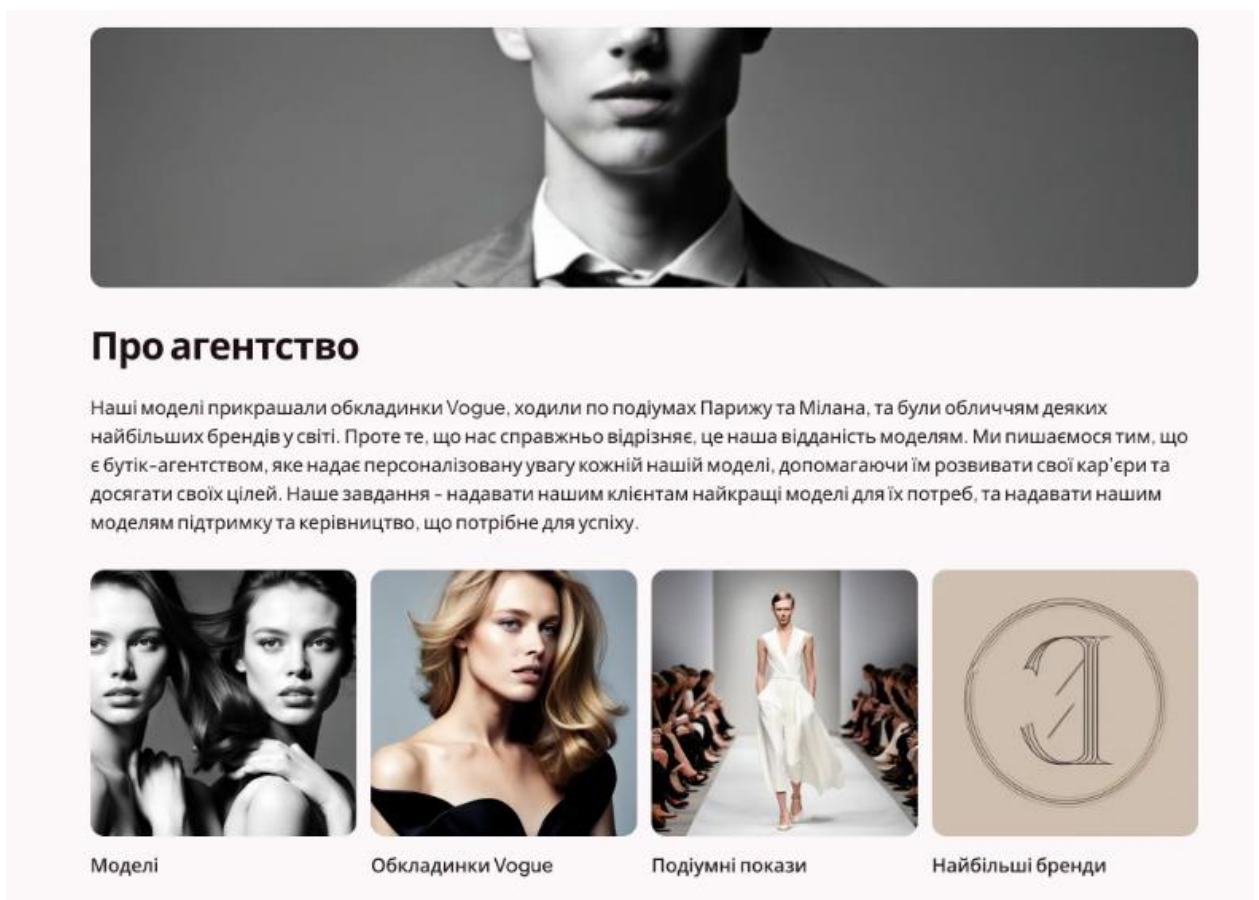


Рисунок 3. 9 -Сторінка опису агентства

Рисунок 3.10 це частина вебсайту відображає розділ "Останні новини" на сторінці модельного агентства, який призначений для інформування відвідувачів про актуальні події та особливості світу моделінгу. Сторінка включає дві новинні картки, кожна з яких має зображення, заголовок новини, короткий опис події та кнопку "Читати далі", що дозволяє користувачам перейти до повного тексту статті.

Цей розділ сприяє залученню відвідувачів, надаючи їм цінну та актуальну інформацію, а також демонструє залученість агентства до важливих подій у світі моди. Він також виконує роль освітнього ресурсу для моделей та інших зацікавлених осіб.

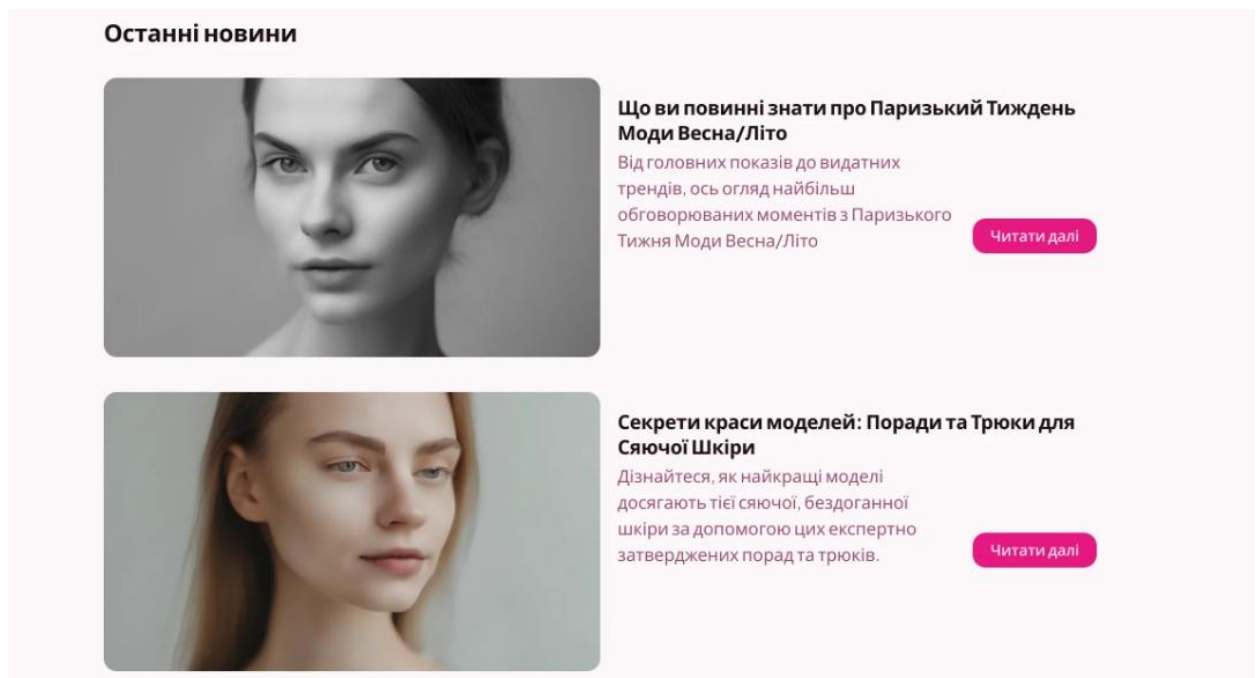


Рисунок 3.10 – Новини

На рисунку 3.11 відображено інтерфейс користувача для функції пошуку і сортування в базі даних моделей на вебсайті модельного агентства. Цей розділ включає численні фільтри для деталізованого пошуку моделей за різними параметрами, такими як стать, вік, розміри одягу та взуття.

Користувачі можуть вибрати спосіб сортування результатів пошуку: за новизною, від А до Я, або налаштовані параметри, які можуть включати

сортування за рейтингом або іншими критеріями. Внизу сторінки є таблиця, що показує результати пошуку з деталями про моделі, такі як ім'я, вік, зріст, розміри одягу та розмір взуття (рис. 3.12).

Цей інтерфейс забезпечує ефективне і гнучке управління даними про моделей, дозволяючи користувачам швидко знаходити необхідну інформацію та адаптувати відображення результатів до своїх потреб. Це є чудовим інструментом для кастинг-директорів і клієнтів, які шукають моделей для різних проектів і подій.

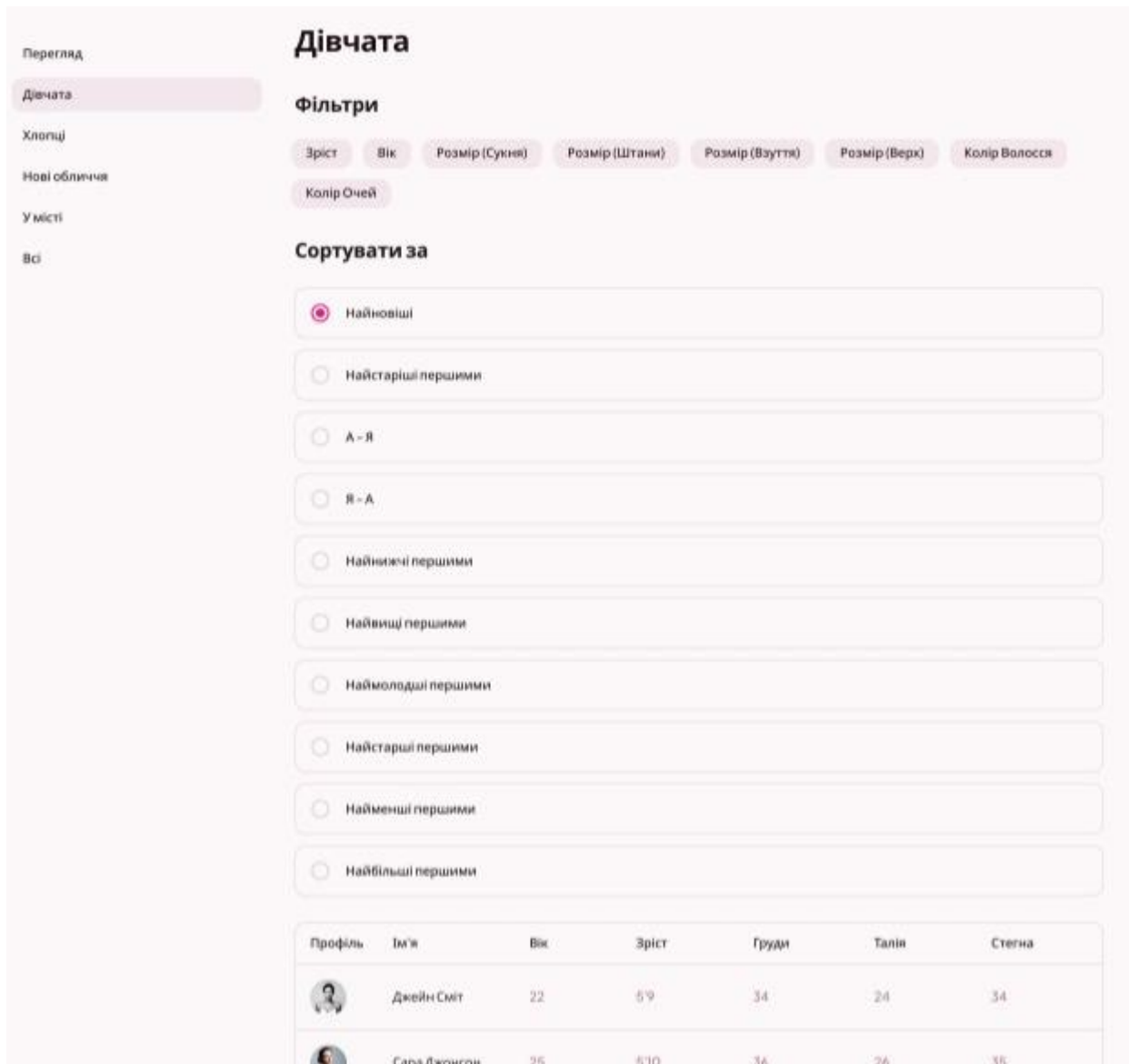


Рисунок 3.11 – Фільтр моделей

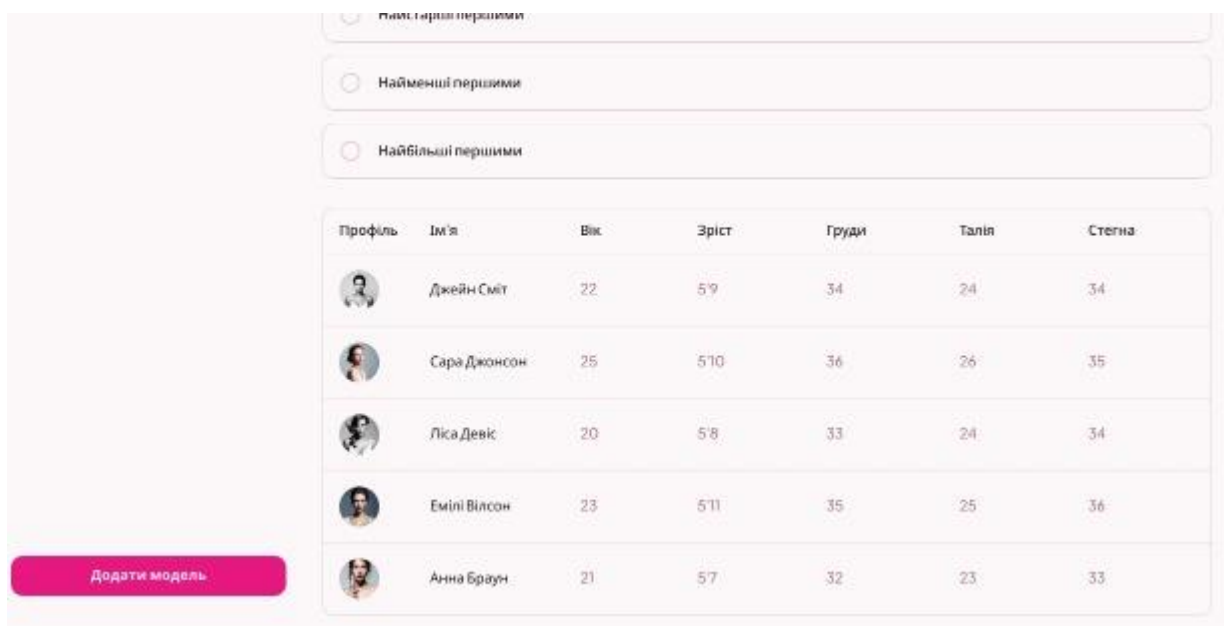


Рисунок 3.12 – Сортуння моделей

На рисунку 3.13 відображено розділ вебсайту модельного агентства під назвою "Нові моделі". Ця сторінка призначена для презентації нових моделей, які нещодавно приєднались до агентства. Верхня частина сторінки містить заголовок і короткий опис, що закликає відвідувачів відкрити для себе нові обличчя та таланти, пропонуючи можливість взаємодії з моделями або отримання додаткової інформації про них.

Нижче розміщені профілі кількох моделей, кожен із яких включає фотографію, ім'я моделі та її основні параметри: зріст, вагу, розміри одягу та розмір взуття. Це дає користувачам змогу швидко оцінити основні характеристики кожної моделі.

Цей розділ є важливим інструментом для залучення уваги клієнтів агентства, дизайнерів та інших зацікавлених сторін, які можуть шукати нові обличчя для рекламних кампаній, модних показів або фотосесій. Він також допомагає моделям отримати більше визнання та можливості для розвитку їхніх кар'єр.

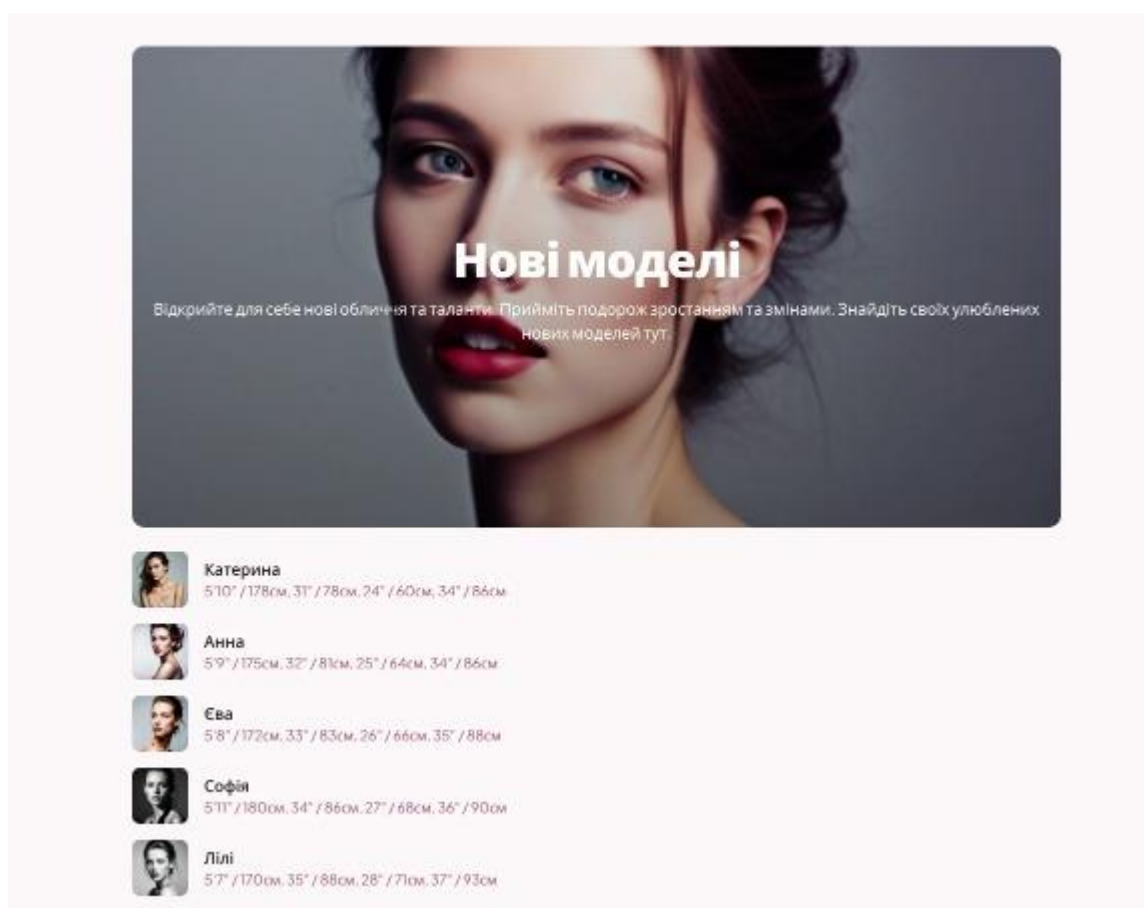


Рисунок 3.13 – Сторінка нових моделей

На рисунку 3.14 представлено форму реєстрації для бажаючих стати моделлю на вебсайті модельного агентства. У формі потрібно ввести повне ім'я, нік у соціальній мережі Instagram, стать, дату народження, контактний номер телефону, країну та місто проживання. Також потрібно надати інформацію про основні фізичні параметри: зріст, обхвати грудей, талії та бедер, розмір взуття, колір очей та волосся. Додатково запитують про наявність тату і пірсингу, вік цих елементів, та необхідно завантажити кілька фотографій: портрет, фото в профіль та фото в повний зріст. В кінці форми є кнопка "Надіслати", через яку можна відправити всю зібрану інформацію агентству для подальшого аналізу та можливості співпраці.

Стати моделлю

Ми завжди шукаємо наступний великий проект. Якщо ви готові стати моделлю, ми хочемо вас побачити.

Ім'я

Прізвище

Стать

Електронна пошта

Король

Місто

Дата народження

Номер телефону

Виміри

Зріст

Обхват грудей

Талія

Стегна

Завантажити фото

Будь ласка, завантажте 3 фото: (портрет, повний ріст і профіль)

Завантажити

Опис

Розкажіть про себе

Надіслати

Рисунок 3.14 - Форма для нових моделей

3.3 Висновок

У третьому розділі було представлено практичну реалізацію вебсайту модельного агентства, що включала розробку програмної частини та створення зручного та привабливого інтерфейсу користувача.

Програмна частина сайту була розроблена з використанням сучасних вебтехнологій, таких як HTML, CSS, JavaScript та PHP. Було створено структуру сторінок, меню навігації та функціональні елементи, такі як форма пошуку моделей та фільтри за різними параметрами. Також було реалізовано

взаємодію з базою даних для зберігання та відображення інформації про моделей, їх портфоліо та фізичні параметри.

Головна сторінка сайту містить основну інформацію про агентство, його послуги та переваги. Сторінка "Про агентство" детальніше розповідає про історію та досягнення компанії. Розділ "Моделі" дозволяє переглядати портфоліо моделей з можливістю фільтрації за різними критеріями. Сторінка "Нові обличчя" презентує молодих талантів, які нещодавно приєднались до агентства. Також було створено форму для подачі заявок на співпрацю з агентством у якості моделі.

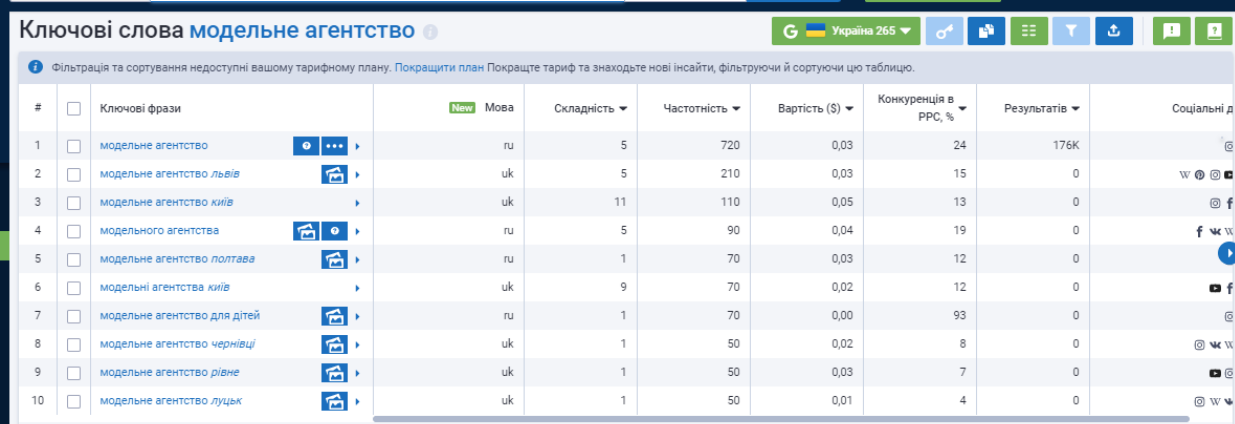
Розроблений інтерфейс є інтуїтивно зрозумілим, легким у навігації та візуально привабливим. Він забезпечує ефективну взаємодію між відвідувачами сайту (потенційними клієнтами та моделями) та модельним агентством. Дизайн сайту відповідає сучасним трендам та підкреслює професійність і високі стандарти агентства в індустрії моди.

Таким чином, практична реалізація вебсайту модельного агентства демонструє успішне втілення концепцій та рішень, розроблених на етапі проєктування. Створений сайт є повнофункціональним та готовим до використання як потужний інструмент для просування послуг агентства, залучення нових клієнтів та моделей, а також для зміцнення іміджу компанії на ринку модельного бізнесу.

4 ОПТИМІЗАЦІЯ ВЕБСИСТЕМИ І РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ

4.1 Семантичне ядро сайту

Для підвищення видимості вебсайту перед потенційними клієнтами у світі інтернету було створено семантичне ядро, яке представляє собою організований перелік запитів користувачів по темі, ранжованих від найпопулярніших до менш поширених (рис. 4.1).



#	Ключові фрази	Мова	Складність	Частотність	Вартість (\$)	Конкуренція в РРС, %	Результатів	Соціальні д
1	модельне агентство	ru	5	720	0,03	24	176K	
2	модельне агентство львів	uk	5	210	0,03	15	0	
3	модельне агентство київ	uk	11	110	0,05	13	0	
4	модельного агентства	ru	5	90	0,04	19	0	
5	модельне агентство полтава	ru	1	70	0,03	12	0	
6	модельні агентства київ	uk	9	70	0,02	12	0	
7	модельне агентство для дітей	ru	1	70	0,00	93	0	
8	модельне агентство чернівці	uk	1	50	0,02	8	0	
9	модельне агентство ріве	uk	1	50	0,03	7	0	
10	модельне агентство луцьк	uk	1	50	0,01	4	0	

Рисунок 4.1 – Список ключових слів українською мовою для «модельного агентства Запоріжжя»

Вивчивши цей список ключових слів, стає очевидно, що для вебсайту модельного агентства необхідно вказувати місцезнаходження та адресу, серед іншого. На кожній сторінці додані такі елементи як заголовок сторінки, опис та ключові слова. Наприклад:

```
<title> Ukrainian Models модельне агентство Запоріжжя</title>
```

```
<meta charset="UTF-8">
```

```
<meta name="description" content="Модельне агентство міста  
Запоріжжя, Україна">
```

```
<meta name="keywords" content="моделі, реклама, просування  
товару, модельне агентство, Запоріжжя, фото, дівчата, хлопці">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Внутрішня структура сайту була оптимізована для пошукових систем, використовуючи розташування тегів різних рівнів від h1 до h4. Тег h1 використовується у заголовку та має стиль display: none, оскільки він не призначений для візуального відображення, але важливий для пошукових систем. Тег h2 містить назви категорій та розділів, тег h3 — назви типів за категоріями. Теги інших рівнів введені залежно від потреби.

Для додаткової оптимізації для пошукових машин кожен тег зображення містить атрибут alt, що використовується браузером при слабкому з'єднанні або якщо зображення вимкнені користувачем. Заголовок доступний на всіх сторінках і містить гіперпосилання на всі інші сторінки.

4.2 Карта сайту

Сучасні пошукові системи здатні самостійно індексувати сайти за заданими параметрами, але в деяких випадках доцільно надати їм конкретні вказівки щодо правильності та послідовності дій. Використовуючи сайтмап, можна досягти наступних цілей:

- сприяти пошуковим роботам у виявленні потрібних сторінок сайту та своєчасній їх індексації;
- встановити пріоритетність сторінок сайту для індексації роботами;
- моніторити зміни (дату модифікації) на кожній сторінці сайту, навіть якщо їх кількість значна (за умови автоматичної генерації).

Крім того, сайтмап дозволяє вчасно відслідковувати періодичність оновлення сторінок і коригувати цей параметр за потреби.

Серед розробників прийнято виділяти два основні типи сайтмапів - html і xml. Кожен з них має свої особливості та призначення, розуміння яких допоможе ефективно використовувати обидва формати для власних потреб.

Сайтмап у форматі html. Це статична сторінка, яка містить перелік усіх основних сторінок сайту. Така карта сайту призначена насамперед для зручності користувачів і дозволяє одним кліком перейти на потрібну сторінку, заощаджуючи час на пошук. Зручним вважається формат у вигляді деревоподібної структури з розгорнутими сторінками. Посилання на статичну карту зазвичай розміщується у підвалі сайту. Якщо сайт має складну структуру, сайтмап може бути представлений вкладеними сторінками, згрупованими за зручними критеріями.

Сайтмап - це файл, розташований у кореневому каталозі ресурсу, який зчитується всіма доступними пошуковими системами. Його функція полягає в тому, щоб показати пошуковим системам усі адреси сторінок, які потребують індексації. Він також надає інформацію про останні оновлення розділів.

Експерти зазначають, що один такий файл може містити до 50 тисяч посилань за один раз. Якщо кількість посилань перевищує цю межу, необхідно створити кілька файлів такого типу. Додатковими перевагами є можливість вказати частоту оновлення ресурсів і пріоритетність індексації.

Правильним кодуванням для файлів такого типу вважається формат UTF-8. Також варто пам'ятати про обмеження розміру, який не повинен перевищувати 10 Мб. Карти більшого розміру також необхідно розділяти на кілька файлів.

Існує декілька способів створення правильного сайтмапу у форматах html і xml:

- ручне створення карти. Цей метод підійде більш досвідченим користувачам, які раніше працювали з основними тегами. Також він актуальний для ресурсів з невеликою кількістю сторінок;

- налаштування sitemap.xml за допомогою онлайн-сервісів. Ці сервіси працюють за схожим принципом - ви вказуєте адресу головної сторінки свого сайту (<https://example.com/>), а також необхідні налаштування (частоту обходу, пріоритетність). Зазвичай вони мають обмеження щодо кількості

посилань як для платних, так і для безкоштовних версій. Крім того, не всі сервіси надають гнучкі налаштування карти сайту (наприклад, неможливо встановити пріоритетність або вона встановлюється однаковою для всіх сторінок сайту тощо), тому таку карту потрібно допрацьовувати вручну;

- використання програм-генераторів карт сайтів. Такі програми широко використовуються як досвідченими фахівцями, так і новачками в цій галузі. Є як безкоштовні, так і платні аналоги з різним набором функцій. Схема дій аналогічна онлайн-сервісам;

- створення карти xml за допомогою плагіна для CMS або за допомогою движка. Ще одним хорошим методом є автоматична генерація карти сайту. Цей метод має ряд переваг, таких як налаштування під потрібні параметри і максимальну кількість можливих адрес. Крім того, при використанні плагінів CMS після внесення змін на сторінках дата оновлення буде автоматично відображатися на карті сайту ресурсу, а нові сторінки також будуть автоматично додаватися до карти.

На жаль, не існує загальноприйнятих стандартів щодо частоти оновлення. Це залежить від частоти створення нових сторінок, зміни адрес або контенту, а також наявності зв'язку з уже існуючими сторінками. Важливо пам'ятати, що не слід зловживати маніпуляціями з частотою обходу (changefreq) та оновлювати дату зміни на сторінці, якщо вона насправді не змінювалася. Наприклад, якщо встановити частоту оновлення раз на рік, роботи відвідуватимуть сайт частіше.

Якщо вказати частоту раз на годину, робот, швидше за все, не витратить свій час на індексацію таких сторінок. Аналогічна ситуація з датою оновлення - якщо в карті сайту буде багато сторінок без змін, але зі "свіжою" датою, роботи не відвідуватимуть ці сторінки. Директиви, зазначені в карті сайту, є рекомендаціями, а не правилами для пошукових роботів.

Карта сайту html створюється для зручності користувачів, тому її оновлення необхідне щоразу, коли ви вносите зміни до структури сайту

(додаєте нові важливі для відвідувача сторінки сайту, видаляєте їх або переміщуєте до іншого розділу).

Створену карту сайту потрібно завантажити до кореневого каталогу сайту. Пошукові роботи цілком здатні самостійно знайти карту, але також є способи вказати посилання на сайт, щоб робот знайшов її раніше. Насамперед це вказівка посилання на карту у файлі robots.txt. Директива має такий вигляд:

Sitemap: <http://mysite.com/sitemap.xml>

Для перевірки правильності роботи карти сайту створено спеціальні програми, які працюють в онлайн-режимі. Вони здійснюють автоматичну перевірку і надають ряд додаткових переваг. Наприклад, вказують на наявність можливих помилок з боку ресурсу. Перевагою є те, що ці програми безкоштовні і працюють практично на всіх платформах.

Щоб здійснити перевірку, необхідно відкрити сторінку сервісу, завантажити потрібний файл у форматі html або ввести у поле відповідний URL сайту. За наявності помилок на сайті програма своєчасно повідомить, які з них викликають збої, і підкаже шляхи їх вирішення.

Для перевірки карти сайту було використано онлайн-сервіс websiteplanet.com (рис. 4.2).

Аналізатор карти сайту XML
Перевірте ваш файл sitemap і отримайте новий і без помилок

Введіть адресу карти сайту XML тут:

<https://www.yourwebsite.com/sitemap.xml> **ПЕРЕВІРИТИ!**

1. Введіть URL-адресу sitemap.xml у текстовому полі
2. Натисніть помаранчеву кнопку "Перевірити"!
3. Перевірте ваші помилки

Також працюватиме карта сайту Gunzip (.GZIP)

Ми перевіriamo, чи правильно працюють всі посилання, що містяться у вашому файлі sitemap

Рисунок 4.2 – Вебсайт перевірки карти сайту

Для перевірки потрібно вказати посилання на файл формату xml (рис. 4.3). Натиснути кнопку перевірки та отримати результат (рис. 4.4).

Рисунок 4.3 – Завантаження файлу

Рисунок 4.4 – Результат перевірки

Підсумовуючи, можна сказати, що карти сайтів є зручним інструментом, завдання якого полягає в спрощенні передачі не лише користувачам структури основних сторінок, але й пошуковим роботам. Без карт можна в принципі досягти індексації роботами, але з ними процес буде швидшим і правильно оформленим. Якщо ви не маєте досвіду роботи з картами сайтів, можна звернутися до професіоналів PROjectSEO, які правильно налаштують sitemap і допоможуть вашому сайту швидше вийти в топ пошукових систем.

4.3 Висновок

У четвертому розділі було розглянуто оптимізацію вебсистеми та надано рекомендації щодо її використання.

Для підвищення видимості вебсайту модельного агентства в пошукових системах було створено семантичне ядро, яке містить ключові слова та фрази, що відповідають тематиці сайту та потребам цільової аудиторії. Ці ключові слова були використані в мета-тегах, заголовках, альтернативних описах зображень та самому контенті сторінок. Також було оптимізовано внутрішню структуру сайту за допомогою правильного використання заголовків різних рівнів (h1-h4).

Для полегшення індексації сайту пошуковими роботами та покращення навігації для користувачів було створено карту сайту у форматах XML та HTML. Карта сайту дозволяє краулерам швидше знаходити та індексувати всі важливі сторінки, а також інформує про їх пріоритетність та частоту оновлення. Крім того, HTML-версія карти сайту є корисною для відвідувачів, оскільки надає зручний спосіб переходу до потрібних розділів.

Для перевірки правильності роботи карти сайту було використано онлайн-сервіс websiteplanet.com. Цей інструмент дозволяє завантажити XML-файл карти сайту та проаналізувати його на наявність помилок або попереджень. У випадку з розробленим сайтом модельного агентства перевірка пройшла успішно, що свідчить про коректність створеної карти.

Таким чином, оптимізація вебсистеми за допомогою створення семантичного ядра та карти сайту є важливим кроком для покращення видимості сайту в пошукових системах та зручності його використання відвідувачами. Дотримання наданих рекомендацій дозволить власникам сайту модельного агентства ефективно просувати свої послуги в Інтернеті та залучати нових клієнтів.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра було розроблено сучасний та функціональний вебсайт для модельного агентства з використанням актуальних вебтехнологій та з урахуванням специфіки діяльності у сфері модельного бізнесу.

У першому розділі було проведено ґрунтовний аналіз предметної області, що включав дослідження принципів проєктування сучасних вебсистем, вивчення актуальних вимог до вебсайтів в Україні та порівняльний аналіз існуючих сайтів модельних агентств. Це дозволило сформулювати чітке розуміння вимог та найкращих практик у сфері розробки сайтів для модельного бізнесу.

Другий розділ був присвячений безпосередньому проєктуванню вебсайту модельного агентства. Було розроблено структуру сайту, спроектовано базу даних для зберігання інформації про моделі та послуги агентства, створено привабливий та зручний дизайн інтерфейсу користувача. Також було обґрунтовано вибір методів та технологій для розробки сайту, що включають HTML, CSS, JavaScript, PHP, Laravel, MySQL тощо.

У третьому розділі було представлено практичну реалізацію спроектованого вебсайту. Розроблено програмну частину сайту з використанням обраного стеку технологій, реалізовано функціональні можливості, такі як фільтрація та сортування моделей, подача заявок на співпрацю тощо. Також створено зручний та естетично привабливий інтерфейс користувача, який забезпечує ефективну взаємодію відвідувачів сайту з модельним агентством.

Четвертий розділ присвячений оптимізації вебсистеми та наданню рекомендацій щодо її використання. Було створено семантичне ядро сайту для покращення його видимості у пошукових системах, розроблено карту сайту для полегшення індексації сторінок та зручності навігації для

користувачів. Також надано загальні рекомендації щодо подальшої підтримки та розвитку сайту.

Таким чином, поставлена мета кваліфікаційної роботи - розробка сучасного та функціонального вебсайту для модельного агентства - була успішно досягнута. Створений сайт відповідає актуальним вимогам та трендам у сфері веброзробки, враховує специфіку діяльності модельних агентств та забезпечує ефективну взаємодію з цільовою аудиторією. Впровадження розробленого сайту дозволить модельному агентству посилити свою присутність в онлайн-просторі, залучити нових клієнтів та моделей, а також підвищити ефективність своєї діяльності загалом.

Подальші напрямки розвитку проєкту можуть включати розширення функціональних можливостей сайту, інтеграцію з соціальними мережами та месенджерами, впровадження технологій віртуальної та доповненої реальності для презентації моделей, а також постійну оптимізацію та адаптацію сайту відповідно до змін у сфері модельного бізнесу та вебтехнологій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Duckett J. HTML and CSS: Design and Build Websites / J. Duckett. — Indianapolis, IN : John Wiley & Sons, 2021. — 490 p.
2. Frain B. Responsive Web Design with HTML5 and CSS3 / B. Frain. — Birmingham : Packt Publishing, 2020. — 408 p.
3. Freeman A. Pro Angular 9 / A. Freeman. — New York : Apress, 2020. — 778 p.
4. Grinberg M. Flask Web Development: Developing Web Applications with Python / M. Grinberg. — Sebastopol, CA : O'Reilly Media, 2022. — 316 p.
5. McFarland D. JavaScript & jQuery: The Missing Manual / D. S. McFarland. — Sebastopol, CA : O'Reilly Media, 2021. — 686 p.
6. Ergo-Models [Електронний ресурс] URL: <https://egomodels.com.ua/> (дата звернення: 29.05.2024).
7. Bravo [Електронний ресурс] URL: <https://bravo-models.com.ua/ua> (дата звернення: 29.05.2024).
8. Veа агентство [Електронний ресурс] URL: <https://www.veamodels.com/uk> (дата звернення: 29.05.2024).
9. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 / R. Nixon. — Sebastopol, CA : O'Reilly Media, 2022. — 832 p.
10. Powers S. Learn to Code HTML and CSS: Develop and Style Websites / S. Powers. — New Riders, 2019. — 384 p.
11. Resig J. Pro JavaScript Techniques / J. Resig. — New York : Apress, 2023. — 376 p.
12. Shklar L. Web Application Architecture: Principles, Protocols and Practices / L. Shklar, R. Rosen. — Chichester : John Wiley & Sons, 2022. — 440 p.
13. Spurlock J. Bootstrap: Responsive Web Development / J. Spurlock. — Sebastopol, CA : O'Reilly Media, 2021. — 128 p.

ДОДАТОК А
Текст програми

```

export default ModelForm;

import React, { useState } from 'react';
import axios from 'axios';

const ModelForm = () => {
  const [formData, setFormData] = useState({
    name: "",
    lastName: "",
    email: "",
    city: "",
    country: "",
  });

  const handleChange = (e) => {
    setFormData({ ...formData, [e.target.name]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    // Basic validation
    if (!formData.name || !formData.lastName || !formData.email ||
!formData.city || !formData.country) {
      alert('Please fill all the required fields');
      return;
    }

    try {
      const response = await axios.post('/api/participants', formData);
      console.log(response.data);
    } catch (error) {
      console.error('There was an error submitting the form!', error);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="text" name="name" value={formData.name}
onChange={handleChange} placeholder="Name" required />
      <input type="text" name="lastName" value={formData.lastName}
onChange={handleChange} placeholder="Last Name" required />
      <input type="email" name="email" value={formData.email}
onChange={handleChange} placeholder="Email" required />
      <input type="text" name="city" value={formData.city}
onChange={handleChange} placeholder="City" required />

```

```

        <input type="text" name="country" value={formData.country}
onChange={handleChange} placeholder="Country" required />
        { /* */ }
        <button type="submit">Submit</button>
    </form>
  );
};

export default ModelForm;

const express = require('express');
const bodyParser = require('body-parser');
const { Sequelize, DataTypes } = require('sequelize');

const app = express();
const port = 3000;

app.use(bodyParser.json());

// Database setup
const sequelize = new Sequelize('database', 'username', 'password', {
  host: 'localhost',
  dialect: 'postgres',
});

const Participant = sequelize.define('Participant', {
  partId: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  uniqId: {
    type: DataTypes.INTEGER,
    allowNull: false,
  },
  townId: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
  partName: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  partBio: {
    type: DataTypes.STRING,

```

```

    allowNull: true,
  },
  partStart: {
    type: DataTypes.DATE,
    allowNull: true,
  },
  score: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
});

```

```

app.post('/api/participants', async (req, res) => {
  try {
    const participant = await Participant.create(req.body);
    res.status(201).json(participant);
  } catch (error) {
    res.status(400).json({ error: error.message });
  }
});

```

```

sequelize.sync().then(() => {
  app.listen(port, () => {
    console.log(`Server is running on http://localhost:${port}`);
  });
});

```

```

const express = require('express');
const bodyParser = require('body-parser');
const { Sequelize, DataTypes } = require('sequelize');

```

```

const app = express();
const port = 3000;

```

```

app.use(bodyParser.json());

```

```

// Database setup
const sequelize = new Sequelize('database', 'username', 'password', {
  host: 'localhost',
  dialect: 'postgres',
});

```

```

// Define models
const Participant = sequelize.define('Participant', {
  partId: {

```

```

    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  uniqId: {
    type: DataTypes.INTEGER,
    allowNull: false,
  },
  townId: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
  partName: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  partBio: {
    type: DataTypes.STRING,
    allowNull: true,
  },
  partStart: {
    type: DataTypes.DATE,
    allowNull: true,
  },
  score: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
});

```

```

const ParticipantInfo = sequelize.define('ParticipantInfo', {
  uniqId: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    allowNull: false,
  },
  firstName: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  lastName: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  phone: {

```

```

    type: DataTypes.STRING,
    allowNull: false,
  },
  email: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  height: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
  bust: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
  waist: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
  hips: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
  dateOfBirth: {
    type: DataTypes.DATE,
    allowNull: true,
  }
});

```

```

const Portfolio = sequelize.define('Portfolio', {
  uniqId: {
    type: DataTypes.INTEGER,
    allowNull: false,
  },
  photoSetName: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  photo: {
    type: DataTypes.TEXT,
    allowNull: false,
  },
  details: {
    type: DataTypes.TEXT,
    allowNull: false,
  }
});

```

```

    }
  });

// API endpoints
app.post('/api/participants', async (req, res) => {
  const t = await sequelize.transaction();
  try {
    const participant = await Participant.create({
      uniqId: req.body.uniqId,
      partName: req.body.firstName,
      // additional participant fields
    }, { transaction: t });

    const participantInfo = await ParticipantInfo.create({
      uniqId: req.body.uniqId,
      firstName: req.body.firstName,
      lastName: req.body.lastName,
      phone: req.body.phone,
      email: req.body.email,
      height: req.body.height,
      bust: req.body.bust,
      waist: req.body.waist,
      hips: req.body.hips,
      dateOfBirth: req.body.dateOfBirth,
    }, { transaction: t });

    await t.commit();
    res.status(201).json({ participant, participantInfo });
  } catch (error) {
    await t.rollback();
    res.status(400).json({ error: error.message });
  }
});

sequelize.sync().then(() => {
  app.listen(port, () => {
    console.log(`Server is running on http://localhost:${port}`);
  });
});

```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Розробка вебсайту модельного агентства.

Виконав студент КНТ-110 Владислав ПІКАЛОВ

Керівник проєкту, к.т.н., доцент Тетяна ГОЛУБ

Рисунок Б.1 – Слайд 1

Мета та задачі

Метою цієї роботи є розробка сучасного та функціонального вебсайту для модельного агентства, який дозволить підвищити ефективність діяльності компанії, розширити її присутність в онлайн-просторі та покращити взаємодію з цільовою аудиторією.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- проаналізувати специфіку діяльності модельних агентств та їх потреби в онлайн-присутності;
- вивчити сучасні тенденції в вебдизайні та технологіях розробки сайтів;
- визначити функціональні та нефункціональні вимоги до вебсайту модельного агентства;
- розробити структуру та дизайн вебсайту відповідно до вимог;
- реалізувати вебсайт з використанням обраного стеку технологій;
- протестувати вебсайт та забезпечити його розгортання на хостингу.

2

Рисунок Б.2 – Слайд 2

Сучасні вебсайти

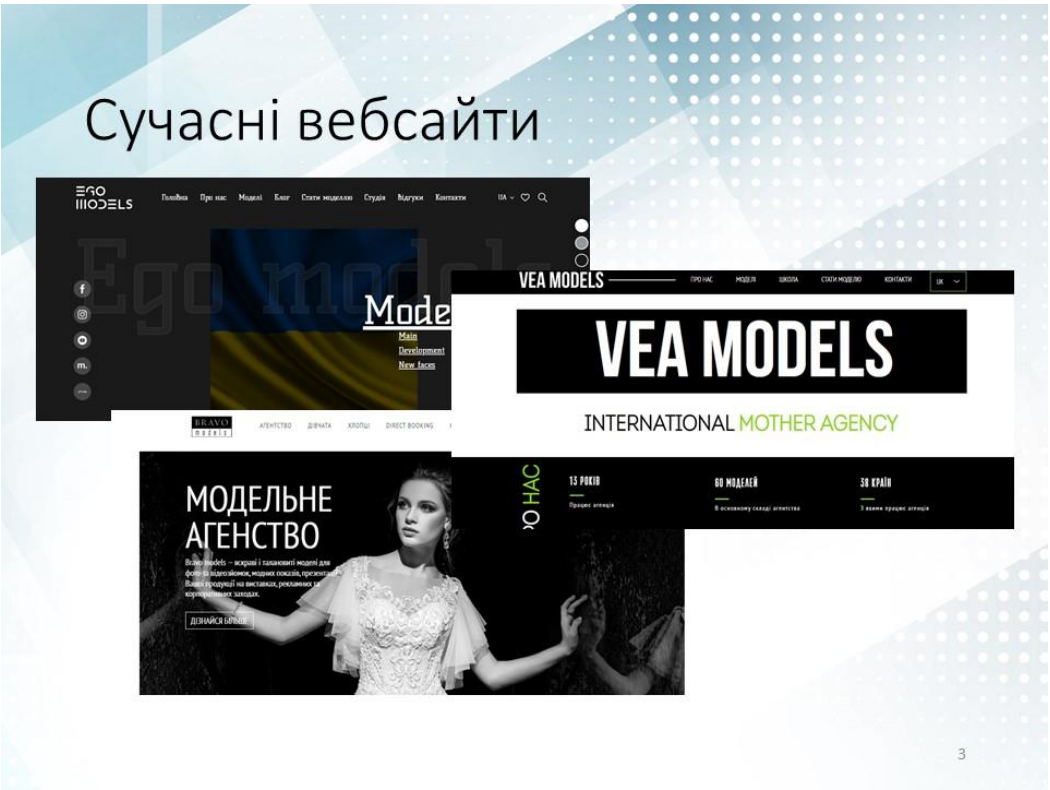


Рисунок Б.3 – Слайд 3

Порівняння вебсайтів

Критерії	Ego	Bravo	VEA
Дизайн	Стильний та сучасний	Привабливий та яскравий	Лаконічний та стильний
Адаптивність	Адаптивний під різні пристрої	-	-
Швидкість завантаження	-	Швидка	Деякі сторінки довго завантажуються
Пошук моделей	Зручний пошук за параметрами	-	Зручний поділ за категоріями
Портфоліо моделей	Детальні з якісними фото	Наявність відео-портфоліо	Не всі містять достатньо інформації
Онлайн-подача заявки для моделей	Можливість онлайн-подачі	Відсутня	-
Інформація про послуги	-	Детальна інформація	-
Блог/новини	Відсутній	-	Наявний з корисними статтями
Розділ "Про нас"	Не надто інформативний	-	-
Соціальні мережі	-	Посилання на соцмережі	-
Додаткові можливості	-	-	Онлайн-бронювання моделей, мультимовність

Рисунок Б.4 – Слайд 4

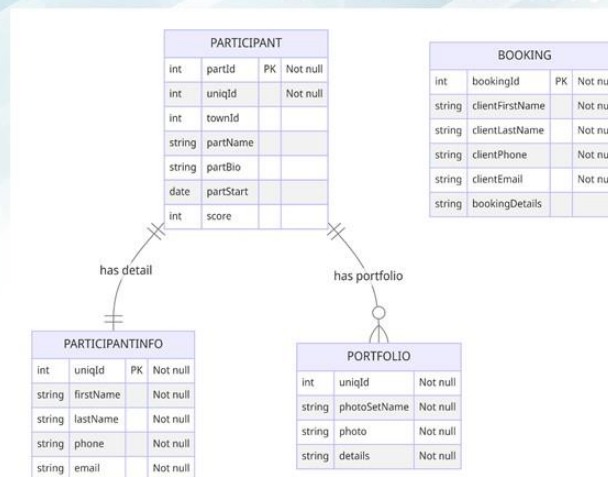
Структура вебсайту



5

Рисунок Б.5 – Слайд 5

Модель бази даних



6

Рисунок Б.6 – Слайд 6

Розробка дизайну вебсайту

Меню	Меню
Мотиваційна інформація	Послуги
Послуги	Моделі
Корисні посилання	Корисні посилання

7

Рисунок Б.7 – Слайд 7

Стек технологій

Область застосування	Обрані технології та інструменти	Обґрунтування вибору
Front-end	- HTML5 - CSS3 - JavaScript - Фреймворк (React, Angular, Vue.js)	- Семантична розмітка та структурування контенту - Візуальне оформлення та адаптивний дизайн - Інтерактивність та динамічність сторінок - Прискорення та спрощення розробки
Back-end	- PHP	- Простота, гнучкість, популярність - Архітектура MVC, вбудовані функції (ORM, шаблонізатор, маршрутизація)
База даних	- MySQL	- Безкоштовність, надійність, швидкість - Зрозумілий синтаксис SQL, ефективні механізми індексації та кешування
CSS-фреймворк	- Bootstrap	- Набір готових адаптивних компонентів - Кросбраузерність та прискорення верстки
Збірник проектів	- Webpack	- Автоматизація рутинних задач розробки – Гнучкий конфігураційний файл та плагіни
Контроль версій	- Git	- Відслідковування змін, робота з гілками - Хостинг репозиторіїв та додаткові функції (issue-трекер, CI/CD)
Якість коду	- ESLint - PHP_CodeSniffer - PHPUnit, Jest	- Статичний аналіз та виявлення помилок - Перевірка на відповідність стандартам кодування - Модульне та інтеграційне тестування
Розгортання	- Docker	- Стандартизація середовища розробки та розгортання - Легкість використання та сумісність з CI/CD процесами

8

Рисунок Б.8 – Слайд 8

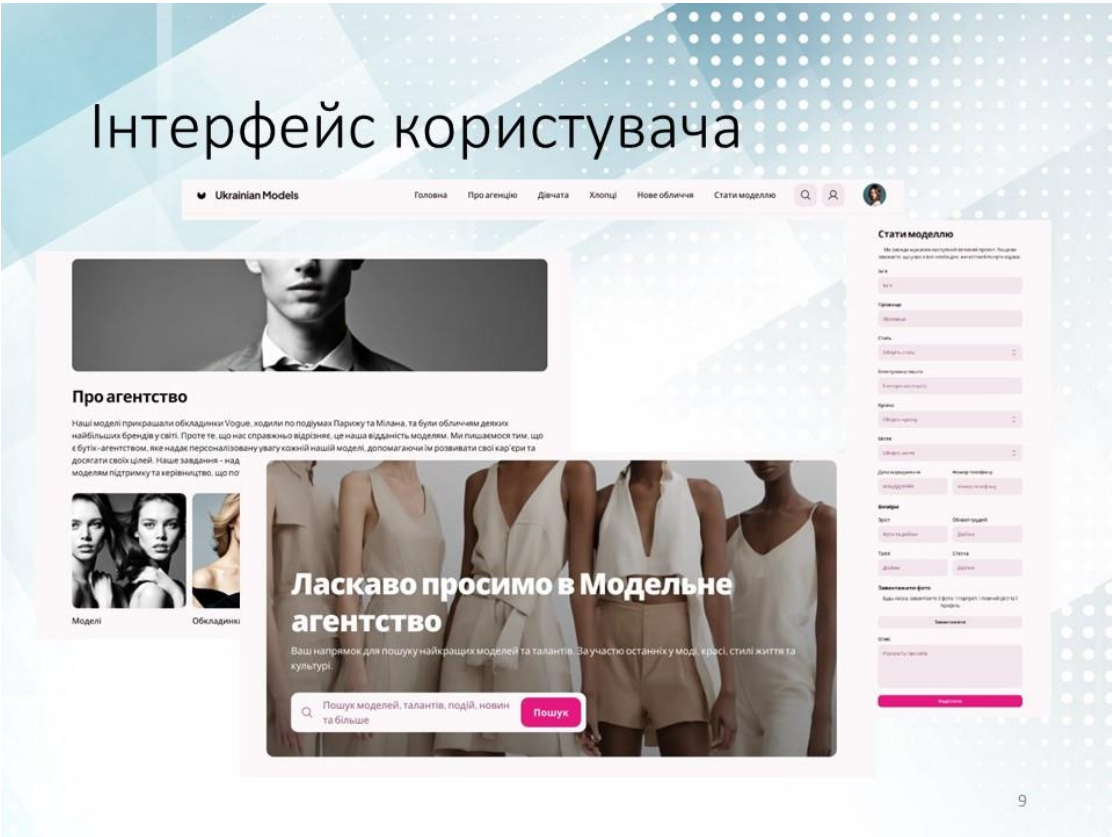


Рисунок Б.9 – Слайд 9

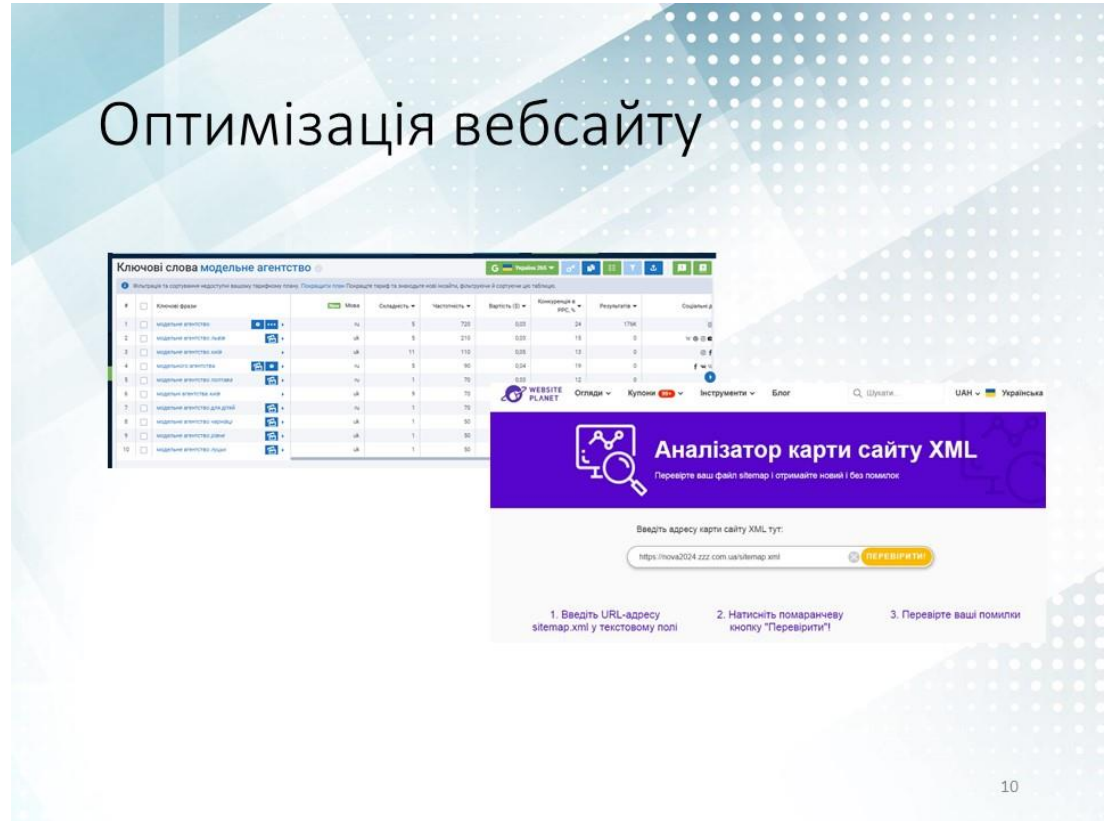


Рисунок Б.10 – Слайд 10

Висновки

У кваліфікаційній роботі бакалавра було успішно розроблено функціональний та сучасний вебсайт для модельного агентства. Робота включала аналіз предметної області, проєктування структури та дизайну сайту, практичну реалізацію з використанням актуальних вебтехнологій, а також оптимізацію та надання рекомендацій щодо використання сайту. Результатом є привабливий та зручний вебсайт, який ефективно представляє модельне агентство в мережі Інтернет та забезпечує взаємодію з клієнтами та моделями.

11

Рисунок Б.11 – Слайд 11