Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка системи автоматизації обліку дисконтних карток ТОВ «Комфорт Кард» м. Київ»

Студента 2 курсу, 5м групи, спеціальності 121 «Інженерія програмного забезпечення», спеціалізації «Інженерія програмного забезпечення» _____ підпис студента — Гулаков Владислав Васильович

Науковий керівник кандидат технічних наук, доцент _____ підпис керівника — Рзаєва Світлана Леонідівна

Гарант освітньої програми доктор технічних наук, професор _____ підпис керівника — Криворучко Олена Володимирівна

КИЇВ – 2019

АНОТАЦІЯ

Гулаков В.В. Розробка програмного забезпечення системи автоматизації обліку дисконтних карток

У випускній кваліфікаційній роботі розглянуто розробку систему автоматизації обліку дисконтних карток та її компонентів. Велика увага приділена зручності та функціональності системи. Застосовано як серверна так і клієнтська частина JavaScript, а також сучасні систем управління базами даних для підвищення ефективності та швидкості роботі системи.

Ключові слова: Розробка програмного забезпечення, розробка систем автоматизації, JavaScript, система управління базами даних.

ANNOTATION

Hulakov V.V. Development of discount card accounting automation software

The final qualification work examine the development of a system of accounting of discount cards and its components is considered. Much attention is paid to the convenience and functionality of the system. Both server-side and client-side JavaScript and advanced database management systems are used to improve system efficiency and speed.

Keywords: Software development, development of automation systems, database management system.

ТЕХНІЧНЕ ЗАВДАННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ОБЛІКУ ДИСКОНТНИХ КАРТОК

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Розробка системи автоматизації обліку дисконтних карток.

Галузь застосування: Інформаційні технології, електронна комерція, комерційна діяльність.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є замовлення від української компанії ТОВ «Комфорт Кард».

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Дана робота присвячена дослідженню автоматизації обліку дисконтних карток в сферах електронної комерції та комерційної діяльності. В роботі досліджується можливість створення оптимального програмного забезпечення для швидкої та зручної роботи при використанні дисконтних карток.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Для вибору оптимальних технологій при розробці програмного забезпечення, необхідно розробити вимоги до кінцевого результату розробки з деталями які можуть вплинути на функціонал програмного забезпечення. На думку автора, це найважливіші аспекти:

- коректне відображення – оптимізація відображення системи на різних екранах електронних пристроїв.
- використання клієнт-серверної моделі – прискорення роботи програмного забезпечення методом розподілення роботи системи між сервером та клієнтом;

- кроссбраузерність системи - можливість запускати додаток на широкому спектрі мобільних і стаціонарних електронних пристроїв;

5. ВИМОГИ ДО ПРОГРАНМОГО ЗАБЕЗПЕЧЕННЯ

Технології що будуть застосовуватися при побудові:

- Мова програмування JavaScript;
- Платформа Node.js;
- Текстовий редактор PHP Storm;
- Технологія хостингу у глобальній мережі та локальний сервер.

6. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання роботи повинна бути розроблена наступна документація:

1) Технічне завдання;
2) Програма та методика користувача;
3) Керівництво користувача.

7. ЕТАПИ ПРОЕКТУВАННЯ

1) Дослідження існуючих систем обліку пластикових карток;
2) Розробка та узгодження технічного завдання;
3) Розробка архітектури БД;
4) Розробка електронного макету;
5) Програмна реалізація системи автоматизації обліку дисконтних карток;
6) Тестування системи;
7) Розробка програми, методики та керівництва користувача;

Київський національний торговельно-економічний університет

Факультет ФОАІС Кафедра «Програмної інженерії та кібербезпеки»

Спеціальність 121 «Інженерія програмного забезпечення»

Спеціалізація «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри

_____Криворучко О.В.

«___» _____2019 р.

Завдання

на випускну кваліфікаційну роботу студентові

Гулакову Владиславу Васильовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Розробка ситеми автоматизація обліку дисконтних карток ТОВ «Комфорт Кард» м. Київ»

Затверджена наказом ректора від «07» *листопада* 2018 р. № 4183

2. Строк здачі студентом закінченої роботи (проекту)_____

3. Цільова установка та вихідні дані до роботи (проекту)

Мета роботи – полягає в розробці системи автоматизації для підвищення ефективності роботи при використанні пластикових карток за допомогою програмного забезпечення і створенні програмних засобів на їх основі для використання в роботі з клієнтами системи.

Об'єкт дослідження - розробка програмного забезпечення системи автоматизації.

Предмет дослідження – розробка системи автоматизації обліку дисконтних карток

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА WEB-РОЗРОБКИ

1.1. Введення в предметну область
1.2. Автоматизація виробництва
1.3. Поняття веб-сайта
1.4. Технології відображення ресурсу
1.5. Аналіз існуючих методів обліку карток
1.6. Висновки до розділу

РОЗДІЛ 2. АРХІТЕКТУРА І МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Робота з макетами програмного забезпечення
2.2. Програмне середовище розробки
2.3. Проектування бази даних
2.4. Редактор коду PHP Storm
2.5. FTP-менеджер FileZilla
2.6. Висновки до розділу 2

РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ ОБЛІКУ

3.1. Вимоги до програмної реалізації
3.2. Архітектура системи
3.3. Адаптивний інтерфейс
3.4. Функціональні можливості
3.5. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	*Вибір теми випускної кваліфікаційної роботи*	*25.09.2018*	
2.	*Розробка та затвердження завдання на проект магістра (Наказ №185 від 07.11.18)*	*07.11.2018*	
3.	*Вступ та перелік літературних джерел*	*28.02.2019*	
4.	*Розробка технічного завдання*	*22.03.2019*	
5.	*Розділ 1. Сучасні методи побудови і розпізнавання опису маршрутизації шляхів*	*18.04.2019*	
6.	*Розділ 2. Запропоновані методи навігаційно комп'ютерних систем*	*24.05.2019*	
7.	*Розділ 3. Розробка та впровадження НККС*	*21.06.2019*	
8.	*Розробка програми та методики тестування*	*18.10.2019*	
9.	*Написання наукової статті*	*27.09.2019*	
10.	*Керівництво користувача*	*23.10.2019*	
11.	*Висновки та пропозиції*	*01.11.2019*	
12.	*Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)*	*13.11.2019*	
13.	*Підготовка автореферату та презентації доповіді*	*14.11.2019*	
14.	*Попередній захист випускної кваліфікаційної роботи*	*14.11.2019 – 15.11.2019*	
15.	*Здача зброшурованої випускної кваліфікаційної роботи*	*25.11.2019*	
16.	*Зовнішнє рецензування випускної кваліфікаційної роботи*	*26.11.2019*	
17.	*Підготовка до публічного захисту випускної кваліфікаційної роботи*	*02.12.2019-04.12.2019*	

7. Дата видачі завдання **«26» вересня 2018 р.**

8. Науковий керівник випускної кваліфікаційної роботи

_____

(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент_____

(прізвище, ініціали, підпис)

11. Відгук керівника випускної кваліфікаційної роботи

____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____

Науковий керівник випускної кваліфікаційної роботи___________________
(підпис, дата)

Відмітка про попередній захист ____
(ПІБ, підпис, дата)

12. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента ______________________________
(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми ____
(прізвище, ініціали, підпис)

Завідувач кафедри ____
(підпис, прізвище, ініціали)

KNUTE КНТЕУ

ЗМІСТ

ВСТУП 3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА WEB-РОЗРОБКИ 7
1.1. Введення в предметну область 7
1.2. Автоматизація виробництва 9
1.3. Поняття веб-сайта 11
1.4. Технології відображення ресурсу 13
1.5. Аналіз існуючих методів обліку карток 15
1.6. Висновки до розділу 1 19
РОЗДІЛ 2. АРХІТЕКТУРА І МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 21
2.1. Робота з макетами програмного забезпечення 21
2.2. Програмне середовище розробки 24
2.3. Проектування бази даних 31
2.4. Редактор коду PHP Storm 35
2.5. FTP-менеджер FileZilla 37
2.6. Висновки до розділу 2 39
РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ ОБЛІКУ .. 41
3.1. Вимоги до програмної реалізації 41
3.2. Архітектура системи 43
3.3. Адаптивний інтерфейс 45
3.4. Функціональні можливості 47
3.5. Висновки до розділу 3 52
ВИСНОВКИ ТА ПРОПОЗИЦІЇ 53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 54
ДОДАТКИ 56

Зм.	Аркуш	№ докум.	Підпис	Дата	КНТЕУ 121– 05-10.МР		
Зав. каф.		Криворучко О.В.			Розробка ситеми автоматизації обліку дисконтних карток ТОВ «Комфорт Кард»	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.				Зміст	2	55
Гарант		Криворучко О.В			ЗМІСТ	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розроб		Гулаков В.В.						

ВСТУП

Актуальність роботи полягає у підвищенні ефективності обліку та інформаційних технологій, вдосконалення статистичного аналізу, впровадження технологій для отримання інформації в режимі реального часу. Впровадження системи інтеграції нових партнерів у систему автоматизованого обліку.

Більшість компаній намагаються залучити клієнтів різними бонусами та заохоченнями. Хтось влаштовує грандіозні розпродажі косметики або одягу минулих колекцій, а деякі магазини впроваджують систему лояльності клієнтів. Її частиною є дисконтні картки, що дозволяють отримати покупцеві знижку при купівлі товару на певну суму. Далі ми розглянемо плюси і мінуси такої системи.

Збільшення прибутку підприємства стоїть на першому місці для будь-якого бізнесу.

Першим завданням цього інструменту є розширення цільової аудиторії за рахунок вигідних пропозицій для покупців. Деякі підприємства замовляють масове виробництво карток напередодні різноманітних локальних акцій та розпродажів. Проведений захід залишає гарне враження у покупців. Впровадження знижкових карт в бізнес необхідно для вирішення наступних завдань:

- Встановлення зворотного зв’язку з клієнтами. Якщо замовити виробництво карт з ідентифікаційним штрих кодом або магнітною смугою, можна отримати персональну інформацію покупця.

					КНТЕУ 121– 05-10.МР		
Зм.	*Аркуш*	*№ докум.*	*Підпис*	*Дата*	*Розробка ситеми автоматизації обліку дисконтних карток ТОВ «Комфорт Кард»*	*Стадія*	*Аркуш*	*Аркушів*
Зав. каф.		Криворучко О.В.				В	*3*	*55*
Керівник		Рзаєва С.Л.						
Гарант		Криворучко О.В						
Розроб		Гулаков В.В.			*ВСТУП*	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

- Аналізу потреб клієнта, щоб розробляти спеціально для нього програми індивідуальних заохочень.
- Підвищення ефективності рекламних кампаній. Різноманітні відеоролики і листівки ніколи не зможуть підняти попит і лояльність до бренду так, як можливість отримати знижку при наступній покупці продукції.

Головний плюс, які виділяють власники дисконтних карток, полягає у можливості купівлі товарів і послуг за зниженою ціною завдяки використанню знижки. Особливо це вигідно клієнтам, здійснюють покупки на великі грошові суми або постійно оновлювати технічні пристрої. Список інших плюсів знижкових карток для покупців виглядає наступним чином:

- можливість накопичувати знижки і бонуси;
- шанс отримати приз в результаті акції;
- можливість накопичити бали для купівлі товару з істотною знижкою або отримання речі безкоштовно;
- отримання інформації про розпродажі, розіграші призів та інших акціях компанії.

Все розпочинається з опрацювання стратегії впровадження торгового інструменту у вибраний бізнес. Після завершення цього етапу приступають до розробки макету картки. Потрібно не тільки продумати дизайн, але і технічні особливості реалізації програми на об’єкта, в торгових точках. Кожній картці присвоюється номер, який потім вноситься в систему компанії. Після цього готовий макет посилають в друкарню, обрану для друку.

Суть даного інструменту зводиться до того, щоб заохочувати споживача при покупці товару. Багато хто думає, що дисконтні картки бувають тільки пластиковими, але це не так. Деякі з них випускаються у

вигляді купона, з допомогою якого можна заощадити, купивши будь-який товар. Карта знижок має свій термін дії. Коли він закінчиться, скористатися бонусами не вийде.Впровадження системи дисконтних карток сприяє підвищенню прибутку підприємства, але має і декілька недоліків:

- клієнти можуть обслуговуватися лише в окремих організаціях;
- додаткові витрати якщо власник втратив картку;
- безпечність даних.

На сьогоднішній день, на ринку України конкурентність для розробок з використанням дисконтних карток низька, *оскільки недоліком конкурентів* є те, що вони пропонують виготовлення, друк та налаштування карток, але залишається проблема обліку.

Мета дослідження полягає в розробці системи автоматизації обліку для підвищення ефективності та швидкості роботи при використанні дисконтних карток.

Об'єктом дослідження є діяльність ТОВ «Комфорт Кард».

Предметом дослідження випускної кваліфікаційної роботи є розробка системи автоматизації обліку дисконтних карт.

Для досягнення поставленої мети необхідно послідовно вирішити такі **завдання**:

- Дослідити предметну область;
- Дослідити і проаналізувати існуючі системи обліку дисконтних карток;
- Спроектувати та розробити систему;
- Протестувати програмне забезпечення.

Наукова новизна дослідження полягає у створенні оптимального програмного забезпечення, швидкого та зручного інструменту для підвищення ефективності роботи маркетингового відділу.

Клієнт може без застосування великих людських ресурсів, зареєструватися та отримати дисконтну карту і використовувати її в закладах-партнерах, відслідковувати свій баланс в реальному часі, дізнаватися про акції та отримувати бонуси та кешбек. Підприємство отримує змогу скоротити число робітників і задіяти більше ресурсів для розвитку та залучення нових партнерів до мережі дисконтних карт.

Практична цінність. Розроблена система буде використана на підприємстві для автоматизації обліку дисконтних карток, впровадження системи бонусів для клієнтів, а також для підвищення ефективності та швидкості роботи маркетингово відділу, за рахунок отримання результатів в режимі реального часу.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА WEB-РОЗРОБКИ

1.1. Введення в предметну область

Карткова система - технічний засіб для автоматизації маркетингу. Мета цієї автоматизації - скорочення транзакцій-ційних витрат - витрат на реалізацію товарів і послуг. Однак впровадження засобів автоматизації також пов'язане з витратами. Найбільші з них припадають на облік угод. Велику економію дають кошти, що забезпечують доступ до всіх реквізитів товару, покупця, продавця, торгової точки, часу і гаранта угоди безпосередньо при її здійсненні. Для цього, власне, і використовуються карти. Карта (або кілька карт), що бере участь в угоді, є або безпосереднім сховищем необхідних реквізитів (так звана offline авторизація), або ключем доступу до баз реквізитів (online авторизація). Тому ми визначаємо сутність карткових технологій як інструменту, що забезпечує підтримку класифікації предметів і суб'єктів угод, максимальне наближення усіх видів реєстрації господарських операцій до місць їх виникнення і автоматичне формування бухгалтерських проводок та інших розрахункових (облікових) документів. Але крім інструменту здійснення угод, карткова система може виконувати і інші ролі спрямовані на вдосконалення процесів ціноутворення на товари і

послуги, що реалізуються підприємством, залучення та закріплення клієнтів, які оплачують покупки готівкою або їх. Всі перераховані ролі типові для впровадження так званих дисконтних карт (або карт лояльності).

Цілі впровадження дисконтних карт:

1. Залучення, закріплення і розширення безлічі клієнтів, які використовують картку. Перше вирішується, по більшій частині, наданням знижок на повторні або регулярно вироблені покупки. Друге - випуском в оборот (емісією) персоніфікованих (приписаних до конкретному власнику) дисконтних карт і урахуванням покупок клієнта з накопиченням всякого роду очок, електронних ваучерів, бонусів та ін.
2. Приховати справжні ціни на товар, що реалізуються підприємством.
3. Реклама. Логотипи дисконтних карт можна (і потрібно!) Використовувати для додаткової реклами продукції підприємства, створення іміджу власника або емітента. Тому, на відміну від розрахункових, дисконтні карти роблять більш яскравими. Якщо впроваджується мультикарта, на ній потрібно розміщувати логотипи найбільш значних учасників, які візьмуть на себе витрати з виготовлення таких карт.
4. Додатковий прибуток. Впровадження системи дисконтних карт пов'язано з витратами. Велика їх частина припадає на виготовлення власне самих карт. Розцінки тут сильно варіюються в залежності від величини партії (чим більше - тим дешевше).

Карткова система - технічний засіб для автоматизації маркетингу. Мета цієї автоматизації - скорочення транзакцій-ційних витрат - витрат на реалізацію товарів і послуг.

1.2. Автоматизація виробництва

Основою будь-якого виробництва є технологічний процес — певна взаємодія знарядь і предметів праці, обслуговуючої і транспортної систем, що утворюють складну багаторівневу виробничу систему, у підсумку чого випускається продукція, яка відповідає критеріям якості у заданих обсягах.

Переміщення предметів праці від однієї стадії обробки до іншої можна визначити як матеріальний потік у виробничому просторі. Для забезпечення роботи виробничої системи, слід організувати ще інформаційний та енергетичний потоки. Виробничий процес можна представити як систему, котра перетворює потоки енергії, матеріалів і інформації.

З погляду автоматизації за характером матеріального потоку, технологічні процеси можна розділити на два типи безперервні і дискретні.

З погляду автоматизації за характером матеріального потоку, технологічні процеси можна розділити на два типи безперервні і дискретні.

Автоматизація безперервних виробництв

В безперервних технологічних процесах матеріальний потік та інформація, що його відображає, безперервні. Матеріальний потік, який проходить через технологічне обладнання і зазнає там в кожну мить часу зміни своїх властивостей, є неперервним.

До безперервних відносять виробництва, у яких вимагається регулювати витрату, тиск, температуру, напругу, переміщення рухомих елементів та інші величини в усьому діапазоні їх змін. Це різноманітні хімічні реактори, процеси приготування харчових продуктів, металургія, системи постачання теплом, водою та електроенергією.

У безперервних виробництвах зайнято мало людей, тому за рахунок автоматизації можна знизити витрати матеріалів та енергії або усталити технологічний процес, усунувши його залежність від суб'єктивних чинників. Для керування таким виробництвом, вимагається узгодження динамічних характеристик об'єкту керування і системи автоматичного регулювання в усьому інтервалі зміни регульованих величин.

Автоматизація дискретних виробництв

Для дискретних технологічних процесів характерна вихідна продукція у вигляді виробів, що обчислюються в штуках. Початкові складники перетворюються циклічно і готова продукція випускається партіями.

До дискретних відносять виробництва з кінцевим числом станів змінних, наприклад з увімкненням (відкриттям) і вимкненням (закриттям) клапанів, засувок, пускачів за сигналами двопозиційних давачів.

Автоматизація дискретного виробництва розвинена менше, внаслідок більшої різноманітності виробів та операцій, підвищених вимог до точності операцій. Тут зайнята значна кількість робітників ручної праці. Для дискретного виробництва характерне величезне число варіантів автоматизації, операцій, що відрізняються послідовністю, затратами і ефективністю. Їх зіставлення вимагає формального опису

алгоритмів керування устаткуванням і розробки моделей організації виробництва.

У реальних задачах частіше за все спостерігається поєднання обох видів виробництв, що називають безперервно-дискретним виробництвом. Крім того, методи автоматизації дискретного виробництва все частіше застосовують до автоматизації безперервного виробництва.

Різновиди автоматизації виробництва

За ступенем повноти розрізняють автоматизацію виробництва:

- часткову — передбачає автоматизацію основних виробничих процесів,
- комплексну — передбачає автоматизацію не лише процесу виробництва, але й процесів керування й обслуговування,
- повну — передбачає автоматизацію всіх основних і допоміжних процесів.

За принципом керування виробничі системи можна поділити на прості, складні та інтелектуальні. Розрізнити їх можна за видом алгоритму роботи і принципом прийняття в них рішень. В блок-схемах алгоритмів роботи простих систем використовуються тільки виконавчі блоки, а в блок-схемах алгоритмів роботи складних систем — також блоки прийняття рішень. Для інтелектуальних систем характерне існування дерева прийняття рішень.

1.3. Поняття веб-сайта

Сайт або веб-сайт (від англ. website, місце, майданчик в інтернеті) — сукупність веб-сторінок, доступних у мережі Інтернет, які об'єднані як за змістом, так і за навігацією під єдиним доменним ім'ям. Фізично сайт може розміщуватися як на одному, так і на кількох серверах.

Сайтом також називають вузол мережі Інтернет, комп'ютер, за яким закріплена унікальна IP-адреса, і взагалі будь-який об'єкт в Інтернеті, за яким закріплена адреса, що ідентифікує його в мережі (FTP-site, WWW-site тощо).

Набір зв'язаних між собою інформаційних онлайнових ресурсів, призначених для перегляду через комп'ютерну мережу за допомогою спеціальних програм — браузерів. Веб-вузол може бути набором документів в електронному вигляді, онлайн службою.

Internet-магазин є однією з найпоширеніших комерційних моделей електронної торгівлі у сфері B2C.

Internet-магазин (електронний, віртуальний, e-shop) являє собою спеціалізований Web-сайт, який належить фірмі-товаровиробнику, торговій фірмі тощо та призначений для просування споживчих товарів на ринку, збільшення обсягів продажу, залучення нових покупців.

Інше визначення Internet-магазину характеризує його як реалізоване в мережі Internet представництво шляхом створення Web-сервера для продажу товарів та пов'язаних з ними послуг користувачам Internet. Кількість представлених на сервері видів товарів може коливатися від кількох одиниць до кількох тисяч.

Характерними рисами Internet-магазинів є те, що вони можуть пропонувати значно більшу кількість товарів та послуг, ніж реальні магазини і забезпечувати споживачів значно більшим обсягом інформації, необхідної для прийняття рішення про покупку. Також завдяки використанню Internet-технологій є можливою персоналізація підходу до споживачів з врахуванням попередніх відвідувань магазину та зроблених в ньому покупок та використання Internet-магазину як ефективного способу маркетингових досліджень (анкетування, конференції покупців і т.п.).

Internet-магазини потребують значно менших витрат на утримання та організацію роботи, оскільки у ньому значно обмеженіша матеріально-технічна база (будівлі, споруди, приміщення) та кількість обслуговуючого персоналу.

1.4. Технології відображення ресурсу

Статичні сайти - складаються з Html файлів і передаються користувачеві у вигляді, в якому зберігаються на сервері

Динамічні сайти - як правило, працюють на основі системи управління сайтом (CMS). Вміст генерується, за допомогою різних скриптів використовуючи базу даних

Сайт візитка - представництво компанії в мережі Інтернет. Надає загальну інформацію про власника (компанія, підприємець і т.д.). Як правило, складається з 5-7 сторінок, при цьому максимально описує вид діяльності. Головні розділи - про компанію, прайс-листи, контактні дані, схема проїзду.

Бізнес сайт - це каталог продукції компанії (товари або послуги). У каталозі присутній докладний опис товару, публікуються сертифікати й нагороди, відгуки клієнтів і звичайно технічні дані. Сайт даної специфікації незамінний для будь-якої компанії.

Промо сайт - веб сайт конкретної торгової марки (бренду). Створюється з метою реклами нового продукту або бренду в цілому. Розділи: акції, вікторини, голосування, ігри і тд.

Інтернет-магазин - це каталог товарів з можливістю онлайн замовлення. Клієнт має можливість оплати товару, за допомогою різних сервісів використовуючи при цьому пластикову карту, рахунок або Інтернет гроші. Розділи: про магазин, допомога, каталог, кошик, сторінка замовлення, контакти.

Корпоративний сайт - містить повну інформацію про компанію (послуги, товари і тд.). Як правило, має складну структуру і має розширену функціональність. Може бути відкритий для відвідувача або призначений для внутрішнього використання (зв'язок, документообіг і тд .).

Розробка дизайну

На даному етапі розробляється концепція надання інформації користувачеві. Дизайнер розробляє макети сторінок сайту (головна, типові). Сучасні сайти складаються з безлічі графічних елементів, тому веб дизайнер продумує всі деталі макета, створює структуру розділів, навігації і тд. Далі структура обговорюється співробітниками студії (верстальники, програмісти, сеошники). Після затвердження структура подається роботодавцю, якщо замовник задоволений, починається отрисовка макетів, як правило, декілька варіантів дизайну. Опрацьовуються всі деталі кнопки, форми, списки, заголовки і тд. Кінцевий результат надається замовнику. Після затвердження макети передаються верстальщикам веб студії.

Верстка

Верстальщик отримує макети, що складаються з шарів розроблені в графічному редакторі (Psd, Ai). Завдання верстальника - перетворення графічних макетів в веб сторінки. Складність даного етапу розробки - сумісність різних браузерів. Веб сторінка повинна однаково коректно відображатися у всіх популярних браузерах, що часто відбувається на оборот. Елементи сторінки в різних браузерах інтерпретуються по-різному, що призводить до не коректного відображення веб сторінок. На допомогу у цій ситуації приходять різні "хакі" придумані розробниками і впроваджувані безпосередньо в саму розмітку гіпертексту (html) або в

каскадні таблиці стилів (css). Після перевірки "кросбраузерності" верстки починається наступний етап - програмування.

Програмування

Завдання програміста об'єднати задумку дизайнерів з роботою верстальників. Фахівці веб студії створюють програмну частину ресурсу. Програмна основа сайту розробляється з нуля або за допомогою розробленої CMS. Враховуючи той факт, що останнім часом системи управління сайтом дуже популярні, як правило, всі сучасні сайти використовують CMS.

Виходячи з технічного завдання, веб студія визначає тип веб сайту . Зважує обсяг робіт і призначає ціну створення сайту.

1.5. Аналіз існуючих методів обліку карток

Дисконтна карта - ефективний маркетинговий інструмент. За нею покупці можуть отримати товар за зниженою вартістю. Для використання цього інструменту компанії потрібно спочатку створити карту. Надалі вона видається покупцю, після чого останній може здійснити покупку зі знижкою. Всі ці операції потрібно правильно враховувати.

Час видачі карти покупцеві принципово важливо, так як в момент передачі права на дисконт вносяться відповідні проводки. Час видачі залежить від політики конкретної компанії. Розглянемо різні варіанти:

- У процесі рекламної кампанії, потрібної для залучення все більшого кола споживачів
- У дату відкриття торгової точки.
- Після покупки в магазині на задану суму.
- Придбання продукції певної марки.

- Також право на дисконт може продаватися. В цьому випадку карта пропонується на касі магазину, а клієнт вирішує, чи купить він її.

Покупцеві надається знижка на підставі дисконтної карти. Враховувати знижку можна тільки після того, як вона фактично надано. Пов'язано це з тим, що до моменту покупки продукції невідомо, чи скористається клієнт запропонованими пільгами. Часто розмір знижки визначається сумою покупки. Якщо виручка від продажів була відображена без урахування знижки, але фактично пільга була надана, потрібно зробити корекцію способом «червоне сторно».

В платіжних документах потрібно вказувати суму з урахуванням пільги, а також підстава надання знижки. Якщо в паперах нічого не вказано, для цілей оподаткування податками приймається ринкова вартість виробів. Податкові органи мають право контролювати правильність здійснення операцій тоді, коли вартість товару зменшена більш ніж на 20%. Отже, 20% - це максимальна знижка, яку може дати споживачеві фірма. Одночасно з цим не можна завищувати роздрібні ціни, так як різниця в 20% визначається на підставі середніх ринкових цін.

При розрахунку ціни з метою оподаткування не визнаються такі пільги, виставленими з наступних причин:

- Тимчасова зміна попиту на товар (наприклад, сезонні зміни).
- Дефекти продукції.
- Закінчення строку придатності.
- Просування нових моделей на ринку.
- Продаж зразків для демонстрації їх властивостей споживачам.

Також можна відзначити такі особливості:

- Податкова база по рекламі не формується, так як дисконтні карти - це саме маркетингове, а не рекламний засіб. Реклама, згідно з нормативним визначенням, представляє собою інформацію про фірму і її продукції. Дисконт не є інформаційним носієм.
- При передачі дисконту споживачеві на безкоштовній основі зобов'язань щодо виконання функцій податкового агента не утворюється. Норма ця заснована на тому, що передача права на дисконт не рахується безоплатним актом. По суті, з точки зору юридичних норм цей процес можна вважати укладенням договору на покупку продукції по зменшеним цінами.
- Карта - це не товар. Це предмет, який встановлює факт виникнення цивільно-правових відносин.

Дисконтна карта - ефективний маркетинговий інструмент. За нею покупці можуть отримати товар за зниженою вартістю. Для використання цього інструменту компанії потрібно спочатку створити карту. Надалі вона видається покупцю, після чого останній може здійснити покупку зі знижкою. Всі ці операції потрібно правильно враховувати.

Отже знаючи систему роботи с дисконтними картками, можна виділити такий найпопулярніший інструмент для роботи з обліком дисконтних карт як програмне забезпечення російської компанії 1С, а також української компанії M.E.Doc.

Фірма «1С» — російська компанія, що спеціалізується на дистрибуції, підтримці і розробці комп'ютерних програм і баз даних ділового та домашнього призначення.

Фірма заснована в 1991 в Росії, має штат 1000 осіб. Засновник і директор фірми — Нуралієв Борис Георгійович. «1С» було спочатку

назвою власної пошукової програми: не більше 1С (секунди) потрібно для отримання необхідної інформації.

Фірма «1С» розробляє платформу для створення систем автоматизації підприємств. В різних країнах на основі цієї платформи розробляються прикладні рішення з урахуванням вимог місцевого законодавства і вимог окремих галузей промисловості.

Також фірма «1С» розробляє продукти для домашніх комп'ютерів та освітньої сфери, такі як: серія навчальних програм «1С: Репетитор», серія «1С: Школа» на платформі «1С: Освіта», ігри.

В Україні права на платформу викуплені компанією ДП «Єврософтпром» і відповідно створено повністю українську систему програм «1С: Підприємство», а найпопулярнішим програмним продуктом з найбільшою кількістю продаж зараз є Бухгалтерія для України.

«M.E.Doc» (My Electronic Document, також Medoc, Медок) — поширене українське програмне забезпечення для подання звітності до контролюючих органів та обміну юридично значущими первинними документами між контрагентами в електронному вигляді. У середині 2017 року програмою користувалася переважна більшість компаній в Україні

Програма M.E.Doc розроблена українськими програмістами під керівництвом Олесі Линник, яка в 2000 році продовжила сімейну справу свого батька Сергія Линника, що займався розробкою бухгалтерських систем з 1990 року і заснував групу компаній «Інтелект-Сервіс». 1993 року його розробка «БЕСТ ЗВІТ» була першою в Україні програмою для автоматизації роботи малого і середнього бизнесу. «БЕСТ ЗВІТ» стала попередником програми M.E.Doc.

Із розвитком нових технологій і вимог ринку програмне забезпечення безперервно актуалізувалося і вдосконалювалося, результатом чого в 2010 році стала поява на українському ринку розробки M.E.Doc.

У 2013 році в цій програмі об'єднали напрацювання з різних напрямків автоматизації бухгалтерської діяльності: звітності, розрахунку заробітної плати, взаємодії всередині корпорацій і обміну документами між контрагентами. Таким чином, M.E.Doc став багатофункціональним комплексом з максимальними можливостями для бухгалтерів в країні. Програмне забезпечення задовольняє потреби в роботі з електронними документами компаній будь-якого масштабу, форми власності та виду діяльності.

Розробки «Інтелект-Сервіс» дозволили перевести процес подачі звітності в Україні в електронну форму. Спочатку ця процедура відбувалася з використанням дискет, а з часом стала повністю дистанційною з використанням електронного цифрового підпису. Зокрема, програма M.E.Doc стала родоначальником електронного документообігу в країні.

1.6. Висновки до розділу 1

Програми лояльності - явище не нове, але не всякий бізнес може собі дозволити запустити і підтримувати таку систему самостійно. Іноді не вигідно випускати пластикові карти, розгортати у себе IT-інфраструктуру для їх обслуговування і наймати фахівців для її підтримки. Найцінніше в таких програмах - це, звичайно, призначені для користувача дані, і отримати їх від клієнтів критично важливо для бізнесу.

У розділі були проаналізовані системи обліку дисконтних карток які існують в Україні та які можливості надає програмне забезпечення цих систем. Було розглянуто поняття веб-сайту, його методи відображення і найпоширеніші типи.

Автоматизація підприємства є важливою темою при створенні системи автоматизації. Аналіз даної теми допоміг зрозуміти як правильно будувати архітектуру програмного забезпечення при розробці системи автоматизації обліку дисконтних карток.

РОЗДІЛ 2. АРХІТЕКТУРА І МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Робота з макетами програмного забезпечення

Для вирішення задачі Випускної кваліфікаційної роботи, необхідно розробити макет, потім приступати до реалізації програмного забезпечення. Це дасть змогу заздалегіть продумавати архітектуру для певного функціоналу зображенго на макеті.

Щоб створити макет можна використати один з найпопулярніших графічних редакторів Photoshop.

Adobe Photoshop — графічний редактор, розроблений і поширюваний фірмою Adobe Systems. Цей продукт є лідером ринку в галузі комерційних засобів редагування растрових зображень і найвідомішим продуктом фірми Adobe. Часто цю програму називають просто Photoshop (Фотошоп). У наш час Photoshop доступний на платформах Mac OS X/Mac OS і Microsoft Windows. Ранні версії редактора були портовані під SGI IRIX, але офіційна підтримка була припинена, починаючи з третьої версії продукту. Для версії CS і CS6 можливий запуск під Linux за допомогою альтернативи Windows API — Wine.

Photoshop головним чином призначений для редагування цифрових фотографій та створення растрової графіки. Особливості

Adobe Photoshop полягають у багатому інструментарії для операції створення і обробки зображень, високій якості обробки графічних зображень, зручності й простоті в експлуатації, широких можливостях до автоматизації обробки растрових зображень, які базуються на використанні сценаріїв, механізмах роботи з кольоровими профілями, які допускають їх втілення в файли зображень з метою автоматичної корекції кольорових параметрів при виводі на друк для різних пристроїв, великому наборі команд фільтрації, за допомогою яких можна створювати найрізноманітніші художні ефекти.

Базові інструменти редагування дозволяють змінювати тон, насиченість зображення, обтинати його, накладати фотофільтри, виправляти перспективу тощо. Photoshop підтримує так звані шари — прозорі області зображення, на яких розміщуються елементи фотомонтажу, текст, геометричні фігури. Програма містить інструменти для роботи з текстом і нескладними фігурами, дозволяє малювати робочі контури, задавати текстам і фігурам стилі оформлення. Для роботи з окремими фрагментами зображення передбачені різні типи виділення: за фігурою, в режимі «малювання» зони виділення, за діапазоном кольорів тощо. Існують різноманітні фільтри для деформації та стилізації зображення, такі як фільтри розмиття, імітації різних художніх технік. Photoshop також містить інструменти для цифрового живопису, зокрема набори пензлів. Користувач може змінювати їх розмір, кут нахилу, колір. Підтримується встановлення сторонніх пензлів, стилів, шрифтів, палітр. Попри те, що спочатку програма була розроблена як редактор зображень для поліграфії, в наш час вона широко використовується і у веб-дизайні. У більш ранній версії була включена спеціальна програма для цих цілей — Adobe ImageReady, яка була виключена з версії CS3 за рахунок інтеграції її функцій в сам Photoshop, а також включення в лінійку

програмних продуктів Adobe Fireworks, що перейшло у власність Adobe після придбання компанії Macromedia.

Photoshop тісно пов'язаний з іншими програмами для обробки медіафайлів, анімації та іншої творчості. Спільно з такими програмами, як Adobe ImageReady (програма скасована у версії CS3), Adobe Illustrator, Adobe Premiere, Adobe After Effects і Adobe Encore DVD, він може використовуватися для створення професійних DVD, забезпечує засоби нелінійного монтажу і створення таких спецефектів, як фони, текстури і т. д. для телебачення, кінематографу і всесвітньої павутини. Основний формат Photoshop, PSD, може бути експортований і імпортований всіма програмними продуктами, переліченими вище. Photoshop CS підтримує створення меню для DVD. Спільно з Adobe Encore DVD, Photoshop дозволяє створювати меню або кнопки DVD. Photoshop CS3 у версії Extended підтримує також роботу з тривимірними шарами.

Підтримується обробка зображень, як з традиційною глибиною кольору (8 біт, 256 градацій яскравості на канал), так і з підвищеною (16 біт, 65536 відтінків в кожному каналі). Можливе збереження у файлі додаткових елементів, як то: напрямних (Guide), каналів (наприклад, каналу прозорості — Alpha channel), шляхів обтравки (Clipping path), шарів, що містять векторні і текстові об'єкти. Файл може включати колірні профілі (ICC), функції перетворення кольору (transfer functions).

Photoshop підтримує такі колірні моделі або способи опису кольорів зображення (в нотації самої програми — режим зображення): RGB, LAB, CMYK, Grayscale, Bitmap, Duotone, Indexed, Multichannel.Через високу популярність Photoshop підтримка його формату файлів, PSD, була реалізована в його основних конкурентів, таких, як Macromedia Fireworks, Corel PHOTO-PAINT, Pixel image editor, WinImages, GIMP, Jasc Paintshop Pro і т. д.

2.2. Програмне середовище розробки

Для вирішення задачі дипломного проектування потрібні програмні засоби за допомогою яких, буде виконаємо багато операцій, які потребуються для створення інтернет магазину.

Node.js — платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Засновником платформи є Раян Дал (Ryan Dahl). Якщо раніше Javascript застосовувався для обробки даних в браузері користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

- Node.js має наступні властивості:
- асинхронна одно-нитева модель виконання запитів;
- неблокуючий ввід/вивід;
- система модулів CommonJS;
- рушій JavaScript Google V8;

Для керування модулями використовується пакетний менеджер npm (node package manager).

Платформа Node.js призначена для виконання високопродуктивних мережевих застосунків, написаних мовою програмування JavaScript. Платформа окрім роботи із серверними скриптами для веб-запитів, також використовується для створення клієнтських та серверних програм.

Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в неблокуючому режимі та визначенні обробників зворотніх викликів (callback). Як способи мультиплексування з'єднань

підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується бібліотека libuv, для створення пулу нитей (thread pool) задіяна бібліотека libeio, для виконання DNS-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що спричиняють блокування, виконуються всередині пулу нитей і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

За своєю суттю Node.js схожий на фреймворки Perl AnyEvent, Ruby Event Machine і Python Twisted, але цикл обробки подій (event loop) у Node.js прихований від розробника і нагадує обробку подій у веб-застосунку, що працює в браузері. При написанні програм для Node.js необхідно враховувати специфіку подієво-орієнтованого програмування, наприклад, замість виконання з очікуванням завершення роботи і наступною обробкою результатів, в Node.js використовує принцип асинхронного виконання, тобто код трансформується в При цьому управління миттєво перейде до коду який слідує після виклику функції db.query, а результат запиту буде оброблений як тільки будуть оброблені дані. Жодна функція в Node.js не повинна безпосередньо виконувати (блокуючі) операції вводу/виводу — для отримання даних з диска, від іншого процесу або з мережі потрібна установка callback-обробника.

Для розширення функціональності застосунків на базі Node.js підготовлена велика колекція модулів, в якій можна знайти реалізацію HTTP, SMTP, XMPP, DNS, FTP, IMAP, POP3 серверів і клієнтів, модулі для інтеграції з різними веб-фреймворками, обробники WebSocket і AJAX, конектори до СКБД (MySQL, PostgreSQL, SQLite, MongoDB), шаблонізатори, CSS-рушії, реалізації криптоалгоритмів і систем авторизації (наприклад, OAuth), XML-парсери.

MongoDB — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми

таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

СКБД управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS. Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

Є докладна і якісна документація, велике число прикладів і драйверів для популярних мов Java, C++, C#, PHP, Python, Perl, Ruby.

При випуску одразу було заявлено, що реліз MongoDB 1.0 готовий до використання у виробництві як одиничний хост, так і у зв'язках master/slave. Код цього релізу досить стабільний і перевірений в промисловій експлуатації протягом 1,5 років. MongoDB рекомендується

розгортати мінімум на двох серверах використовуючи реплікацію Master/Slave. Це забезпечує наявність актуальних даних при виході з ладу однієї з СКБД. MongoDB — продукт досить молодий, і відтак у ньому зустрічаються помилки, з'являються нові можливості тощо. Характерний високий темп розробки (проект пишуть не тільки волонтери, а й компанія людей на повній зайнятості). Компанія-розробник надає платні підтримку, хостинг, консультації.

Основні можливості MongoDB:

- Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних)
- Досить гнучка мова для формування запитів
- Динамічні запити
- Повна підтримка індексів
- Профілювання запитів
- Швидкі оновлення «на місці»
- Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео
- Журналювання операцій, що модифікують дані в БД
- Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг
- Може працювати відповідно до парадигми MapReduce

Vue.js — JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних.

HTML це мова, яка дозволяє представляти інформацію (наприклад, наукові дослідження) в Інтернеті. Те, що ви бачите при перегляді сторінки в Інтернеті, це інтерпретація вашим браузером HTML-тексту. Ви можете відобразити html код сторінки прямо у вікні

браузера. Для цього перейдіть до сторінки яка вас цікавить, і виберіть у браузері команду Вид-> Перегляд html-коду. Відкриється нове вікно браузера з html кодом сторінки. У цьому вікні можна не лише переглянути код, але й скопіювати його для використання в якості основи своєї web-сторінки.

Cascading Style Sheets (каскадні таблиці стилів) - технологія опису зовнішнього вигляду документа, написаного мовою розмітки. CSS використовується переважно для оформлення HTML- і XHTML-документів, але іноді і для інших XML-структурованих документів (наприклад, в браузере Mozilla для оформлення елементів графічного інтерфейсу, XUL).

CSS використовується розробниками веб-сторінок для завдання кольорів, шрифтів, розташування і інших аспектів представлення документа. Основною метою розробки CSS було розділення вмісту (написаного на HTML або іншій мові розмітки) від представлення стилю документа. Це розділення може збільшити доступність документа, надати велику гнучкість і можливість управління його виглядом, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях.

Стандарт CSS визначає пріоритети, у порядку яких застосовуються правила стилів, якщо для якогось елементу підходять деякі правила одночасно. Це називається "каскадом", в якому для правил розраховуються пріоритети або "ваги", що робить результати передбаченими.

Таблиця стилів складається з набору правил. Кожне правило, у свою чергу, складається з одного або декількох селекторів, розділених комами і блоку визначень.

JavaScript — це особлива мова програмування, який базується на об’єктному представленні браузера. Він потрібен для того, щоб надати сайту більше інтерактивності в порівнянні зі звичайним статичним HTML-документом. Наприклад, в інтерфейсі можна буде реалізувати мінливі малюнки, рухомий рядок з тексту та багато іншого! Відмінність JavaScript полягає в тому, що текст програми вбудовується в документ HTML і аналізується самим браузером. JavaScript — це мова програмування сценаріїв на веб-сторінках.

jQuery - бібліотека Javascript, що фокусується на взаємодії Javascript і HTML. Була опублікована на комп'ютерній конференції «Barcamp» у Нью-Йорку Джоном Ресигом в 2006 році. У чому головна перевага jquery? Вона закладена на рівні ядра - це вибір елементів об'єктної моделі документів. Крім того, завдяки наявності плагінів, базова функціональність jquery може бути розширена. Для початку роботи з jquery необхідно скачати саму бібліотеку з будь-якого доступного джерела. Далі її необхідно ініціалізувати.

Основне завдання jQuery — це надавати розробнику легкий та гнучкий інструментарій кросбраузерної адресації DOM об'єктів за допомогою CSS та XPath селекторів. Також даний фреймворк надає інтерфейси для Ajax-застосунків, обробників подій і простої анімації.

Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

Приклад власної наробки з використанням jQuery:

```
        $.ajax({
        url: 'index.php?route=checkout/cart/add', type:
'post',
        data: 'product_id=' + product_id + '&quantity=' +
(typeof(quantity) != 'undefined' ? quantity : 1),
        dataType: 'json',
        beforeSend: function() {$('#cart >
button').button('loading');},
        complete: function() {$('#cart >
button').button('reset');},
        success: function(json) {$('.alert-dismissible,
.text-danger').remove();
        if (json['redirect']) {location = json['redirect'];}
        if (json['success']) {
        $('#content').parent().before('<div class="alert
alert-success alert-dismissible"><i class="fa fa-check-
circle"></i> ' + json['success'] + ' <button type="button"
class="close" data-
dismiss="alert">&times;</button></div>');
    setTimeout(function () {
    $('#cart > button > #cart-
total').html(json['total']);}, 100);
    $('html, body').animate({ scrollTop: 0 }, 'slow');
    $('#cart > ul').load('index.php?route=common/cart/info
ul li');}},
    error: function(xhr, ajaxOptions, thrownError) {
    alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);});
```

2.3. Проектування бази даних

Велика різноманітність функцій NodeJS позбавлять вас від написання багаторядкових призначених для користувача функцій на C або Pascal.

Створення інформаційної системи - це складний процес, в якому бере участь колектив розробників, розбивається на стадії проектування, програмної реалізації та експлуатації.

У процесі створення інформаційної системи готуються робочі документи, що служать основою для всіх розробників і користувачів системи.

Проектування бази даних полягає в многорівневому описі майбутньої БД з різним ступенем деталізації і формалізації, в ході якого проводиться уточнення і оптимізація її структури.

Проектування включає опис предметної області і завдань інформаційної системи, далі йде до логічного опису даних і потім - до фізичної моделі БД. Розрізняють три етапи деталізації опису об'єктів БД і їх взаємозв'язків за трьома основними рівнями моделювання системи - концептуальному, логічному і фізичному.

На концептуальному рівні проектування проводиться смислове (семантичне) опис інформаційного змісту предметної області, визначаються межі предметної області, проводиться абстрагування від несуттєвих для даної інформаційної системи деталей. В результаті визначаються що моделюються об'єкти, їх властивості та зв'язку. Виконується структуризація знань про предметну область, стандартизується термінологія. Потім будується концептуальна модель, що описується природною мовою. Для опису властивостей і зв'язків об'єктів застосовують різні діаграми.

На наступному кроці приймається рішення про те, в який СУБД буде реалізована БД. Визначальними параметрами є вид програмного

продукту і категорія користувачів (професійні програмісти або кінцеві користувачі, або і те, і інше). Іншими показниками, що впливають на вибір СУБД, є: зручність і простота використання; якість засобів розробки, захисту і контролю БД; рівень комунікаційних засобів (застосування в мережах); фірма-розробник; вартість. Кожна конкретна СУБД працює з певною моделлю даних.

На логічному рівні проводиться відображення даних концептуальної моделі в логічну модель, підтримувану обраної СУБД. Тут об'єктом роботи виступають самі дані, їх структура і правила побудови. Логічна модель не залежить від конкретної СУБД - побудована на основі таблиць логічна модель може бути реалізована на будь-який СУБД реляційного типу.

На фізичному рівні проводиться вибір раціональної структури зберігання даних і методів доступу до них, вирішуються питання ефективного виконання запитів до БД, будуються додаткові структури, наприклад, індекси. У фізичної моделі міститься інформація про всі об'єкти БД (таблицях, індексах, процедурах і ін.) І використовуваних типах даних. Фізична модель залежить від конкретної СУБД. Однієї і тієї ж логічної моделі може відповідати кілька різних фізичних моделей. Фізичне проектування є початковим етапом реалізації БД.

Таблиця 2.1.

Структура таблиці «Користувачі»

№	Назва поля	Тип даних	Призначення
1	user_id	INTEGER	Первинний ключ
2	email	VARCHAR(64)	Е-мейл користувача
3	password	VARCHAR(64)	Пароль

№	Назва поля	Тип даних	Призначення
4	firstname	VARCHAR(64)	Ім’я
5	lastname	VARCHAR(64)	Прізвище
6	phone	VARCHAR(64)	Номер мобільного телефону
7	country	VARCHAR(64)	Країна
8	city	VARCHAR(64)	Місто
9	birth_date	DATE	Дата народження
10	reg_date	DATE	Дата реєстрації
11	cards	VARCHAR(255)	Картки

Таблиця 2.2.

Структура таблиці «Картки»

№	Назва поля	Тип даних	Призначення
1	card_id	INTEGER	ID картки
2	card_numer	INTEGER	Номер картки
3	card_owner	INTEGER	Зовнішній ключ, посилання на таблицю «Користувачі»
4	card_bal	INTEGER	Баланс карти

Таблиця 2.3.

Структура таблиці «Новини»

№	Назва поля	Тип даних	Призначення
1	news_id	INTEGER	ID новини
2	author	INTEGER	Зовнішній ключ, посилання на таблицю «Користувачі»
3	news_text	INTEGER	Текст новини
4	category	INTEGER	Категорія новини

Таблиця 2.4.

Структура таблиці «Пропозиції»

№	Назва поля	Тип даних	Призначення
1	promo_id	INTEGER	ID пропозиції
2	partners	INTEGER	Партнери які беруть участь
3	promos	INTEGER	Список товарів або послуг у пропозиції
4	category	INTEGER	Категорія новини

Таблиця 2.5.

Структура таблиці «Транзакції»

№	Назва поля	Тип даних	Призначення
1	trans_id	INTEGER	ID транзакції
2	promo_id	INTEGER	Зовнішній ключ, посилання на таблицю «Пропозиції»
3	trans_date	DATE	Категорія новини
4	user_id	INTEGER	Зовнішній ключ, посилання на таблицю «Користувачі»

2.4. Редактор коду PHP Storm

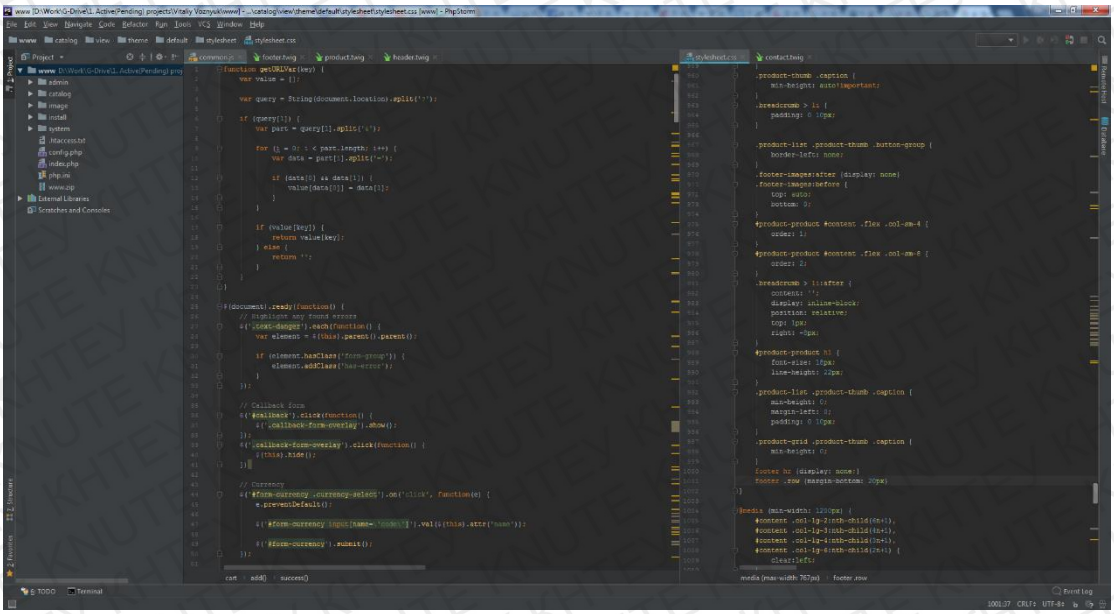

Рис. 2.1. Середовище розробки PhpStorm

JetBrains PhpStorm — комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3/5.4/5.5/5.6/7.0/7.1 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни.

Ключовою особливістю IDE є її розуміння коду, який ви пишете. IDE не має простого списку команд, які можна використовувати у вашій улюбленій мові, наприклад PHP. Він має складні алгоритми для розуміння вашого коду. PhpStorm не запропонує вам команду чи вираз, який буде синтаксично неправильним. Він знає, наприклад, що ви можете 't писати "print (' Hello World '); " безпосередньо всередині класу, не вставляючи його в функцію. Тому він не може запропонувати команду "print () ", коли вона не може бути використана.

Підсвічування коду

Це важливо. Більшість простих редакторів також можуть робити виділення основного коду, але IDE - це інша історія. Тут ви можете мати різні налаштування підсвічування для різних мов, таких як PHP, HTML, CSS і так далі. Кожен з його власними правилами. Наприклад, мені подобається мати рядки зеленими в CSS та оранжевому в PHP та HTML.

Зліва ви можете побачити всі підтримувані типи файлів і мови (деякі можуть бути додані плагінами); в центрі ви можете побачити різні елементи мови PHP, які можна налаштувати окремо, у верхньому правому - це параметри, кожен може мати елемент, а внизу - попередній перегляд

Коли ми проводили полювання на чарівні рядки в нашому сеансі рефакторінгу, ми змогли легко визначити ці рядки, просто знайшовши оранжевий матеріал на екрані. Те ж саме стосується магічних констант і чисел. Коли їх шукали, синій колір був нашим другом.

Структура та макет необхідні, коли нам потрібно розуміти вихідний код, і в цьому питанні важливо належне забарвлення та підсвічування.

PhpStorm може запропонувати, що жоден інший редактор або IDE для PHP не може зробити. Це набір рефакторингових інструментів, які настільки необхідні в нашій повсякденній роботі.

Підсвічування, відступи, управління проектами, шаблони, макроси - це всі функції, знайдені в більшості редакторів. Інтеграція з тестовими середовищами та відладчиками за визначенням є частиною будь-якої IDE. Проте розумні та складні інструменти рефакторингу - це зовсім інша історія.

Ще однією перевагою PhpStorm є те, що вам часто не доводиться проводити свої тести. Коли ми робили ручне рефакціонування, ми виконуємо наші тести після кожного кроку. Якщо ми зробили помилку або замінили те, що, на нашу думку, було однаковим, але насправді це не було, нам потрібна була захисна мережа, щоб швидко сказати нам про це. Однак PhpStorm не замінить те, що не є логічно правильним у контексті. Ви можете бути здивовані тим, що фрагмент коду, який ви також очікуєте отримати, не буде, але принаймні ви можете бути впевнені, що отриманий код буде правильним. Крім того, ми маємо менше кроків. Вручну ми робимо лише один крок. Після цього єдиного кросу ми повинні проводити наші тести один раз.

Це не має значення, як ви дивитесь на нього, легше, швидше і набагато менше помилок, ніж будь-який інший спосіб зробити рефакторинг.

2.5. FTP-менеджер FileZilla

FTP є скороченням від англ. File Transfer Protocol - протокол передачі файлів, який застосовується для обміну файлами по TCP / IP мереж між двома комп'ютерами (клієнт і сервер). Протоколу передачі файлів більше 40 років, він був розроблений перш ніж з'явився TCP / IP і тим більше HTTP, проте він досі актуальний і використовується для підключення до віддалених серверів і обміну файлами.

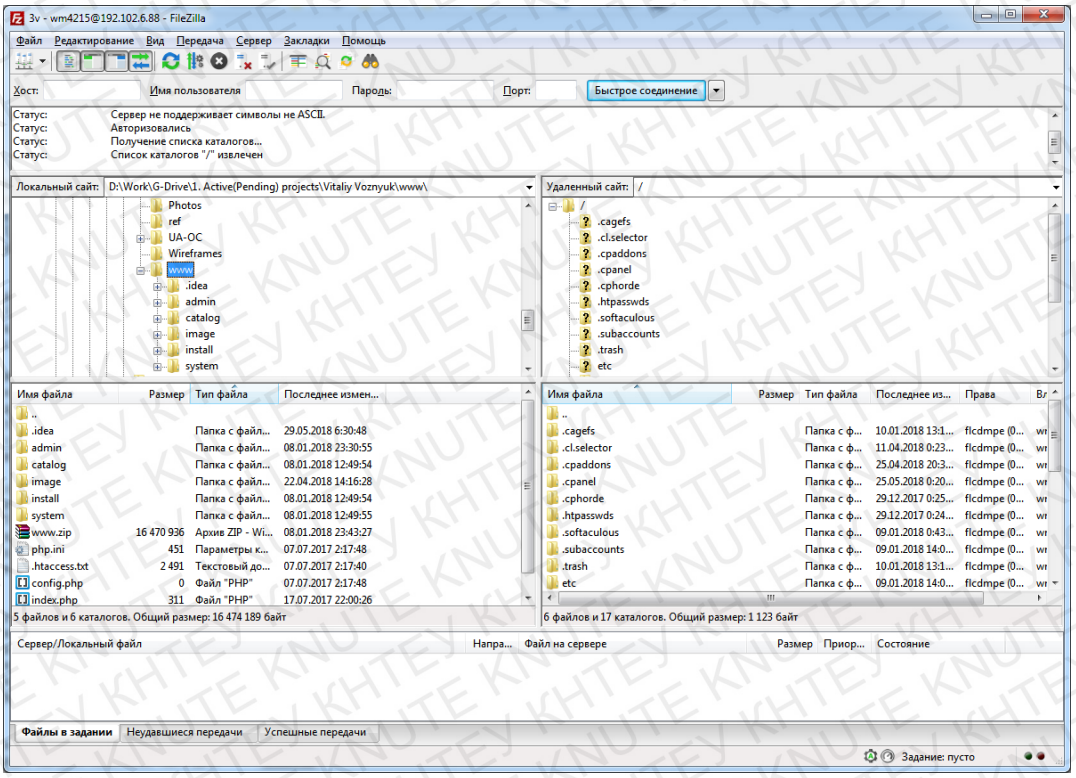

Рис. 2.2. FTP-менеджер FileZilla

Даний протокол застосовує різні мережеві з'єднання для передачі команд і файлів між клієнтом і сервером. FTP сервер, являє собою комп'ютер з встановленим на нього спеціальним програмним забезпеченням і очікують зовнішнього підключення від інших комп'ютерів.

FTP клієнт є програмою, яка робить спробу з'єднається з серверним комп'ютером, як правило до порту номер 21. Після успішного підключення до FTP-сервера, можна здійснювати різноманітні операції над даними розташовуються на ньому, зокрема, переглянути вміст каталогів, завантажувати і викачувати файли з FTP сервера, перейменовувати, призначати права доступу, видаляти файли з сервера та інше.

З'єднуючись з FTP сервером допустимо пройти авторизацію надаючи дані для входу без використання шифрування, а також можна

підключитися анонімно, якщо це дозволяє FTP сервер. Для захищеної передачі, яка приховує (шифрує) логін, пароль і дані, FTP з'єднання може бути зашифровано за допомогою SSL / TLS (FTPS). SSH протокол передачі файлів (SFTP) іноді також використовується, але між ним і FTPS є істотна різниця.

Основне призначення FTP протоколу - це завантаження файлів і їх скачування з віддаленого сервера. Для передачі файлів в пасивному режимі ініціюється з'єднання FTP клієнтом з обумовленого діапазону портів до порту сервера. В активному режимі FTP сервер підключається до клієнта з порту 20 до певного порту, який повідомив йому клієнт. Основна відмінність між даними режимами, полягає в тому, з якого боку відкривається з'єднання для передачі файлів.

Для ефективної роботи з FTP ми рекомендуємо скористатися FTP менеджером FileZilla. Програма відмінно підійде як для новачків, так і для досвідчених користувачів. Легкість освоєння, кросплатформенність, безліч підтримуваних мов, велика кількість налаштувань і можливостей роблять її однією з кращих FTP клієнтів.

2.6. Висновки до розділу 2

У розділі було проаналізовано розробку програмного забезпечення. Перший важливий етап при розробці, це макет програмного забезпечення та опис функціоналу фінального результату. Для розробки макету інтерфейсу був використаний візуальний редактор Adobe Photshop, в якому було створено майбутній макет програмного забезпечення. Також в розділі були проаналізовані інструменти які були використані при розробці системи, такі як PHP Strom, FileZilla.

Проаналізовано платформу NodeJS яка буде використана при розробці, а також проаналізовано і спроектовано базу даних для системи, яка буде створена за допомогою MongoDB.

Проаналізовано такі інструменти як HTML, JavaScript та CSS для створення веб-сторінок. З їх допомогою буде створений інтерфейс та функціонал для системи на основі платформи NodeJS.

РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ ОБЛІКУ

3.1. Вимоги до програмної реалізації

Після аналізу, були сформовані вимоги до програмного забезпечення, тому для реалізації системи були визначені такі найважливіші аспекти при розробці:

- Коректне відображення;
- Використання клієнт-серверної моделі;
- Кроссбраузерність системи;
- Швидкість системи;
- Зручність системи

Після аналізу, були сформовані вимоги до програмного забезпечення, тому для реалізації системи були визначені такі найважливіші аспекти при розробці:

Коректне відображення системи, є одним із головних аспектів, оскільки при неправильній візуалізації інтерфейсу програмного забезпечення, можуть некоректно працювати функції, а також заплутувати та вводити в оману користувача, що може вплинути на час виконання задач, зв’язаних з системою та вплинути на безпеку даних.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін

даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери — це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми — і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів — клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка застосунку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

3.2. Архітектура системи

При проектуванні архітектури системи та взаємодії її модулів, велику роль відіграє швидкість. При роботі з великою кількістю даних, деякі інструменти розробки не виправдовують витрати потужностей ситеми на обробку інформації, оскільки алгоритми розраховані на велику швидкість при не великих об’ємах інформації.

Після досліджень попередніх параграфів, можна дійти висновку, що оптимальний варіант для системи – веб-простір. Це дає змогу працювати

системі цілодобово, а також надати змогу користувачам використовувати систему з будь-якої точки світу.

Для забезпечення нормальної роботи системи при одночасному підключенні багатьох клієнтів та швидкості взаємодії з базою данних, була обарана платформа Node.js — платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Платформа Node.js призначена для виконання високопродуктивних мережевих застосунків. Платформа окрім роботи із серверними скриптами для веб-запитів, також використовується для створення клієнтських та серверних програм.

Візуальну складову системи буде відтворювати фреймворк VueJS — JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних.

Одна із найвиразніших особливостей Vue — це ненав'язлива реактивна система. Моделі це просто плоскі JavaScript об'єкти. Це робить керування станами дуже простим та інтуїтивним. Vue надає оптимізований ре-рендеринг з коробки без потреби робити що-небудь додатково. Кожен компонент слідкує за своїми реактивними залежностями під час рендерингу, тому система знає точно коли має відбуватись ре-рендеринг і які компоненти потрібно ре-рендерити.

Для отримання даних транзакцій дисконтних карток, ми будемо використовувати стороннє API, яке отримує всі запити зі сканерів та считувачів карт. Прикладниий програмний інтерфейс (англ. Application Programming Interface) — набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено - це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного

забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

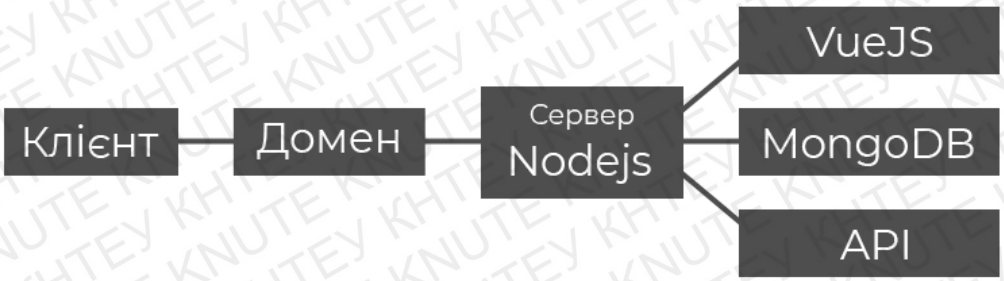

Рис. 3.1. Схема взаємодії користувач-система

3.3. Адаптивний інтерфейс

Адаптивний інтерфейс —веб-сторінки, що забезпечують оптимальне відображення та взаємодію сайту з користувачем незалежно від роздільної здатності та формату пристрою, з якого здійснюється перегляд сторінки.

Метою адаптивного інтерфейсу є практичне відображення інформації та зручна навігація на всіх пристроях із доступом до інтернету, від стаціонарних ПК до мобільних телефонів. За технологією адаптивного веб-дизайну не потрібно створювати окремі версії веб-сайту. Один сайт може працювати на всьому спектрі пристроїв.

Популярність адаптивного інтерфейсу зростає з кожним днем так як вже зараз кількість мобільного трафіку сягає більше половини від всього інтернет трафіку. Ця тенденція настільки поширена, що Google 21 квітня 2015 року запустив у своїй пошуковій системі алгоритм оцінки сайту на відповідність принципам «дружного» до мобільних пристроїв

інтерфейсу. Від цього показника залежить як високо сторінка буде представлена в результатах мобільного пошуку, а отже дана оцінка частково діє як штраф для сайтів, які не відповідають стандартам інтерфейсу для мобільних пристроїв.

У розробці ми використаємо декілька методів створення адаптивного інтерфейсу, для коректного відображення системи на різних електорнних пристроях.

Перший із методів створення адптивного інтерфейсу – це застосування CSS медіа-запитів. Медіа запити (media queries) - це правила CSS, які дозволяють управляти стилями елементів в залежності від значень технічних параметрів пристроїв. Іншими словами, це конструкції, які дозволяють визначати на підставі деяких умов які стилі необхідно використовувати на веб-сторінці, а які ні.

Другий метод створення адптивного інтерфейсу – використання JavaScript, оскільки ми можемо відслідковувати зміну довжини чи висоті екрана, отримати їх значення, а також є можливість запустити довільну функцію при певних умовах.

3.4. Функціональні можливості

У підскумковій системі, є три основних функціональних компонента: інформаційний сайт, особистий кабінет користувача, особистий кабінет адміністратора. Приклади знімків екрану працюючої програми можна знайти в Додатку 1. Для використання системи користувачу потрібно пройти авторизацію або зареєструватися, а до того часу він буде бачити головну сторінку системи.

Рис. 3.2. Головна сторінка

Кожен зареєстрований користувач отримує доступ до особистого кабінету який містить в собі основну інформацію системи та користувача. До кожного облікового запису можна додати дисконтну картку, щоб отримати можливість користуватися функціоналом системи, а також користуватися акційними пропозиціями. Для залучення клієнтів до отримання карти, на усіх сторінках є помаранчева кнопка, що дає можливість користувачу будь-де перейти до операції додавння карти.

З одного боку, все просто і зрозуміло: Комфорт Кард надає користувачам знижки в обмін на інформацію про себе. Однак процес реєстрації може в буквальному сенсі розполохати всіх користувачів

(особливо мобільних). За різними оцінками, від 15% до 68% користувачів видаляють свіжовстановленому додатки, якщо їм здається, що onboarding (процес реєстрації) занадто довгий. Тому зручна реєстрація стала одним із ключових завдань.

Було прийнято рішення вирішити цю задачу за допомогою досить простої хитрості: при первинній реєстрації користувач вводить тільки ім’я та номер телефону. Все, кабінет користувача готовий до використання і накопичення балів Комфорт Кард. Додаткові відомості знадобиться вводити тільки коли користувач захоче витратити накопичені бали. Тоді у нього з'явиться набагато більший стимул вводити потрібну інформацію.

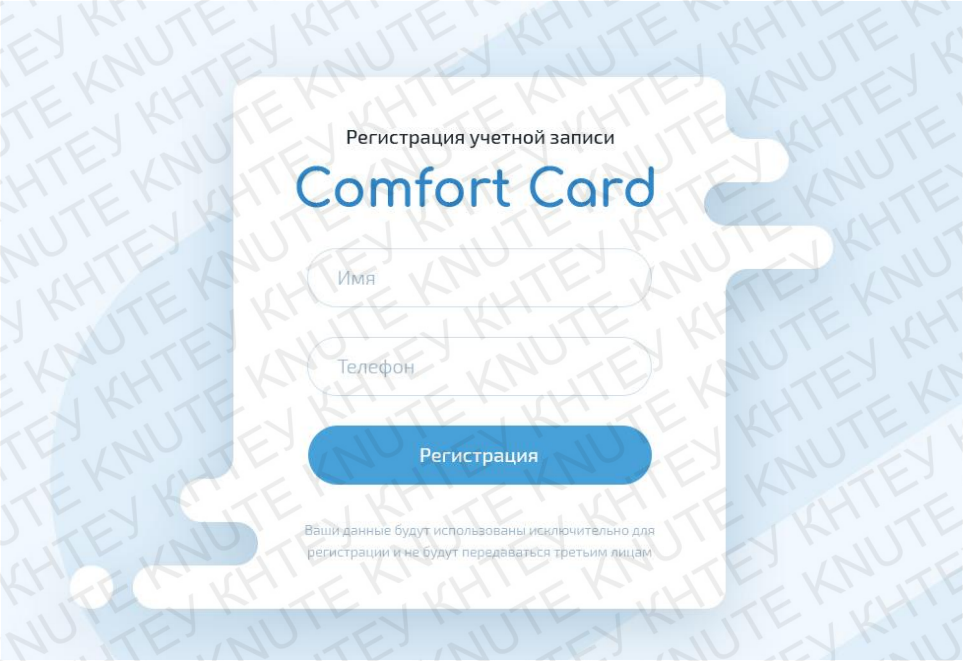

Рис. 3.3. Реєстрація користувача

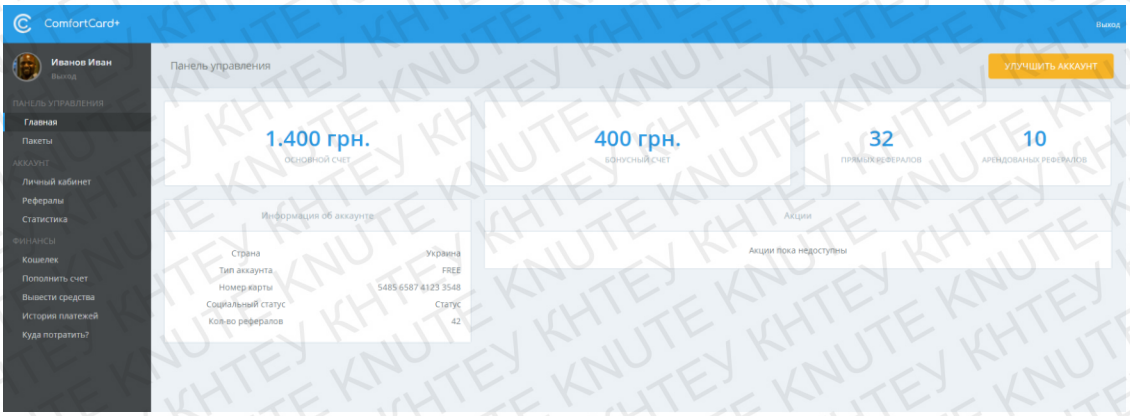

Рис. 3.4. Головна сторінка особистого кабінету

У користувача присутня реферальна система яка відображається у пункті меню «Реферали». Під реферальною системою мається на увазі змога користувача отримувати бонус або іншу винагороду за запрошення. Користувач може запросити будь-кого відправивши персональне посилання на сайт, яке містить в собі ідентифікатор запрошуючого.

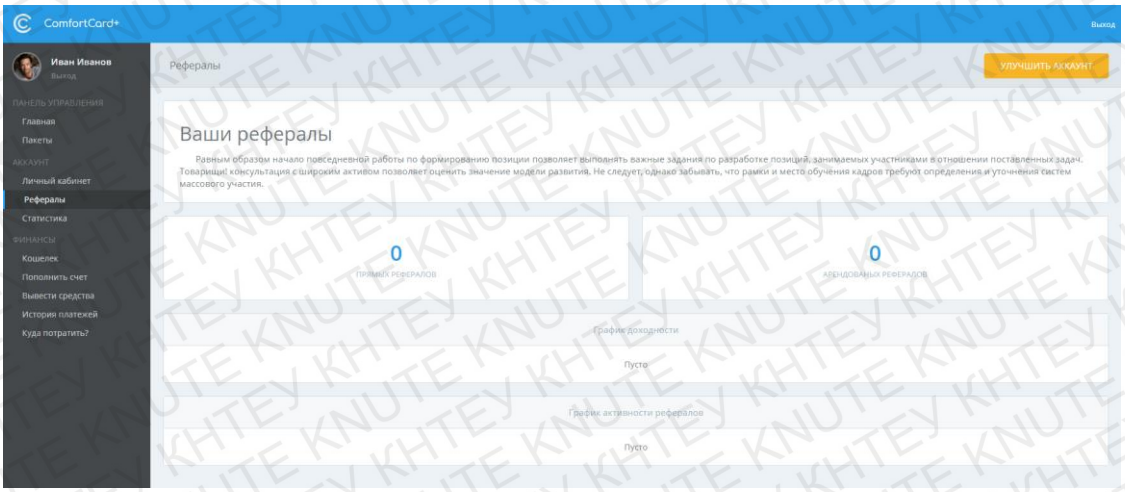

Рис. 3.5. Сторінка «Реферали»

Кожен користувач має сторінку профіля де він може змінити, додати або видалити свої персональні дані. В системі зберігається така

інформація про користувача: ім’я, прізвище, електронна скринька, телефон, країна, місто, дата народження.

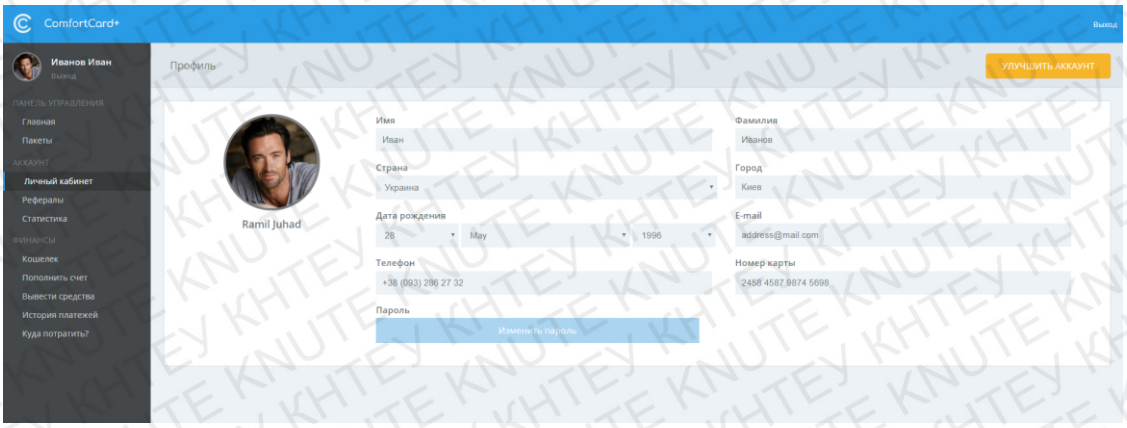

Рис. 3.6. Профіль користувача

Найцікавіша сторінка для користувача та володаря дисконтної картки є сторінка з пропозиціями від партнерів. На ній користувач може обрати вигідну пропозицію, щоб обмінати накопичені бали, купити будь-яку продукцію, зробити замовлення, тощо.

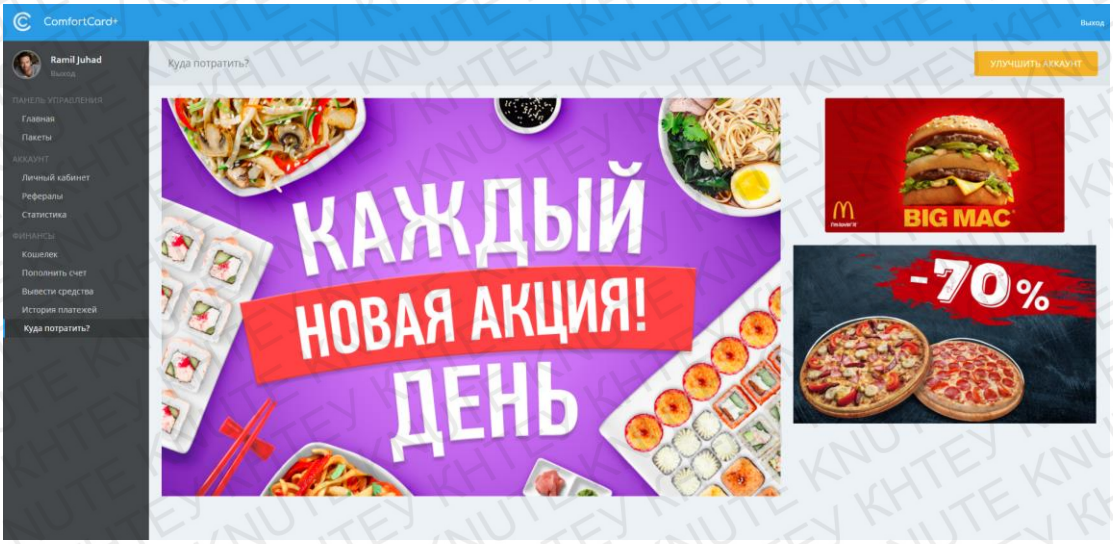

Рис. 3.7. Профіль користувача

У адміністратора є можливість керувати пропозиціями, додавати, відаляти та редагувати. Також адміністратор має змогу обрати для кого доступна дана пропозиція. Наразі є декілька типів користувачів: гість, звичайний користувач, володар картки, срібній володар картки, золотий володар картки, адміністратор.

Для адміністратора існує окрема сторінка, для маніпулювання даними зв'язаними з картками, користувачами, пропозиціями та транзакціями проведеними на сайті та поза ним.

В автоматизованій системі обліку дискотних карток важливим аспектом роботи адміністратора є обробка великої кількості інформації. Можливість імпорту і експорту файлів прискорює роботу підприємства і дає змогу зосередитись на важливих речах, а не на технічних вадах.

Оскільки безпека даних в такій системі є безумовно важливою, всі форми на сайті захищені проти зловмисних ін'єкцій, підбору паролей, неправильно введених даних. Перевірка відбувається на стороні клієнта, тобто у браузері користувача. Таким способом ми знижуємо можливість зависання або уповільнення роботи серверу, при обробці багатьох користувачів одночасно.

Два інших великих компонента системи, з якими було потрібно інтегруватися - rewards-платформа (система заохочень), що дозволяє з користю витратити накопичені бали, і карта партнерів та розробка особистиого кабінету користувача. Окрема історія стосується побудови БД яка повинна швидко працювати для відображення даних в реальному часі.

Серед цікавих завдань по частині бекенда було таргетування пропозицій певним учасникам по ряду ознак - статтю, віком, країні проживання, поточного географічному положенню і датою. Крім того, пропозиції можуть бути безпосередньо таргетовані на конкретних користувачів, а частина з них повинна відображатися клієнтам тільки до

першого користування послугою. Також потрібно було придумати архітектуру, яка дозволила б швидко шукати відповідні пропозиції для кожного користувача.

Push-повідомлення також персоналізовані і при необхідності можуть бути відправлені тільки за потрібне сегментам користувачів. Також є кілька категорій автоматичних повідомлень - наприклад, привітання з днем народження, або, наприклад, якщо користувач довго не заходив в додаток, йому прийде запрошення нарешті це зробити.

3.5. Висновки до розділу 3

Розробляючи комплексні автоматизовані системи обліку для забезпечення швидкої роботи сайту та бази даних потрібно заздалегідь підготувати сценарії використання системи та на основі вихідних даних проектувати архітектуру проекту. При використанні такого методу розробник має змогу визначити на ранніх етапах розробки які модулі програмного забезпечення можна застосувувати при їх необхідності для оптимізації та пришвидшення роботи системи.

В наш час, технології дають змогу відправляти, отримувати та обробляти інформацію в реальному часі, тому їх використання позитивно впливає на залучення нових клієнтів.

Також необхідним критерієм є правильна архітектура бази даних та вибір оптимальної СУБД, що прискорює роботу з базою даних.

Використовуючи найбільш популярні мови та технології ми маємо змогу знайти оптимальні рішення для вирішення задач розробки програмного забезпечення і надати функціональний та зручний інтерфейс, потрібний для вирішення задач користувача.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

На основі аналізу предметної області, існуючих рішень та популярних інструментів. Для розробки програмного забезпечення були обрані оптимальні програмні ресурси такі як NodeJS який є серверною платформою, MongoDB як базу даних та використано допоміжні програми для прискорення розробки.

Структура подібної системи створюється після аналізу багатьох факторів. Кожен його елемент продумується до дрібниць з метою досягнення максимальної оптимізації.

Було створено потужний інструмент який має достатньо функціоналу щоб вести повноцінну діяльність пов’язану з використанням дискотних карток. Систему можна масштабувати додаючі нові модулі, так як код написано модульним методом що дає змогу не використовувати непотрібні модулі, а тому не витрачати даремно потужності сервера. Це в свою чергу дало змогу підвищити об’єм відвідувачів які одночасно може обробляти сервер.

Розроблено особистий кабінет який є зручним і зрозумілим у викристанні для користувачів. Користувач має змогу маніпулювати своїми даними в налаштуваннях облікового запису, а також відстежувати зміни на рахунку або активність своїх рефералів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про захист персональних даних» (Відомості Верховної Ради України (ВВР), 2010, № 34, ст. 481)

2. Г.М. Бойко, А.М. Бакал (НПУ імені М.П. Драгоманова). “Електронний журнал як система інформаційної підтримки кредитно-модульної організації навчального процесу”

3. WEB-додаток Instudies [Електронний ресурс]. – Режим доступу: instudies.com (дата звернення 05.03.2014)

4. Уразливості веб-серверів [Електронний ресурс]. – Режим доступу: http://alls.in.ua/45916-urazlivosti-veb-serveriv.html (дата звернення 15.03.2014)

5. Т. Карпова. Базы данных. Модели, разработка, реализация. СПб.: Питер, 2001. - 304 с.

6. Електронний посібник з мови розмітки гіпертексту HTML [Електронний ресурс]. Режим доступу: http://htmlbook.ru/html5

7. Електронний посібник по CSS [Електронний ресурс]. Режим доступу: http://www.w3schools.com/css/

8. Електронний посібник JavaScript [Електронний ресурс]. Режим доступу: https://learn.javascript.ru/

9. Електронний посібник Adobe Photoshop [Електронний ресурс]. Режим доступу: https://helpx.adobe.com/ru/photoshop/user-guide.html

10. Способи попадання вірусів на сайт [Електронний ресурс]. – Режим доступу: http://www.siteguard.ru/p9.html

11. Класи атак [Електронний ресурс]. – Режим доступу: http://bug.kpi.ua

					КНТЕУ 121– 05-10.МР		
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка ситеми автоматизації обліку дисконтних карток ТОВ «Комфорт Кард»	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.				СВД	54	55
Керівник		Рзаєва С.Л.						
Гарант		Криворучко О.В			СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розроб		Гулаков В.В.						

12. Кваліфікаційні вимоги. [Електронний ресурс]. – Режим доступу: http://www.jobs.ua/ukr/dkhp/articles-2949/

13. DoS-атака. Стаття з вільної енциклопедії знань «Вікіпедія» [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org/wiki/DoS-атака

14. Горбунов-Посадов М. М. Расширяемые программы. – М.: Полиптих, 1999. – 336 с. – ISBN 5-901-11801-4

15. Інформаційна система Байкальського державного університету економіки і права [Електронний ресурс]. – Режим доступу: http://www.isea.ru/edu-portal/ (дата звернення 06.03.2014)

16. Інформаційна система «Електронний університет» Хмельницького національного університету [Електронний ресурс]. – Режим доступу: http://isu1.tup.km.ua/isu/ (дата звернення 06.03.2014)

17. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский / СПб: Питер, 2000.-384 с.

18. Ржеуцька С.Ю. - Бази даних. Мова SQL. / К. Вологда: ВоГТУ – 2012.

19. Тестування [Електронний ресурс]. – Режим доступу: http://www.yiiframework.com/doc/guide/1.1/uk/test.overview

20. Гольцман В. MySql 5.0. Библиотека программиста. (Интерфес с PHP, Perl, Java) / В. Гольцман - Спб: БХВ-Перербург, 2005. – 252с.

21. Грабер М. " SQL. Справочное руководство" / М. Грабер – М.: "Лори", 2000, - 291 с.

22. Ебби Майкл. Oracle 9i: Первое знакомство : пер. з англ. / Майкл Ебби, Майкл Кори, Йен Амбросон – К. : «Лори», 2003. – 506 с.

23. Каучмен Джейсон С. OCA Oracle9i Associate DBA. Подготовка администраторов баз данных: пер. с англ. / Джейсон С. Каучмен, Судхир Н. Марисети – М.: Издательство «Лори», 2006. – 680с.

ДОДАТКИ