**Чернівецький національний університет імені Юрія Федьковича**

**Інститут фізико-технічних та комп’ютерних наук**

(повна назва інституту/факультету)

**Кафедра комп’ютерних наук**

(повна назва кафедри)

**Веб-додаток для пошуку та створення кулінарних рецептів «Кулінарна книга»**

**Дипломний робота**

**Рівень вищої освіти - перший (бакалаврський)**

Виконала: студентка 4 курсу, групи 444

Напряму підготовки «Комп’ютерні науки»

(шифр і назва спеціальності)

Марковська Мілана Олегівна

(прізвище та ініціали)

Керівник \_ \_ Яцько О.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

(прізвище та ініціали, кафедра, університет)

Національна шкала\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_

Кількість балів \_\_\_ \_ \_\_Оцінка: ECTS\_ \_\_

**Голова ЕК** \_\_\_\_\_\_\_\_\_\_\_\_ \_ \_ \_ \_\_\_

(підпис) (прізвище та ініціали)

**До захисту допущено:**

**Протокол засідання кафедри №**

від „9” червня 2019 р.

зав. кафедри\_\_\_\_\_\_\_\_\_проф. Ушенко Ю.О.

**ЗМІСТ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ,  ОДИНИЦЬ ТА ТЕРМІНІВ | | | |  |
| [ВСТУП](#page25) | | | |  |
| [1](#page31). | АНАЛІЗ ПРЕДМЕТОНОЇ ОБЛАСТІ [ОГЛЯД ТА ПОРІВНЯЛЬНІЙ АНАЛІЗ АНАЛОГІВ ОБ’ЄКТА](#page31) ДОСЛІДЖЕННЯ | | |  |
|  | 1.1 | | Аналіз предметої області. Обгрунтування необхідності створення програсного продукту |  |
|  | 1.2 | | Порівняльний аналіз аналогів об’єкта досліджень |  |
|  | 1.3 | | Технічне завдання на розробку програмного забезпечення до інформаційної системи |  |
| [2](#page44). | [АНАЛІЗ І ПРОЕКТУВАННЯ СИСТЕМИ](#page44) | | |  |
|  | [2.1](#page47). | Обгрунтування вибору засобів для вирішення поставлених задач | |  |
|  | [2.2](#page56). | Проектування логіки, архітектури та поведінки системи | |  |
| 3. | [РОЗРОБКА СИСТЕМИ](#page132) | | |  |
|  | [3.1](#page132). | Алгоритми обробки та аналізу даних | |  |
|  | [3.2](#page137). | Розробка бази даних | |  |
|  | [3.3](#page139). | Розробка інтерфейсу системи | |  |
|  | 3.4 | Опис розробки програмних компонентів | |  |
| 4. | ПРИКЛАД РОБОТИ СИСТЕМИ | | |  |
| [ВИСНОВКИ ........................................................................................](#page141) | | | |  |
| [СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.........................................................](#page142) | | | |  |
| [ДОДАТКИ ..........................................................................................](#page143) | | | |  |
| ДОДАТОК А | | | |  |
| ДОДАТОК Б | | | |  |

# 

**АНОТАЦІЯ**

В зв’язку з популярністю різних служб доставок та популяризації іноземних страв з різних відео та фото-ресурсів, актуальним є розробка веб-програми «Кулінарна книга». Даний вебсайт дає можливість користувачу переглядати рецепт необхідному йому рецепту, фільтрувати їх та створювати власний.

В ході виконання дипломної роботи розроблено веб-додаток «Кулінарна книга» з назвою Culinary Book.

Було проаналізовано предметну область програмного продукту та описано специфікацію вимог. Здійснено аналіз існуючих аналогів, визначено їх характеристики та обґрунтовано розробку даного кінематографічного вебсайту.

Реалізовано програмний продукт за допомогою таких технологій, як Angular, JavaScript, PostGre, Node js та інших додаткових бібліотек, фреймворків. Методи розробки базуються на використанні середи розробки WebShtorm.

Здійснено опис інструкції користувача, наведено приклад використання системи.

Розроблений веб-додаток зручний у використанні, є адаптивним під всі пристрої та значно спрощує пошук рецептів з усією необхідною інформацією.

**Ключові слова:** КУЛІНАРНА КНИГА, ВЕБ-ДОДАТОК, РЕЦЕПТ, ІНФОРМАЦІЯ, ВЕБСАЙТ, ФРЕЙМВОРК, БІБЛІОТЕКА.

# **ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ТА ТЕРМІНІВ**

|  |  |  |
| --- | --- | --- |
| API |  | Опис способів (набір класів, процедур, функцій, структур або констант), якими одна комп'ютерна програма може взаємодіяти з іншого програмою |
| CSS |  | Мова, що використовується для опису зовнішнього вигляду сторінок (Cascading Style Sheets) |
| HTML |  | HyperText Markup Language |
| IDE |  | Інтегроване середовище розробки |
| SCSS |  | Препроцесор для CSS, який дозволяє значно пришвидшити написання коду |
| REST API |  | це загальні принципи організації взаємодії додатка / сайту з сервером за допомогою протоколу HTTP. Особливість REST в тому, що сервер не запам'ятовує стан користувача між запитами - в кожному запиті передається інформація, що ідентифікує користувача і всі параметри, необхідні для виконання операції. |
| ANGULAR |  | JS-фреймворк з відкритим програмним кодом, який розробляє Google. Призначений для розробки односторінкових додатків, що складаються з одної HTML сторінки з CSS і JavaScript. Його мета — розширення браузерних застосунків на основі шаблону Модель-вид-контролер (MVC), а також спрощення їх тестування та розробки. |
| JS |  | динамічна, об'єктно-орієнтованапрототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки. |
| TS |  | - мова програмування, представлена Microsoft восени 2012; позиціонується як засіб розробки веб-застосунків, що розширює можливості JS.  TypeScript є зворотньо сумісним з JavaScript. Фактично, після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою Node.js. |
| Rx JS |  | -це бібліотека для реактивного програмування за допомогою Observables, щоб полегшити складання асинхронного коду або коду на основі зворотного виклику. Цей проект є переписом Reactive-Extensions / RxJS з кращою продуктивністю, кращою модульністю, кращими налагоджувальними стеками викликів, залишаючись в основному зворотно сумісними, з деякими ламаючими змінами, що зменшують поверхню API |
| Node js |  | платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JS. Дає виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їхнього виконання.. |
| Express |  | програмний каркас розробки серверної частини веб-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення.  Він спроектований для створення веб-застосунків і API |
| UML |  | Мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення |
| СУБД |  | Система управління базами даних |

ВСТУП

Кожен день люди готують собі їсти, і в наш час – який насичений різними доставками і кафе/ресторанами, люди надихаються і хочуть повторити те що їм сподобалось самотужки. На допомогу ім приходять такі ресурси як веб-додатки з кулінарними рецептами.

Темою дипломної роботи є веб-додаток для пошуку та створення кулінарних рецептів.

**Актуальність роботи**. Тема веб-додатку «Кулінарна книга» стає з кожним роком більш актуальна, через пониження популярності у книжкових/газетних та журнальних таких довідників, також популяризація закордоних страв також впливає на проблему. Шукаючи в інтернеті аналоги я побачила значну нестачу таких довідників на українській мові, знайшла я не більше двух і зрозуміла, що вони більше не редактуються і не оновлюються. Таким чином додаток дозволяє реалізувати всі необхідні користувачу функції, такі як відображення, пошук, фільтрація, перегляд відеоресурсу про певний рецепт, спрощено візуальний інтерфейс який дозволяє швидко адаптуватися та зручно працювати на будь-яких сучасних мобільних девайсах

**Наукова новизна.** Веб-додаток має зручний інтерфейс та написаний за допомогую сучасних технологій які будуть ще довго підтримуватись розробниками та будуть актуальними( фреймворк та бібліотеки під нього ), спрощено візуальний інтерфейс який дозволяє швидко адаптуватися та зручно працювати на будь-яких сучасних мобільних девайсах..

**Структура та обсяг роботи.**  ДР складається зі вступу, … розділів, висновків, списку використаних джерел (… найменування), … додатки. Загальний обсяг роботи становить … сторінок основного тексту, … рисунки та … таблиць.

**РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ ОБ’ЄКТА ДОСЛІДЖЕННЯ**

* 1. **Аналіз предметої області. Обгрунтування необхідності**

**створення програмного продукту**

Культ їжі притаманний людям ще з давніх часів, ця тема завжди романтизувалась і була в попиті завжди, адже це необхідність для життя та отримання задоволення від чогось нового та смачного. Під час прийому їжі люди спілкуються, збирається сім’я, друзі, закохані, ці моменти – відкладаються в пам’яті, адже вони пов’язані не лише з гарними бесідами, а й їжою, яка є невід’ємною частиною нашого життя.

Тому предметною областю дипломної роботи є створення веб-ресурсу для перегляду та пошуку всієї необхідної інформації про певну страву. Адже вибрати рецепт, не прочитавши про нього нічого, та не подивившись про нього інформацію, таку як опис, перелік інгрідієнтів, фотографію експозиції - неможливо.

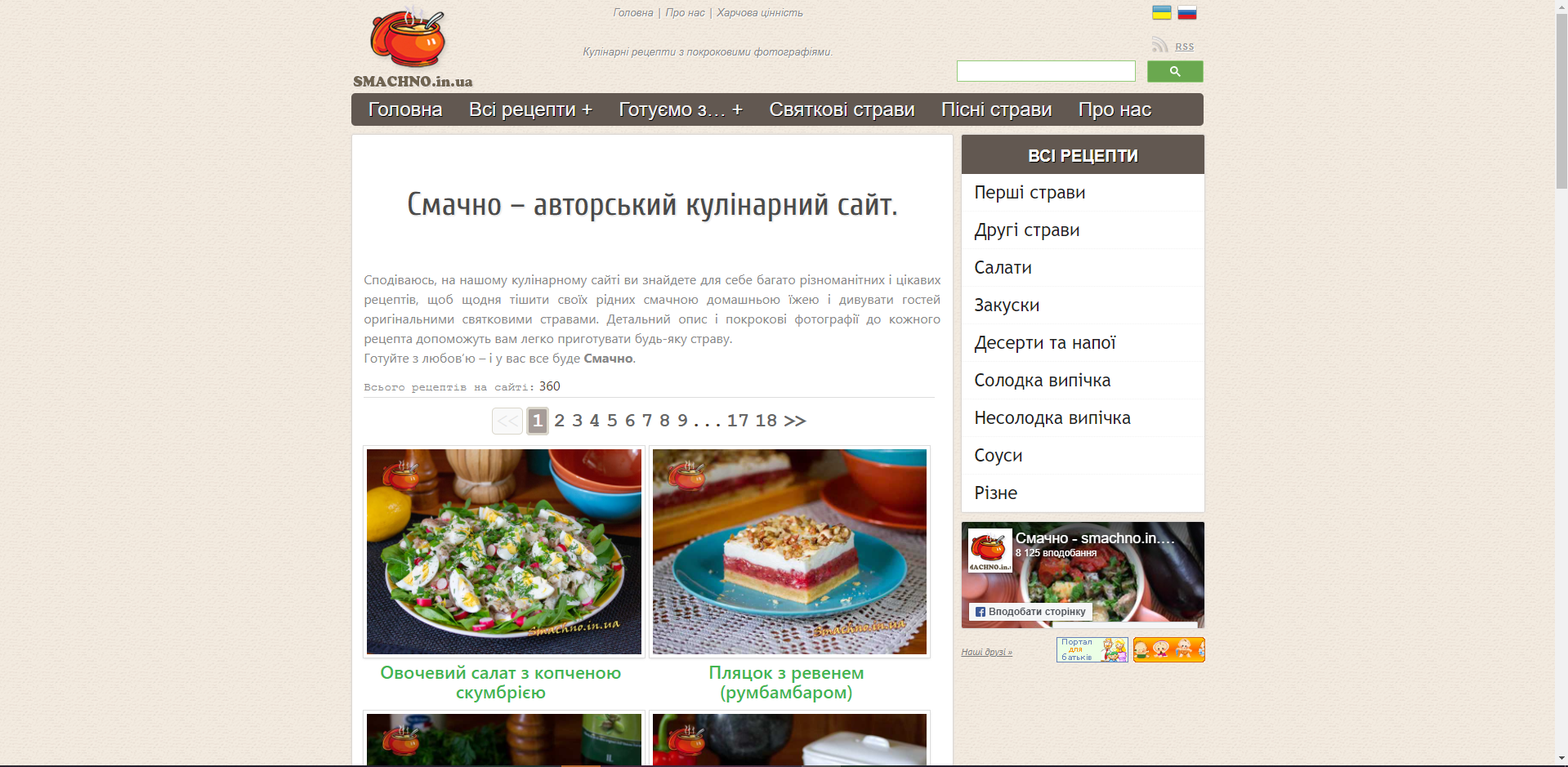
Куланарна книга – додаток який необхідний для пошуку рецептів, він містить достовірну інформацію, таку як перелік інгрідієнтів, відео-ресурсу з інструкцією, тощо, все що необхідно для звичайного користувача який випадково зайде на веб-сайт для пошуку страви на сніданок чи вечерю. Веб-додаток буде добре відображатись і на мобільних приладах, що також сприяє до збільшення популярності ресурсу. Кулінарна книга має сучасний дизайн який базується на приємних та теплих колярах з різними анімаціями, що надають додатку живності та сприяють покращенню візуальної складової для користувача. До того, існує значна нестача існуючих україномовних ресурсів

* 1. **Порівняльний аналіз аналогів об’єкта досліджень**

Шукаючи аналоги мого програмного продукту я знайшла онлайн-ресурс «smachno.in.ua» за посиланням [1]. Данний ресурс має велику кількість різноманітних піднастройок під фільтрацію, вона призначена для пошуку різних страв під певний період дня, і також має можливість міняти мову на російську.

Головну сторінку та вкладку з певним рецептом вказано на рисунку 1.1

А)



Б)

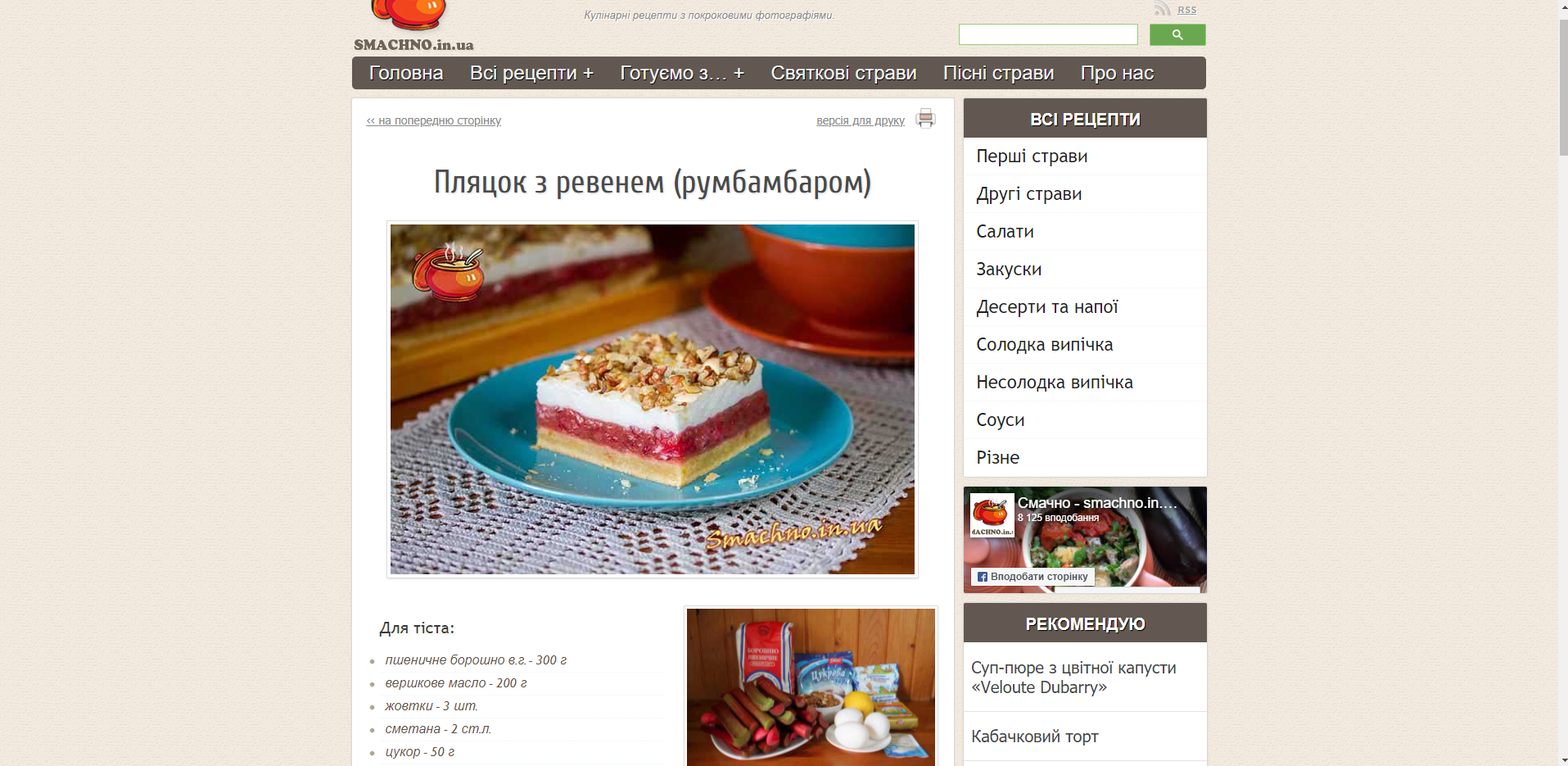


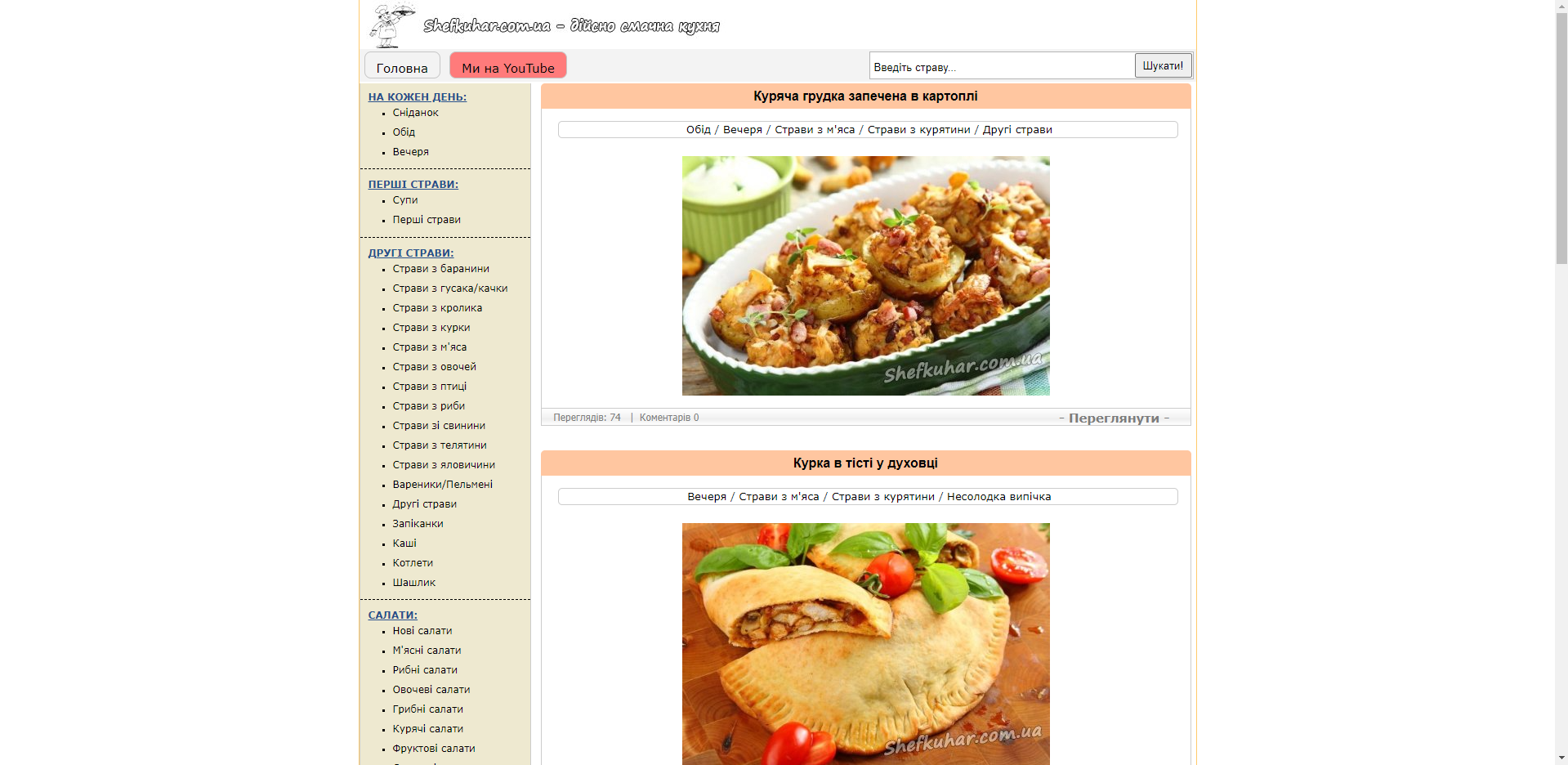
Рисунок 1.1 – Головна сторінка «а», сторінка певного рецепта «б» smachno.in.ua

На представленому веб-ресурсі для пошуку рецептів до страв є недоліки: важкий дизайн( великі відступи, малий шрифт, відсутність анімацій, немає відео-ресурсу з інструкцією, немає функції для користувача – створити власний рецепт, та перенасичення непотрібними для користувача різних вкладок.

Також за приклад я обрала веб-додаток «shefkuhar.com.ua». На ресурсі є більше фільтрів для пошуку, але відсутній повністю дизайн та будь-яка взаємодія з користувачем, що робить його сухим і недоробленим для користувача.

Вигляд головної сторінки та вкладки з певним рецептом можна переглянути на рисунку 1.2

а)



Б)

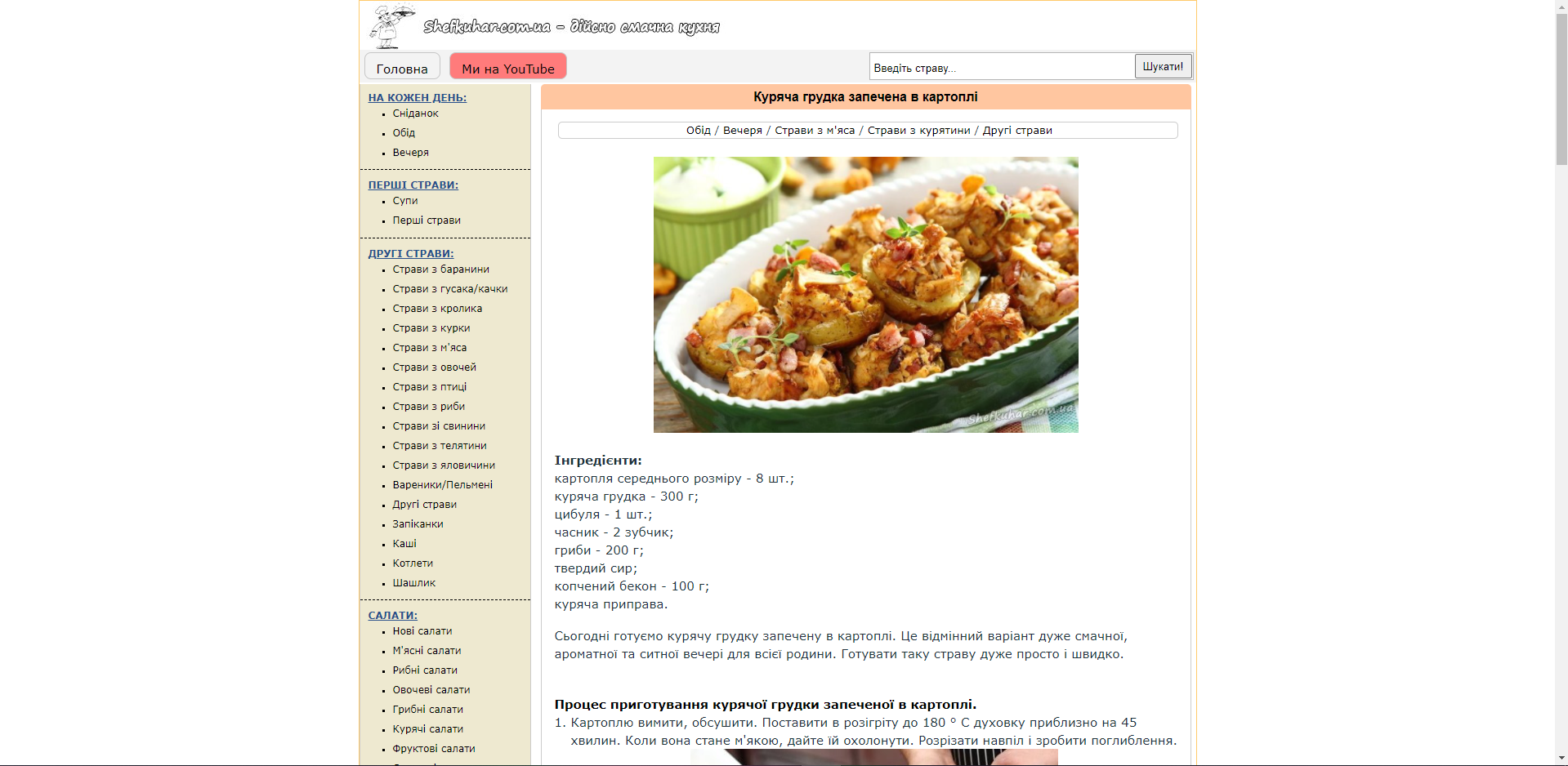


Рисунок 1.2 – Головна сторінка «а», сторінка певного рецепта «б»

**1.3. Технічне завдання на робзробку програмного забезпечення до інформаційної системи**

Метою створення специфікації вимог до даного веб-додатка є визначення функцій системи та її не функціональних вимог.

Основними не функціональними вимогами до «Кулінарної вниги» є:

* Зручний для користувача інтерфейс, приємні коляри
* Адаптивність( підходить під всі пристрої )
* Робота з СУБД PostGre
* Структурований та оптимізований код
* Використання сучасних технологій

Основні функціональні вимоги( можливості ), які має виконувати даний додаток, є :

- Фільтрація рецептів за назвою

- Перегляд певного рецепту

- Вкладка де користувач може створити власний рецепт

- Можливість переглядати відео-ресурс з певним рецептом

- Адаптивність

Метою створення даного ресурса є спрощення пошуку користувачу певних страв і інформації до нього

**Висновки до першого розділу**

У першому розділі за метою було виконання та обгрунтування вибору задач для дипломного дослідження «Кулінарна книга». Проведене дослідження аналогів на вільному ринку показало, що є нестача ресурсів на українській мові і що є попит на них. Враховуючи недоліки та переваги кожного з них, було виявлено, що вони не зовсім вирішують відповідним чином функціональні та не тільки функціональні вимоги користувачів.

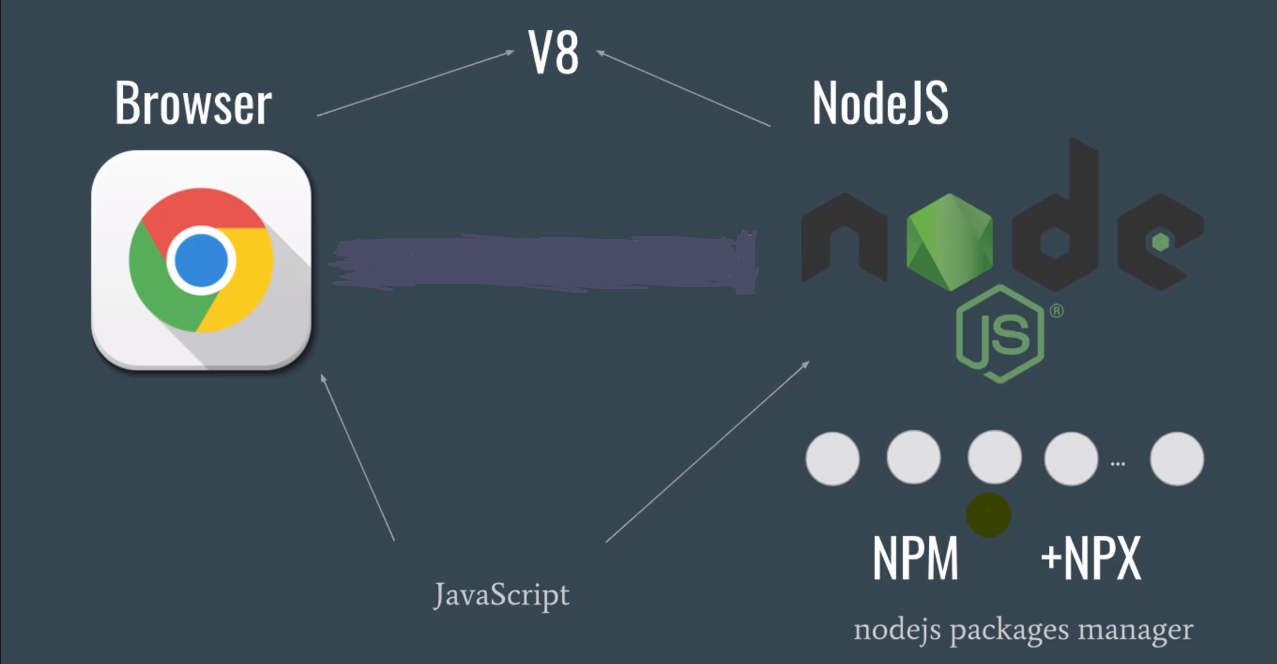
Описані в основних моментах інструменти які використовувались для розробки ДР, більше деталізовано про які буде відведено в 2гому розділі.

Звертаючи увагу на все перераховане було поставлено завдання до створення власного веб-ресурсу, і як результат в кінці ми отримали зручний і компактний веб-застосунок «Кулінарна Книга».

**РОЗДІЛ 2 АНАЛІЗ І ПРОЕКТУВАННЯ СИСТЕМИ**

**2.1. Обґрунтування вибору засобів для вирішення поставлених задач**

На сьогоднішній день, уявити розробку Клієнтської частини( front-end ) майже не можливо без 3 основних ‘мовах’, таких як JavaScript. Css, Html( мова розмітки ). В даному проекті замість звичайного Css використовувався Scss – це препроцесор, який значно спрощую розробку коду. Також крім чистого JS( який використовувався на бек енді ), використовувався фреймворк Angular, який містить все з ‘коробки’, тобто здатний структурувати важку логіку і є написаний на TypeScript( строго-типізований JS, який має велику перевагу, а саме, те що його легко відладжувати)



HTML (HyperText Markup Language, мова розмітки гіпертексту) − це стандарт верстки веб-сторінок, який визначає, які елементи і як саме повинні розташовуватися в документі. Браузери не відображають теги HTML, але застосовують їх для інтерпретації текстів вебсторінок. HTML може працювати з CSS та JS, а отже і з React, в якого головна задача правильно і точно перемалювати вибраний контент при змінні вхідних даних до компонента.

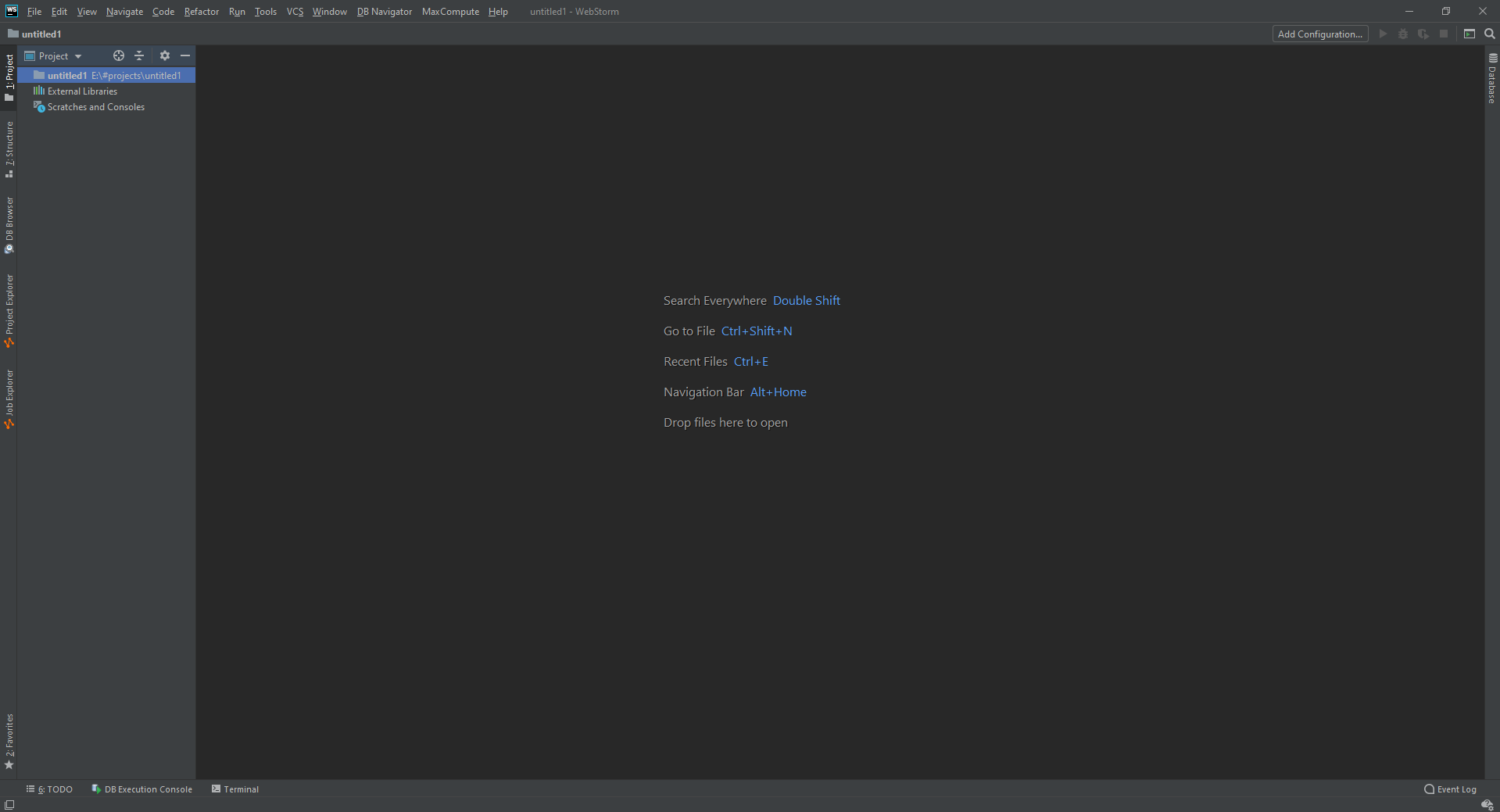
CSS (Cascading Style Sheets, каскадні таблиці стилів) − це набір параметрів форматування, який застосовується до елементів документа, щоб змінити їх зовнішній вигляд. Це дає змогу окремо кожній компоненті надавати стилізацію, в потрібний проміжок часу її змінювати.

React Bootstrap це, мабуть, найпопулярніша реалізація Bootstrap-компонентів, створена за допомогою React. Для роботи з існуючими темами Bootstrap ці компоненти спираються тільки на таблиці стилів Bootstrap.

Для виконання інших завдань, пов’язаних з функціоналом веб-додатка було обрано мову програмування JavaScript та JavaScript бібліотеку React.

JavaScript – це мова програмування, що дозволяє зробити веб-сторінку інтерактивною, тобто такою що реагує на дії користувача. Послідовність інструкцій (що називається програмою, скриптом або сценарієм) виконується інтерпретатором, вбудованим в звичайний Web -браузер. Іншими словами, код програми вбудовується в HTML - документ і виконується на боці клієнта. Для виконання програми не потрібно навіть перезавантажувати Web-сторінку, всі програми виконуються в відповідь на будь-яку подію. Наприклад, перед відправленням даних форми можна перевірити їх на допустимі значення і, якщо значення не відповідають очікуваним, заборонити відправлення даних. JavaScript - об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript. Його зазвичай використовують, як вбудовану мову для програмного доступу до об'єктів. Найбільш широке застосування знаходить у браузерах як мова сценаріїв для надання інтерактивності веб-сторінкам.

Для написання коду використовувався такий програмний продукт як WebShtorm



**2.2. Проектування логіки, архітектури та поведінки системи**

Розробка веб-додатку дуже виснажливий процес, який займає у розробника багато часу, а особливо, якщо він працює сам. Проаналізувавши задачу та методи її вирішення, було вирішено створювати програму на основі веб-інтерфейсу. Одна з найбільших переваг даного веб-інтерфейса є те, що він універсальний( повністю адаптивний на всіх пристроях, тобто можемо відкрити в любому розширенні не втрачаючи читаємість і контент )

Для реалізації задачі було вирішено використовувати базу даних PostGre,

Через свій зручний інтерфейс( pgAdmin ) та встроєний sql shell, також він має чудово документацію, та має багато задокументовано різнтх прикладів використання, що означає, що він є співмістимим з більшістю сучасних технологій.

. Як протокол застосовувавася REST (скор. англ. Representational State Transfer, «передача репрезентативного стану») — підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів. Був описаний і популяризований 2000 року Роєм Філдінгом, одним із творців протоколу HTTP. В основі REST закладено принципи функціонування Всесвітньої павутини і, зокрема, можливості HTTP. Філдінг розробив REST паралельно з HTTP 1.1 базуючись на попередньому протоколі HTTP 1.0.

Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад, HTML, XML, JSON). Будь-який REST протокол (HTTP в тому числі) повинен підтримувати кешування, не повинен залежати від мережевого прошарку, не повинен зберігати інформації про стан між парами «запит-відповідь». Стверджується, що такий підхід забезпечує масштабовність системи і дозволяє їй еволюціонувати з новими вимогами.

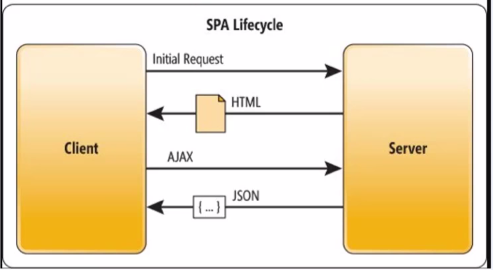


Рисунок – Діаграма роботи SPA( взаємодія між сервером та клієнтом)

Для початку було вирішено статично представити структури моделі у вигляді діаграми класів. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На ній показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини (див. рисунок 2.5)

При проектуванні архітектури веб-застосунку потрібно [20] дотримуватися наступних основних принципів, які допоможуть створити її ефективною та звезти до мінімуму вимоги і витрати до технічного обслуговування, а також сприяє зручності використання:

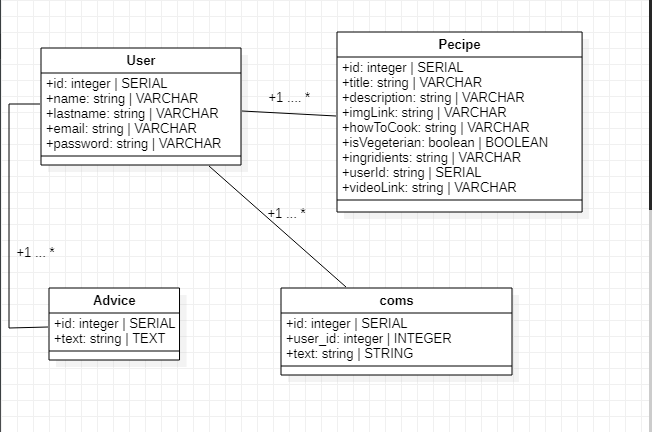
* поділ задачі - поділити веб додаток на певні модулі, з невеликим перекриттям в функціональності. Важливим фактором є мінімізація точок взаємодії для досягнення високої згуртованості і низького зчеплення. Тим не менше, поділ функціональних можливостей в неправильних межах може привести до сильного зв'язку і складності між функціями, навіть якщо функціональні можливості, що містяться в функції не призводить до істотного перекриття;
* одиничний принцип відповідальності - кожен компонент або модуль повинен відповідати тільки за одну конкретну функцію або функціональність, або агрегації згуртованої функціональності.
* принцип найменшого знання - компонент або об'єкт не повинен знати про внутрішні деталі інших компонентів або об'єктів.
* принцип неповторюваності - необхідність вказати намір в одному місці. Наприклад, з точки зору розробки веб додатків, конкретні функціональні можливості повинні бути реалізовані тільки в одному компоненті; функціональні можливості не мають бути продубльовані в будь-якому іншому компоненті;
* принцип зведення до мінімуму авансової розробки - розробляти тільки необхідні функціональні можливості. В інших випадках, особливо для швидкої розробки, необхідно уникати великої авансової розробки. Якщо ваші вимоги до сайту не ясні, або якщо є можливість розвитку з часом, слід уникати великих передчасних конструкторських робіт.

При розробці веб-додатку, метою створення архітектури сайту є зведення до мінімуму складності шляхом відділення структури в різних проблемних областях. Наприклад, призначений для користувача інтерфейс, доступ до даних - все це представлення різних проблемних областей. У кожній області, компоненти дизайну повинні зосередитися на конкретній області, і не слід змішувати код з інших областей. Наприклад, на сторінці деталів обраного фільму, не має бути коду дизайну головної сторінки та ін.

При проектуванні архітектури веб-застосунку потрібно [20] дотримуватися наступних основних принципів, які допоможуть створити її ефективною та звезти до мінімуму вимоги і витрати до технічного обслуговування, а також сприяє зручності використання:

* поділ задачі - поділити веб додаток на певні модулі, з невеликим перекриттям в функціональності. Важливим фактором є мінімізація точок взаємодії для досягнення високої згуртованості і низького зчеплення. Тим не менше, поділ функціональних можливостей в неправильних межах може привести до сильного зв'язку і складності між функціями, навіть якщо функціональні можливості, що містяться в функції не призводить до істотного перекриття;
* одиничний принцип відповідальності - кожен компонент або модуль повинен відповідати тільки за одну конкретну функцію або функціональність, або агрегації згуртованої функціональності.
* принцип найменшого знання - компонент або об'єкт не повинен знати про внутрішні деталі інших компонентів або об'єктів.
* принцип неповторюваності - необхідність вказати намір в одному місці. Наприклад, з точки зору розробки веб додатків, конкретні функціональні можливості повинні бути реалізовані тільки в одному компоненті; функціональні можливості не мають бути продубльовані в будь-якому іншому компоненті;
* принцип зведення до мінімуму авансової розробки - розробляти тільки необхідні функціональні можливості. В інших випадках, особливо для швидкої розробки, необхідно уникати великої авансової розробки. Якщо ваші вимоги до сайту не ясні, або якщо є можливість розвитку з часом, слід уникати великих передчасних конструкторських робіт.

При розробці веб-додатку, метою створення архітектури сайту є зведення до мінімуму складності шляхом відділення структури в різних проблемних областях. Наприклад, призначений для користувача інтерфейс, доступ до даних - все це представлення різних проблемних областей. У кожній області, компоненти дизайну повинні зосередитися на конкретній області, і не слід змішувати код з інших областей. Наприклад, на сторінці деталів обраного фільму, не має бути коду дизайну головної сторінки та ін.



UML діграма класів

Провівши ієрархічний аналіз вимог до програмного забезпечення, було виділено пакети, що відповідають підсистемам, які будуть використовуватися для представлення користувацького інтерфейсу



Рисунок Декомпозиції



Рисунок Декомпозиція роботи «Обслуговування користувачів кулінарної книги»

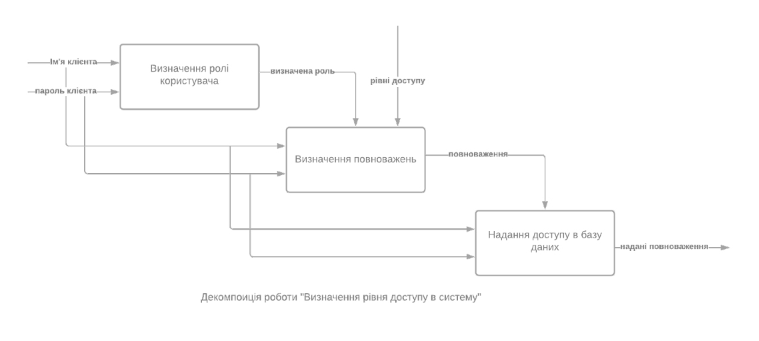


Рисунок Декомпозиція роботи «Визначення рівня доступу в систему»

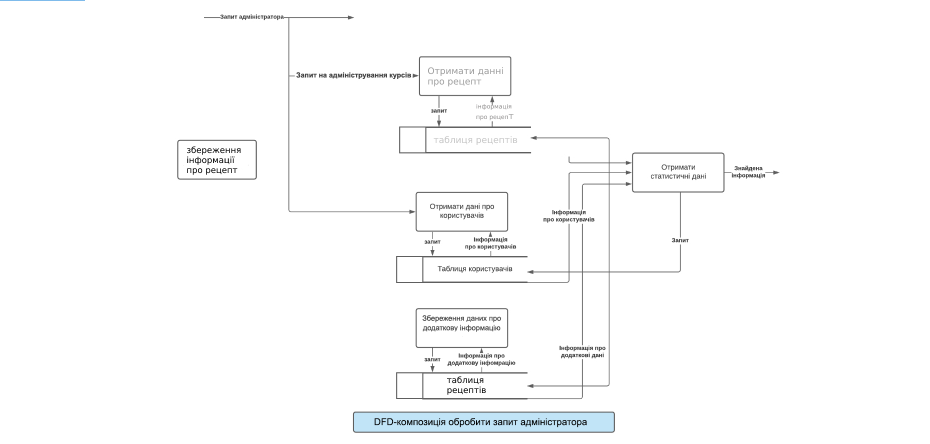


Рисунок Декомпозиція повноважень адміністратора

Висновок до другого розділу

В другому розділі було вибрано засоби для вирішення задач, створення Бази Даних – PostGre, мови програмування JS, фреймворк для фронтенда Angular, програма для відображення StarUml. Спроектовано логіку, архітектуру та поведінку системи.