

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ  
ОДЕСЬКОГО НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО  
УНІВЕРСИТЕТУ»**

*Спеціальність: 121 «Інженерія програмного забезпечення»*

*Освітня програма: «Розробка програмного забезпечення»*

*Група: 4РП-05*

# **Дипломний проект**

**здобувача освіти денної форми навчання  
РП.05.22.000.ДП**

***ТИМОЩУК  
ВОЛОДИМИР  
ОЛЕГОВИЧ***

**м. Одеса  
2022 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОДЕСЬКОГО  
НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-05

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

### Побудова серверної частини для інтернет-магазину з використанням технологій мікросервісної архітектури

Проектний матеріал складається з пояснювальної записки на 44 сторінках та графічного (презентаційного) матеріалу на 10 аркушах (слайдах).

Дипломник \_\_\_\_\_ (Тимошук В.О.)

Керівник \_\_\_\_\_ (Медведев А.О.)

#### Консультанти:

з економічної частини \_\_\_\_\_ (Копайгородська Т.Г. )

з охорони праці \_\_\_\_\_ ( Чорновол Н.І. )

з дотримання вимог ЄСКД \_\_\_\_\_ (Петрашова В.І.)

старший консультант \_\_\_\_\_ (Скорнякова О.В. )

#### До захисту допущений

Голова циклової комісії \_\_\_\_\_ (Скорнякова О.В. )

Завідувач відділення \_\_\_\_\_ (Суліма Ю.Ю.)

Захист «    » \_\_\_\_\_ 2022 р.                      Протокол ДКК № \_\_\_\_\_

Оцінка ДКК \_\_\_\_\_

Секретар ДКК \_\_\_\_\_

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОДЕСЬКОГО**  
**НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»**

Відділення комп'ютерних систем Комісія КТ та III  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**

**на дипломний проект (роботу)**

Здобувачеві (здобувачці) освіти Тимощук Володимир Олегович  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Побудова серверної частини для інтернет-магазину  
з використанням технологій мікросервісної архітектури

затверджена наказом по коледжу від “ 30 ” грудня \_\_\_\_\_ 202 1 р. № 306-A2-ОД

2. Термін задачі закінченого проекту (роботи) \_\_\_\_\_

3. Вихідні данні до проекту (роботи) Microsoft Visual Studio, .Net, C#, MS SQL Server, HTML  
JavaScript, Docker, Swagger, API, HTTP

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)  
1. Серверне представлення структури мікросервісної архітектури. 2. Економічний  
розрахунок. 3. Охорона праці.

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

Презентація (10 слайдів)

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний	Медведев А.О.		
Економічний	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Скорнякова О.В.		

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

#### КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Розділ 1. Серверне представлення структури мікросервісної архітектури		
2	Розділ 2. Економічний розрахунок		
3	Розділ 3. Охорона праці		
4	Розробка презентації до дипломної роботи		
5	Чистове оформлення пояснювальної записки		
6	Підготовка доповіді до захисту		
7	Отримання рецензії, відповіді на зауваження рецензента		
8	Захист роботи		

Дипломник \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)



## ЗМІСТ

ВСТУП .....	7
1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ. СЕРВЕРНЕ ПРЕДСТАВЛЕННЯ СТРУКТУРИ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ. ОБМІН ДАНИМИ МІЖ СЕРВІСАМИ.....	8
1.1 Представлення мікросервісної архітектури серед ASP.NET .....	8
1.2 Структура обміну даними між сервісами. Мости Owin. ....	11
1.3 Докер файли для зв'язку між контейнером та даними контейнера .....	15
1.4 Опис підключення для отримання даних .....	17
1.5 Технологія контейнеризації .....	18
1.6 Реалізація мікросервісної архітектури.....	19
1.6.1 Зв'язки контейнерів та запити у коді .....	19
1.6.2 Виведення даних на API – запит .....	20
1.6.3 Документація та структура системи .....	22
2 ЕКОНОМІЧНИЙ РОЗРАХУНОК .....	30
3 ОХОРОНА ПРАЦІ .....	36
ВИСНОВКИ .....	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	44

					РП 05.22.000 ДП ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

У сучасному світі при розробці програмного забезпечення дедалі частіше виникає ситуація, в якій вимоги до системи змінюються вже на кінцевих етапах розробки. До того ж, спостерігається постійний приріст функціональності у існуючих системах. Крім цього, до сучасного ПЗ пред'являються вкрай високі вимоги щодо продуктивності та відмовостійкості. Класична монолітна архітектура додатків не відповідала вимогам сучасного світу і на зміну їй прийшов сервісно-орієнтований підхід. Такий підхід має значні переваги в порівнянні з монолітними програмами, такі, наприклад, як краща масштабованість, менша зв'язаність між модулями, кращий контроль на етапах розробки, тестування та розгортання. Системи, побудовані за допомогою сервісно-орієнтованої архітектури, також розподілені. Взаємодія між сервісами здійснюється із використанням різних протоколів. Мікросервісна архітектура є одним із видів сервісно-орієнтованої архітектури. Це означає, що основний архітектурний компонент, з якого будується система, це сервіс. На жаль, ціною вищевказаних переваг є значне ускладнення системи. Потрібно підтримувати контракти протоколів, боротися з недоступністю послуг, управляти розподіленими транзакціями. Також до цих проблем відноситься необхідність розгортання великої кількості сервісів у різних оточення на всіх етапах розробки.

**Мета роботи** – розробити робочий проект з використанням мікросервісної архітектури на базі технологій .Net.

					РП 05.22.000 ДП ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

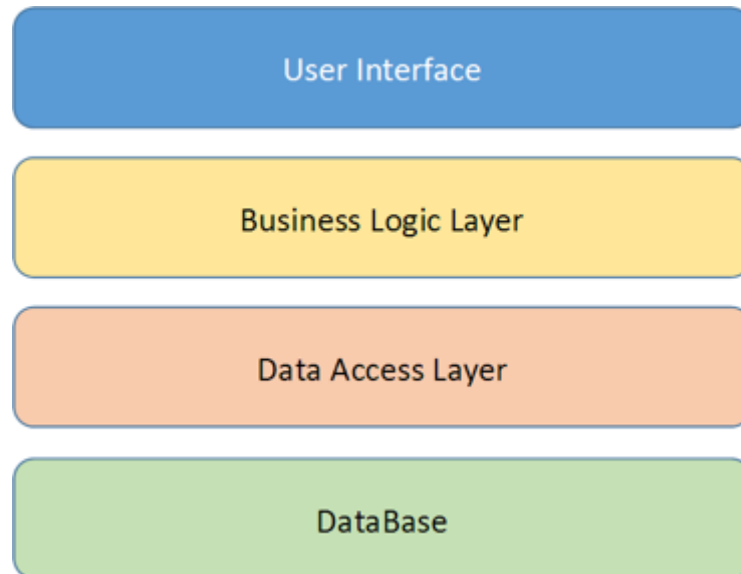
# 1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ.

## СЕРВЕРНЕ ПРЕДСТАВЛЕННЯ СТРУКТУРИ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ. ОБМІН ДАНИМИ МІЖ СЕРВІСАМИ

### 1.1 Представлення мікросервісної архітектури серед ASP.NET

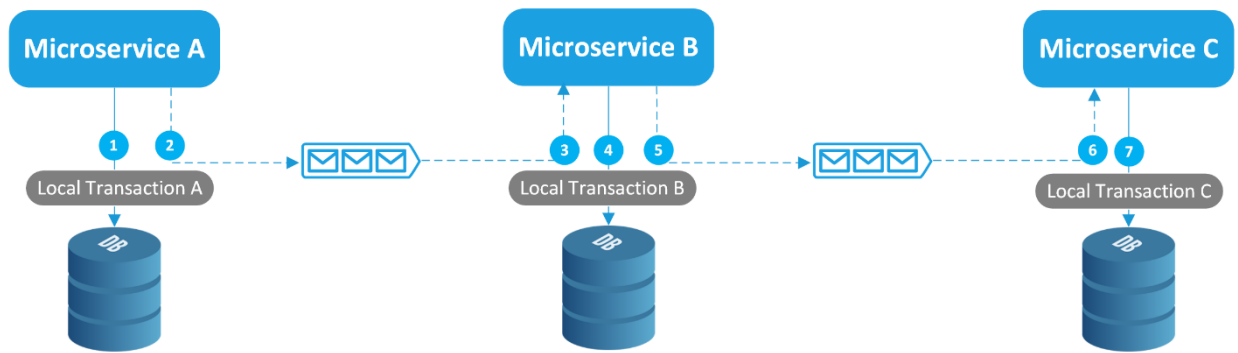
Мікросервісна архітектура - альтернатива моноліту, який передбачає єдине розгортання всього додатка. Хоча монолітну архітектуру легше реалізувати, пізніше це створить проблеми масштабованості, уповільнить процес розробки нових функцій, а помилка в модулі може зруйнувати весь додаток. У матеріалі розберемо переваги та недоліки мікросервісної архітектури, а також подивимося, які мови програмування найбільше підходять для мікросервісів.

Класична монолітна програма зазвичай має стандартну структуру Інтерфейс -> Бізнес-логіка -> Дані.



**Рисунок 1.1 – Структура монолітної програми**

Мікросервіси ж відштовхуються від бізнес-логіки:



**Рисунок 1.2 – Структура мікросервісної архітектури**

Мікросервісна архітектура - це варіант архітектури ПЗ, в основі якої лежать мікросервіси та їх взаємодія.

Мікросервіси – невеликі, слабо пов'язані модулі, які можна легко змінити.

Основний принцип архітектури — створення додатка, у якому компоненти його бізнес-логіки діляться на невеликі сервіси, які можна розгортати і використовувати незалежно друг від друга. Така архітектура дає командам оптимізувати випуск релізів.

Перевага використання мікросервісів:

- Модульність. За конкретні операції відповідають окремі частини програми.
- Надійність. При відмові одного з мікросервісів порушується робота тільки тих функцій, за які він відповідає. Це не зможе пошкодити програму повністю.
- Змінність. Можна змінити мікросервіс незалежно від інших сервісів системи.
- Масштабованість. Можна будь-якої миті додати або видалити модуль. Якщо мікросервіс досягне граничного навантаження, можна буде розгортати нові екземпляри сервісу у сусідньому кластері.

					РП 05.22.001 ДП ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- Тестування. Нові функції можна тестувати одразу у продакшені. Досить впровадити їх для деяких сегментів користувачів, щоб перевірити, як вони працюватимуть.

- Незалежна робота команд. Команди, що працюють із мікросервісами, можуть працювати без прив'язки до інших команд. Це дозволяє швидше реалізовувати та постачати нові функції.

- Більше уваги до якості. Кожен мікросервіс відповідає за певні бізнес-завдання. Команди, які працюють над мікросервісами прагнуть дати найкращий результат, тому що розуміють, що на цьому етапі він залежить тільки від них.

Зазвичай мікросервісна архітектура застосовується як один із варіантів масштабування програми. Усього таких варіантів може бути три:

- Sharding («розбиття» або просто «шардинг») – дані та інструменти для доступу до них розміщуються на різних вузлах.

- Mirroring (створення дзеркал) – дублювання всіх даних по безлічі однакових вузлів.

Власне, мікросервіси – функціональність розбита по бізнес-завданнях, кожен сервіс може бути створений своїми засобами розробки.

До сервісів даних архітектури обов'язково необхідно віднести запити на багаторівневу архітектуру. Багаторівнева архітектура АВР виглядає так:

					РП 05.22.001 ДП ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

## Layers

Presentation

View Models (Javascript)

Views (HTML/CSS)

Localization

Web

Web API Controllers

MVC Controllers

Application

Application Services

DTOs

DTO Mappers

Domain

Entities

Value Objects

Repositories

Business Classes

Domain Services

Unit of Work

Domain Events

Infrastructure

ORM

DB Migrations

## Others

Server Side

Dependency Injection

Logging

**Рисунок 1.3 – Структура багаторівневої серверної архітектури**

### **1.2 Структура обміну даними між сервісами. Мости Owin.**

Контейнери Docker – це навмисно ізольовані середовища. Кожен контейнер має свою власну файловою системою, до якої не можуть отримати доступ інші контейнери або ваш хост.

Іноді контейнерам може знадобитися обмін даними. Хоча ви повинні прагнути, щоб контейнери були самодостатніми, існують сценарії, в яких спільне використання даних неминуче. Це може бути так, щоб другий контейнер міг отримати доступ до комбінованого кешу, використовувати базу даних з файловою підтримкою, створити резервну копію або виконувати операції з даними користувача, такими як контейнер оптимізатора зображень, який обробляє фотографії профілю, завантажені через окремий контейнер веб-сервера .

Розглянемо кілька способів передачі даних між вашими контейнерами Docker.

Це забезпечує механізм автоматичного монтування томів, які вже використовуються наявним контейнером:

					РП 05.22.001 ДП ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

```
PM> docker run -d --volumes-from example --name backup backup-image: latest
```

### Рисунок 1.4 – Запуск даних у пакет контейнера

На цей раз резервний контейнер отримує том із загальними даними, змонтований у його каталог/data. Прапор `--volumes-from` отримує всі визначення томів, прикріплені наприклад контейнера. Він особливо підходить для завдань резервного копіювання та інших недовговічних контейнерів, які діють як допоміжні компоненти для основної служби.

Як правило, контейнери Docker є ефемерними і виконуються рівно стільки, скільки потрібно для виконання команди, виданої в контейнері. За промовчанням будь-які дані, створені всередині контейнера, доступні лише з контейнера і лише під час роботи контейнера.

Тома Docker можна використовувати для обміну файлами між хост-системою та контейнером Docker. Наприклад, припустимо, у нашому випадку проекту використовувати офіційний образ Docker Nginx та зберегти постійну копію файлів журналу Nginx для подальшого аналізу. За промовчанням образ nginx Docker буде реєструватися в каталозі `/var/log/nginx` всередині контейнера Docker Nginx. Зазвичай він недоступний із файлової системи хоста.

Наступна команда створить каталог з ім'ям `nginxlogs` у домашньому каталозі поточного користувача та зв'яже його з `/var/log/nginx` у контейнері:

```
PM> docker run --name=nginx -d -v ~/nginxlogs:/var/log/nginx -p 5000:80 nginx
```

### Рисунок 1.5 – Отримання даних за каталогом порту 5000:80

Де:

- `--name=nginx` дає ім'я контейнеру, щоб легше посилатися на нього.
- `-d` запускає процес у фоновому режимі. В іншому випадку ми просто побачили б порожню підказку Nginx і не змогли б використовувати цей термінал, поки не вб'ємо Nginx.
- `-v ~/nginxlogs:/var/log/nginx` пов'язує каталог `/var/log/nginx` із контейнера Nginx з каталогом `~/nginxlogs` на хост-машині. Docker

					РП 05.22.001 ДП ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

використовує а : для поділу шляху хоста від шляху контейнера і шлях хоста завжди стоїть на першому місці.

- -p 5000:80 настраює порт форвардинг. Контейнер Nginx прослуховує порт 80. Цей прапор зіставляє порт 80 контейнера з портом 5000 в хост-системі.

- nginx вказує, що контейнер повинен бути побудований з образу Nginx, який видає команду nginx-g "daemon off" для запуску Nginx.

Примітка: Прапор -v дуже гнучкий. Він може прив'язати або створити том у разі невеликого коригування синтаксису. Якщо перший аргумент починається з/або ~/, то створиться прив'язка.

-v /path:/path/in/container монтує каталог хоста, /path в /path/in/container

-v path:/path/in/container створює том з ім'ям path без зв'язку з хостом.

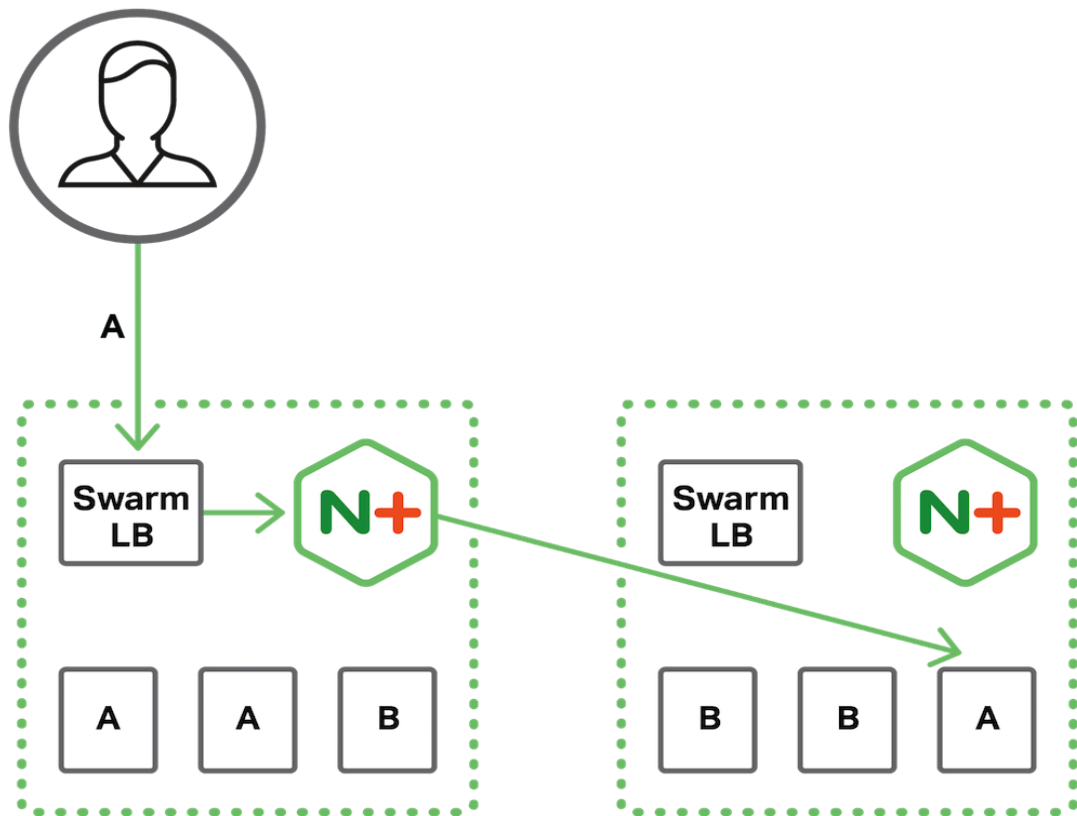


Рисунок 1.6 – Застосування інструменту Nginx у контейнері

OWIN визначає стандартний інтерфейс між веб-серверами .NET та веб-додатками. Метою інтерфейсу OWIN є поділ сервера та програми, заохочення розробки простих модулів для веб-розробки .NET та, будучи відкритим

стандартом, стимулювання екосистеми з відкритим вихідним кодом інструментів веб-розробки .NET.

```
// This method gets called by the runtime. Use this method to add services to the container.  
// For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?linkID=398940  
  
public void ConfigureServices(IServiceCollection services)  
{  
    :  
    services.AddOcelot().AddCacheManager(settings => settings.WithDictionaryHandle());  
}
```

### Рисунок 1.7 – Налаштування оселота за запитом на API OWIN

OWIN визначається як стандартний інтерфейс між веб-серверами .NET та веб-додатками. Це проект із відкритим вихідним кодом, що належить спільноті. Це просто специфікація, а чи не реалізація. Розроблений Microsoft проект "Катана" знаходиться на вершині OWIN. Раніше веб-програми .NET дуже тісно інтегрувалися з IIS. Він спрямований на відкритість та відкритий вихідний код для роз'єднаної залежності IIS. Це буде підтримувати проектування та розробку програм .NET у середовищі, відмінному від Корпорації Майкрософт. OWIN не тільки відокремлює веб-сервери від .NET, але й пропонує підтримку таких функцій, як саморозміщення та аутентифікація.

					РП 05.22.001 ДП ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

"Routes": [
  //Catalog API
  {
    "DownstreamPathTemplate": "/api/v1/Catalog",
    "DownstreamScheme": "http",
    "DownstreamHostAndPorts": [
      {
        "Host": "localhost",
        "Port": "8000"
      }
    ],
    "UpstreamPathTemplate": "/Catalog",
    "UpstreamHttpMethod": [ "GET", "POST", "PUT" ],
    "FileCacheOptions": { "TtlSeconds": 30 }
  },
  {
    "DownstreamPathTemplate": "/api/v1/Catalog/{id}",
    "DownstreamScheme": "http",
    "DownstreamHostAndPorts": [
      {
        "Host": "localhost",
        "Port": "8000"
      }
    ],
    "UpstreamPathTemplate": "/Catalog/{id}",
    "UpstreamHttpMethod": [ "GET", "DELETE" ]
  },
  {
    "DownstreamPathTemplate": "/api/v1/Catalog/GetProductByCategory/{category}",
    "DownstreamScheme": "http",
    "DownstreamHostAndPorts": [
      {
        "Host": "localhost",
        "Port": "8000"
      }
    ],
    "UpstreamPathTemplate": "/Catalog/GetProductByCategory/{category}",
    "UpstreamHttpMethod": [ "GET" ]
  },
  ...
]

```

**Рисунок 1.8 – Реалізація у проекті протів для роутингу**

### 1.3 Докер файли для зв'язку між контейнером та даними контейнера

На початку практичного використання безперервної інтеграції було зазначено наявність кількох етапів при складанні. Найбільш яскравий приклад - це етап тестування. У складання може бути безліч швидких тестів і невелика кількість довгих тестів, що охоплюють усю бізнес-логіку сервісу. Якщо

					РП 05.22.001 ДП ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

запускаються всі тести поспіль без огляду на час виконання, неможливо отримати швидкий зворотний зв'язок, коли тести виконуються з помилкою.

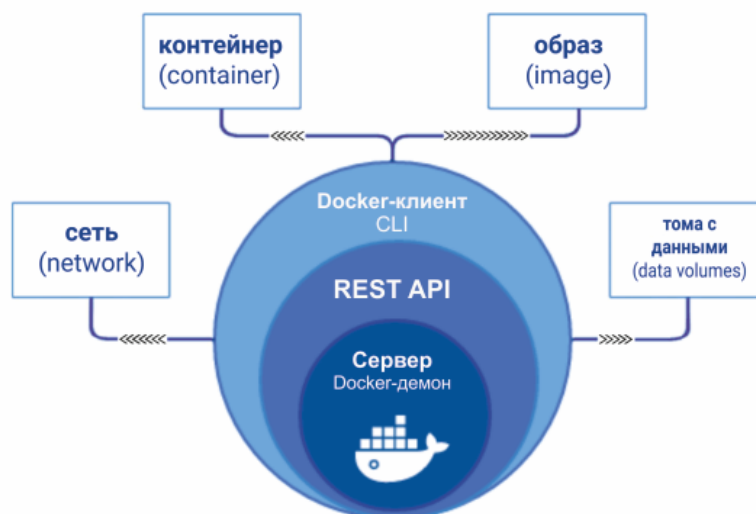
Для відповіді необхідно дочекатися, коли всі тести виконаються. Таким чином, якщо швидкий тест виконався з помилкою, потрібно дочекатися виконання всіх повільних тестів перед тим, як отримати результат. Вирішення цієї проблеми полягає в тому, щоб додати різні етапи тестування в процес складання. Один етап для швидких тестів, інший для повільних. Концепція етапів складання дає відмінну можливість спостерігати прогрес складання цілком, даючи зворотний зв'язок якості функціонування нового коду.

Процес безперервного постачання будується на концепції етапів збирання.

Безперервна поставка – це підхід, в якому є можливість постійно отримувати зворотний зв'язок про кожну зміну в коді і таким чином вважати, чи ця зміна є кандидатом у реліз, який буде випущений в промислову експлуатацію [5]. Щоб повністю охопити цю концепцію, потрібно змодельовати всі процеси від фіксування змін у коді до виходу складання в промислову експлуатацію. При безперервному постачанні розширюється ідея етапів складання, щоб повністю охопити процес, включаючи автоматичне та ручне тестування. На рисунку 1.9 зображено етапи складання.

					РП 05.22.001 ДП ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

## Компоненты Docker Engine



**Рисунок 1.9 – Контейнеризация образов у докер файлі**

Коли версія вихідного коду проходить етапи автоматичного тестування, вона потрапляє на етап ручного тестування (UAT). Це має бути відображено у спеціальних інструментах, які автоматизують процес безперервного постачання. Після моделювання кожного етапу, який починається з фіксацій зміни в кодї, а закінчується розгортанням на промислову експлуатацію можна точно судити як програмне забезпечення і значно знизити тимчасові витрати між різними релізами системи, оскільки є можливість централізованого спостереження за кожною збіркою. Таким чином, можна точково вносити удосконалення у процес розробки.

### 1.4 Опис підключення для отримання даних

Наскрізнї реалізації мікросервісів:

- Впровадження централізованого розподіленого ведення журналу за допомогою Elastic Stack (ELK); Elasticsearch, Logstash, Kibana та SeriLog для мікросервісів

					РП 05.22.001 ДП ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- Використання функції Перевірка працездатності в серверних ASP.NET мікрослужбах
- Використання даних в окремій службі, яка може відслідковувати працездатність та навантаження між службами, а також повідомляти про працездатність мікрослужб шляхом виконання запитів за допомогою перевірки АПІ шлюзу.

### 1.5 Технологія контейнеризації

Один з принципів мікросервісної архітектури говорить про те, що сервіс повинен бути ізольованим і автономним, повністю інкапсулюючи оточення виконання. Для дотримання цього принципу всі компоненти, такі як операційна система, середовище виконання та код мікросервісу, що виконується, повинен бути автономним і ізольованим. Єдиний спосіб досягти цього - слідувати підходу один мікросервіс на одну віртуальну машину. Однак це призведе до недостатньої утилізації ресурсів віртуальної машини.

Також у багатьох випадках через додаткові накладні витрати можуть звести всі переваги мікросервісів нанівець. Технологія контейнеризації далеко не нова і не новаторська технологія. Вона використовується досить тривалий час. Тим не менш, дана технологія стала набирати значної популярності з приходом хмарних технологій. Недоліки традиційних віртуальних машин стали каталізатором зростання популярності контейнерів. Постачальники інструментів для роботи з контейнерами, наприклад Docker, значно спростили технологію контейнеризації, що сприяло впровадженню даної технології в широкі маси. Популярність DevOps та мікросервісної архітектури також прискорила переродження технології контейнеризації.

Технологія контейнеризації надає приватне оточення в операційній системі. Ця технологія також називається віртуалізація операційної системи. У цьому підході ядро операційної системи надає ізольований віртуальний простір. Кожен з віртуальних просторів називається контейнером. Контейнери

					РП 05.22.001 ДП ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяють процесам створювати ізольоване оточення на операційній системі, що містить ці контейнери. На рисунку 1.10 зображені різні шари, залучені до процесу контейнеризації.

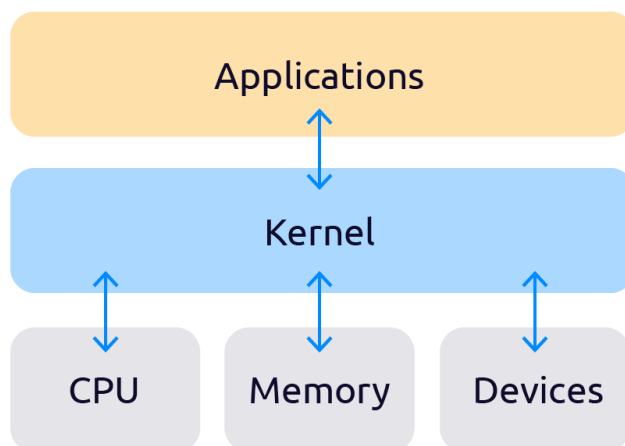


Рисунок 1.10 – Шари контейнеризації

## 1.6 Реалізація мікросервісної архітектури

### 1.6.1 Зв'язки контейнерів та запити у коді

Реалізуємо зв'язку для запиту на контейнер з кошиком на обробку запиту для отримання відомостей про збереження даних:

```
public class BasketService : IBasketService
{
    private readonly HttpClient _client;

    public BasketService(HttpClient client)
    {
        _client = client ?? throw new ArgumentNullException(nameof(client));
    }

    2 references | 0 changes | 0 authors, 0 changes
    public async Task<BasketModel> GetBasket(string userName)
    {
        var response = await _client.GetAsync($"/api/v1/Basket/{userName}");
        return await response.ReadContentAs<BasketModel>();
    }
}
```

Рисунок 1.11 – Клас кошика для отримання шлюзу на сервер

Передамо отриману відповідь на контролер для відпрацювання даних, що повертаються:

```
[HttpGet("{userName}", Name = "GetShopping")]
[ProducesResponseType(typeof(ShoppingModel), (int)HttpStatusCode.OK)]
0 references | 0 changes | 0 authors, 0 changes
public async Task<ActionResult<ShoppingModel>> GetShopping(string userName)
{
    var basket = await _basketService.GetBasket(userName);

    foreach (var item in basket.Items)
    {
        var product = await _catalogService.GetCatalog(item.ProductId);

        // set additional product fields
        item.ProductName = product.Name;
        item.Category = product.Category;
        item.Summary = product.Summary;
        item.Description = product.Description;
        item.ImageFile = product.ImageFile;
    }

    var orders = await _orderService.GetOrdersByUsername(userName);

    var shoppingModel = new ShoppingModel
    {
        UserName = userName,
        BasketWithProducts = basket,
        Orders = orders
    };

    return Ok(shoppingModel);
}
```

**Рисунок 1.12 – Отримання результату у контролері**

## 1.6.2 Виведення даних на API – запит

Мікросервіс api шлюзу Ocelot

- Реалізація шлюзів API за допомогою Ocelot
- Приклади мікрослужб/контейнерів для перенаправлення через шлюзи

API

- Запуск різних типів контейнерів шлюзу API/BFF

					РП 05.22.001 ДП ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

- Шаблон агрегування шлюзу в Shopping.Aggregator

Далі налаштуємо АПІ запит на сервер, для цього створимо OWIN request:

```

0 references | 0 changes | 0 authors, 0 changes
public static class HttpClientExtensions
{
    5 references | 0 changes | 0 authors, 0 changes
    public static async Task<T> ReadContentAs<T>(this HttpResponseMessage response)
    {
        if (!response.IsSuccessStatusCode)
            throw new ApplicationException($"Something went wrong calling the API: {response.ReasonPhrase}");

        var dataAsString = await response.Content.ReadAsStringAsync().ConfigureAwait(false);

        return JsonSerializer.Deserialize<T>(dataAsString, new JsonSerializerOptions { PropertyNameCaseInsensitive = true });
    }
}

```

### Рисунок 1.13 – Обробка контентного запиту на клієнт

Що стосується служби ідентифікації, у проекті вона не проходить через шлюзи АРІ, оскільки вона є спільною для всієї системи, але за допомогою Ocelot її також можна включити до списків перенаправлення.

Всі ці служби в даний час реалізовані як служби веб-АРІ ASP.NET Core, як видно з коду. Давайте зосередимося на одній із мікрослужб, наприклад в коді мікрослужби каталогу Catalog.

```

[HttpGet]
[Route("items/{id:int}")]
[ProducesResponseType((int)HttpStatusCode.BadRequest)]
[ProducesResponseType((int)HttpStatusCode.NotFound)]
[ProducesResponseType(typeof(CatalogItem),(int)HttpStatusCode.OK)]
public async Task<IActionResult> GetItemById(int id)
{
    if (id <= 0)
    {
        return BadRequest();
    }
    var item = await _catalogContext.CatalogItems.
        SingleOrDefaultAsync(ci => ci.Id == id);

    //...

    if (item != null)
    {
        return Ok(item);
    }
    return NotFound();
}

```

HTTP-запит у результаті запустить такий код С#, який отримує доступ до бази даних мікрослужби та виконує будь-яку необхідну дію.

						РП 05.22.001 ДП ПЗ	Арк.
							21
Змн.	Арк.	№ докум.	Підпис	Дата			

Якщо говорити про URL-адресу мікрослужби, при розгортанні контейнерів на локальному комп'ютері розробки (локальному вузлі Docker) кожен контейнер мікрослужби завжди має внутрішній порт (зазвичай це порт 80), вказаний у файлі Dockerfile, як показано нижче.

```
FROM mcr.microsoft.com/dotnet/aspnet:3.1 AS base
WORKDIR /app
EXPOSE 80
```

Порт 80, показаний у кодї, знаходиться всередині вузла Docker, тому клієнтські програми не мають доступу до нього.

Клієнтські програми мають доступ тільки до зовнішніх портів (якщо є), які публікуються при розгортанні за допомогою docker-compose.

Ці зовнішні порти не слід публікувати при розгортанні робочого середовища. Тому необхідно використовувати шлюз API, щоб уникнути прямого обміну даними між мікрослужбами та клієнтськими програмами.

Проте при розробці слід звертатися до мікрослужб або контейнерів безпосередньо і запускати їх через Swagger. Ось чому в eShopOnContainers зовнішні порти, як і раніше, вказуються навіть у тому випадку, якщо вони будуть використовуватися шлюзом API або клієнтськими програмами.

### 1.6.3 Документація та структура системи

UI частина веб-додатку виглядає наступним чином. На рисунку 1.14 зображена головна сторінка додатку, яка містить наступні елементи:

- Шапка(header)
- Баннер
- Картки з товарами
- Футер

Шапка виконує роль керування додатком. Вона має меню, де виконується редірект до наступних сторінок: Home (домашня сторінка), Product (товари магазину), Cart (корзина користувача), Order (інформація про

					РП 05.22.001 ДП ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

замовлення), Contact (сторінка з інформацією про зв'язок). Також на шапці додатку розташовано пошук товарів.

Баннер виконує інформативну роль. Він складається зі слайдеру та картки з найуспішнішим продуктом в інтернет-магазині (рис. 1.15). Якщо користувач натисне на кнопку «View», він потрапить на сторінку товару, де може ознайомитися з описом та замовити його.

Картки з товарами (рис. 1.16) складаються з двох компонентів (нові продукти та кращі продукти). Вони мають зображення товару, коротку інформацію, інформацію про ціну та кнопку «Add to cart». Натиснувши кнопку, користувач зможе переглянути товар у своєму кошику та замовити його.

					РП 05.22.001 ДП ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

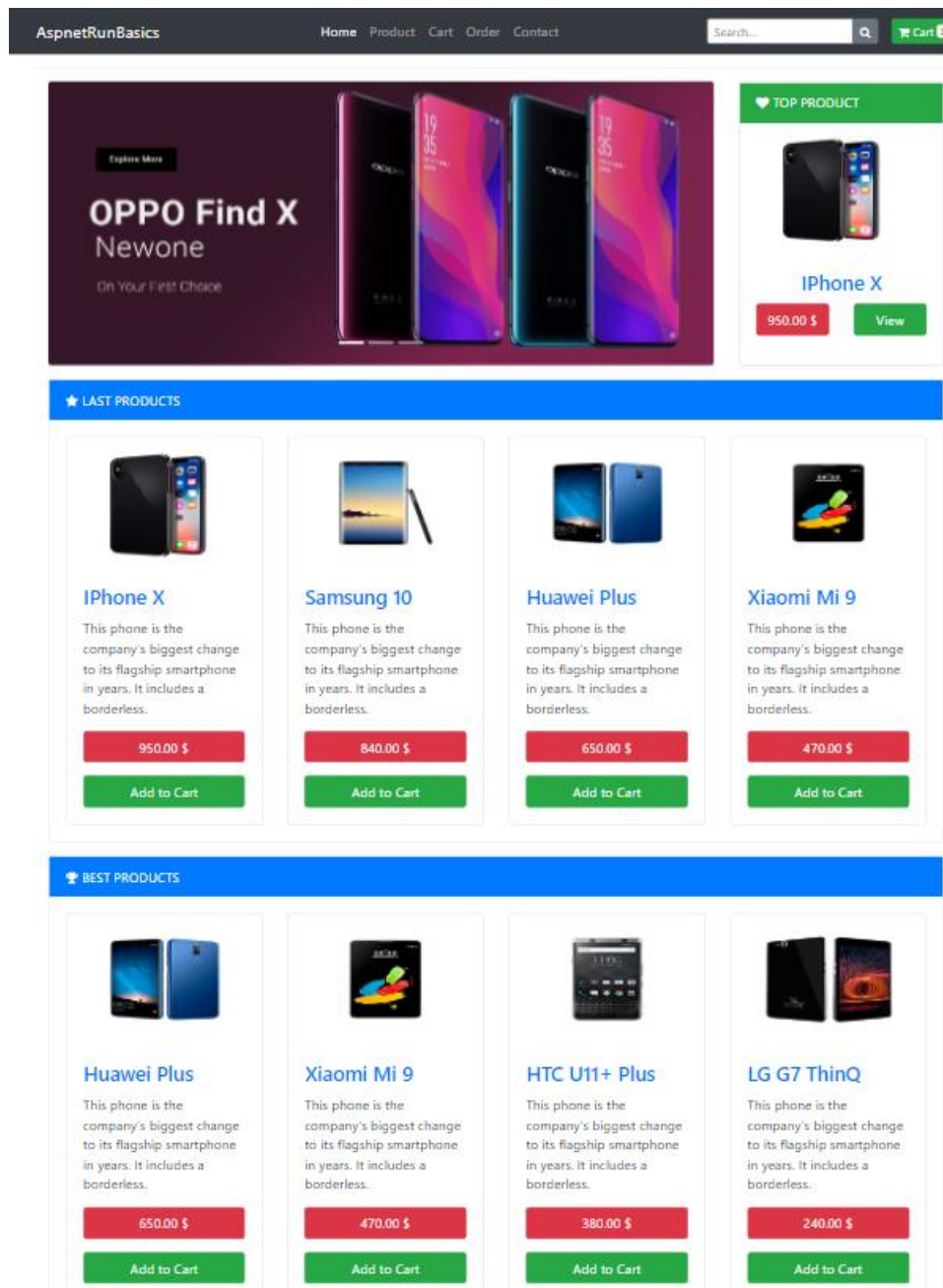


Рисунок 1.14 – UI веб-додатка

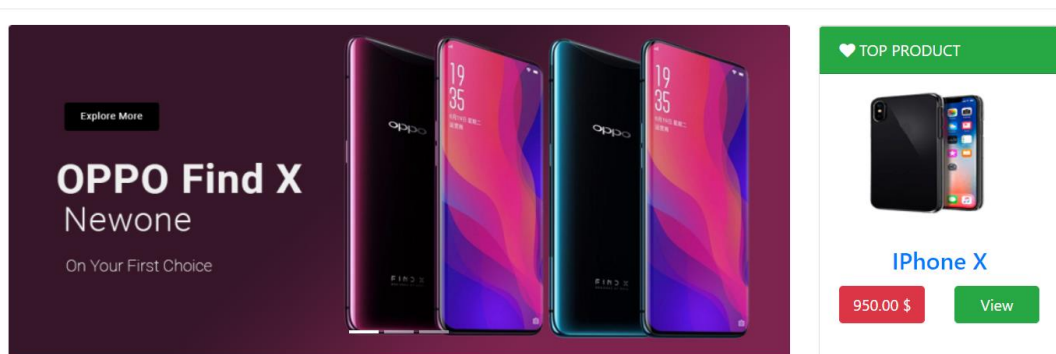
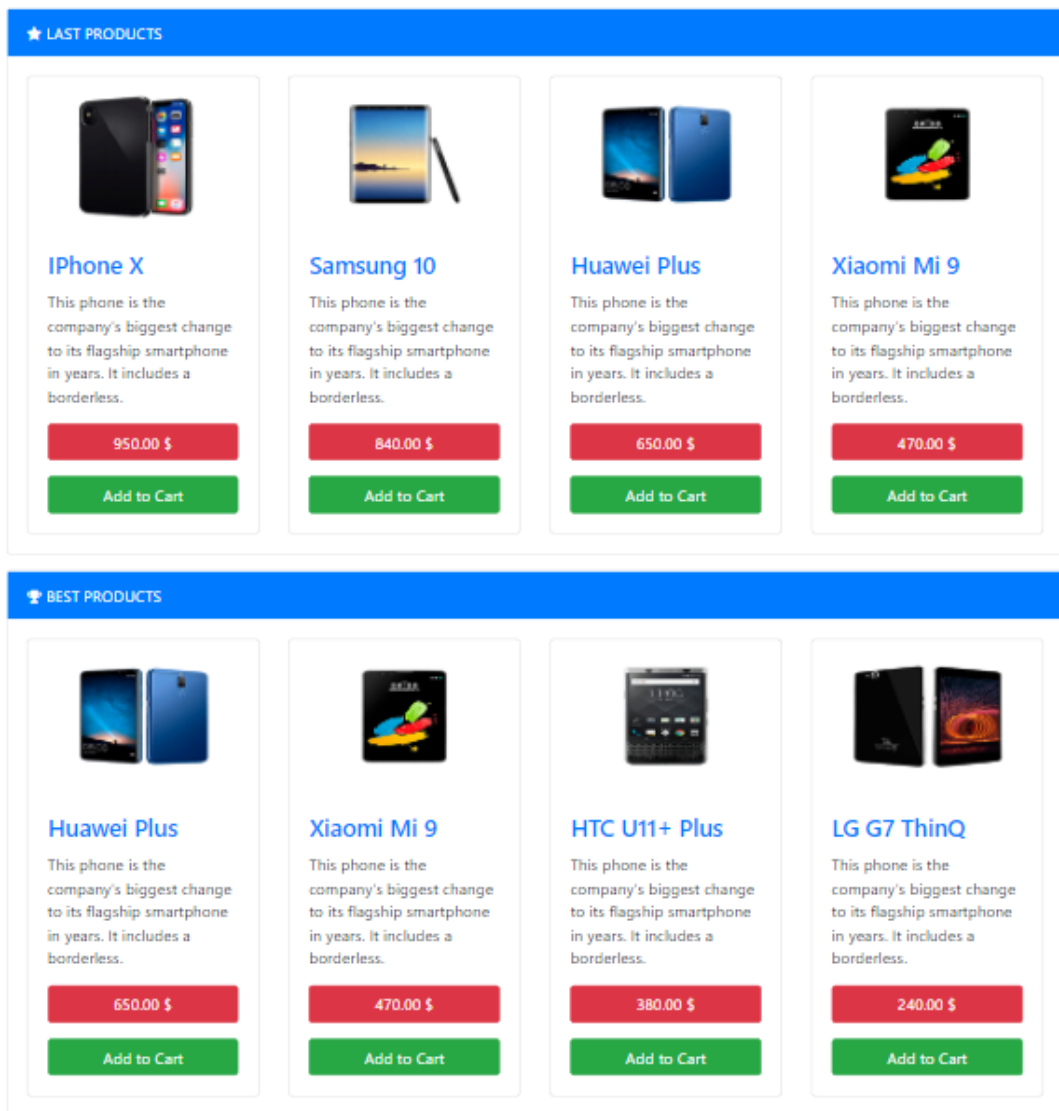


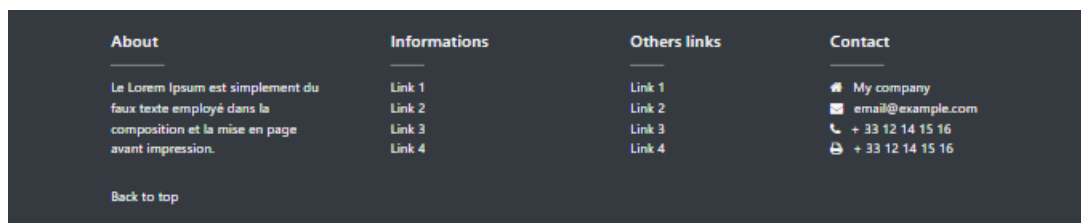
Рисунок 1.15 – Баннер головної сторінки

Змн.	Арк.	№ докум.	Підпис	Дата



**Рисунок 1.16 – Картки з товарами**

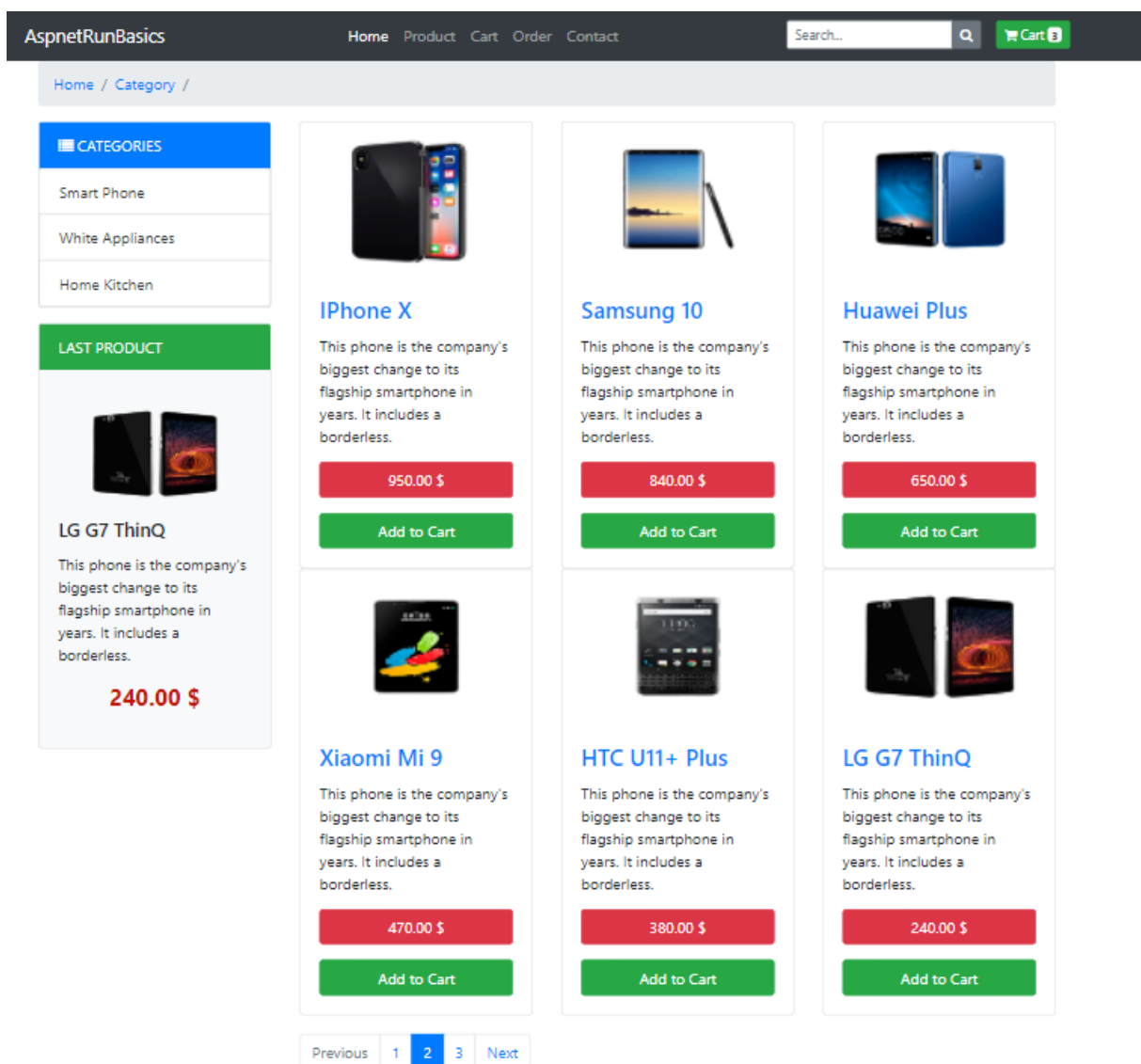
Футер виконує роль нижньої частини додатку, він має карту сайту та контактні дані(рис. 2.7)



**Рисунок 1.17 – Футер додатку**

Таким чином, головна сторінка додатку має просту, але зручну структуру для користувача.

Якщо користувач перейде до сторінки Product, він потрапить на сторінки зі всіма продуктами, які зберігаються у базі даних. UI цієї сторінки зображений на рис. 2.8.



**Рисунок 1.18 – Каталог з товарами**

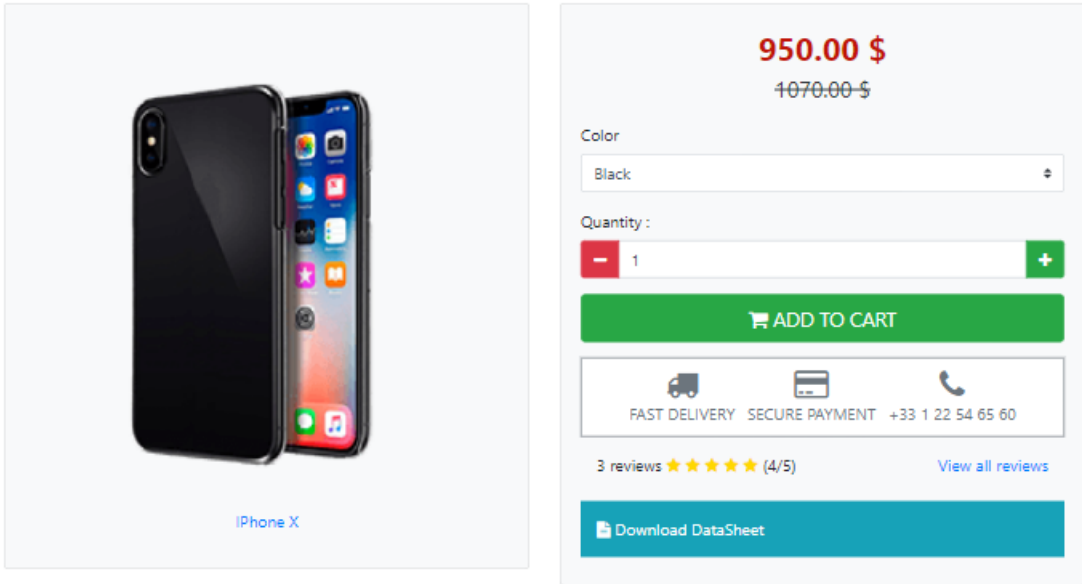
Каталог з товарами складається з наступних компонентів:

- Категорії. Користувач може обирати потрібні категорії, щоб побачити певні товари
- Картки з товарами, натиснувши на які користувач може переглянути повну інформацію про товар(рис. 2.9) або додати відразу у кошик.
- Інтерактивний елемент з пагінацією, який необхідний для того, щоб зручно було переглядати товари. На одній сторінці міститься 6 товарів, щоб переглянути далі, потрібно перейти на наступну сторінку.

					РП 05.22.001 ДП ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

AspnetRunBasics Home Product Cart Order Contact Search... Cart 0

Home / Product / iPhone X



950.00 \$  
1070.00 \$

Color  
Black

Quantity :  
- 1 +

ADD TO CART

FAST DELIVERY SECURE PAYMENT +33 1 22 54 65 60

3 reviews ★★★★★ (4/5) View all reviews

Download DataSheet

DESCRIPTION

This phone is the company's biggest change to its flagship smartphone in years. It includes a borderless.

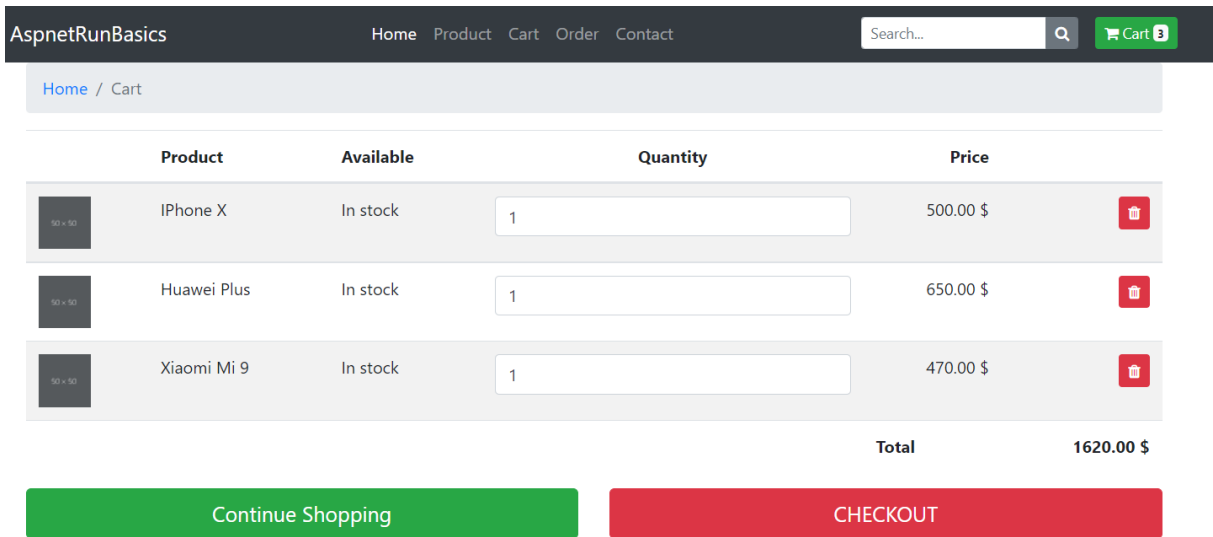
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ut, tenetur natus doloremque laborum quos iste ipsum rerum obcaecati impedit odit illo dolorum ab tempora nihil dicta earum fugiat. Temporibus, voluptatibus. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ut, tenetur natus doloremque laborum quos iste ipsum rerum obcaecati impedit odit illo dolorum ab tempora nihil dicta earum fugiat. Temporibus, voluptatibus.

### Рисунок 1.19 – Повна інформація про товар

Якщо користувач перейде на сторінку товару, він зможе дізнатися більше про певний товар, обрати кількість та замовити його.

Для того щоб переглянути товар у кошику, потрібно натиснути на кнопку Cart у головному меню. Сторінка кошика зображена на рис. 2.10. Користувач може побачити список з товарами на обраних двох варіантах продовження: «Continue shopping»(продовжити покупки) або «Checkout»(оформити замовлення). Продовження покупок поверне користувача до каталогу, а оформити замовлення направить користувача до форми, яку потрібно заповнити щоб отримати товар(рис. 2.11).

					РП 05.22.001 ДП ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		



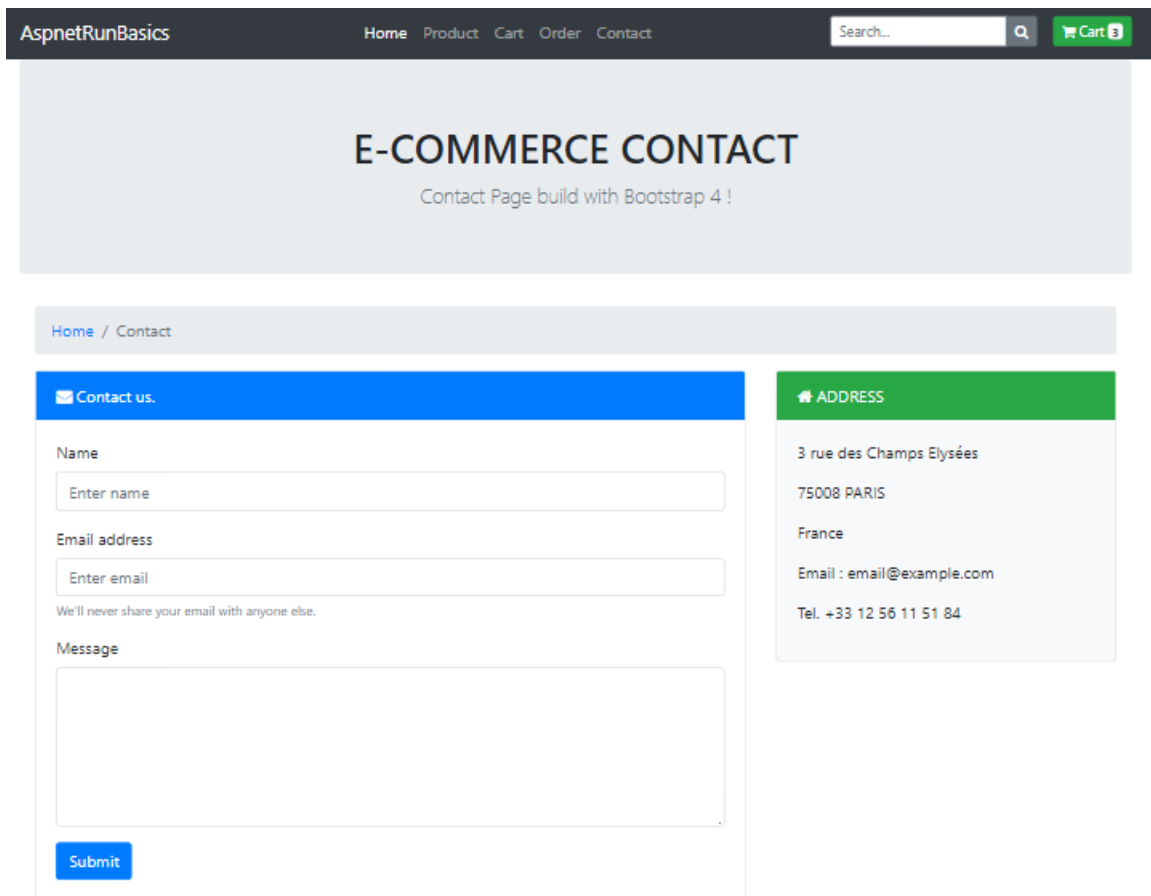
**Рисунок 1.20 – Кошик користувача**

The screenshot shows the checkout page with a breadcrumb 'Home / Cart / CheckOut'. It is divided into two main sections: 'Billing address' and 'Your cart'. The 'Billing address' section contains fields for First name, Last name, Username, Email (Optional), Address, Address 2 (Optional), Country, State, and Zip. There are checkboxes for 'Shipping address is the same as my billing address' and 'Save this information for next time'. The 'Payment' section has radio buttons for 'Credit card' (selected), 'Debit card', and 'Paypal'. Below are fields for Name on card, Credit card number, Expiration, and CV. The 'Your cart' section shows a list of items: iPhone X (\$500.00), Huawei Plus (\$650.00), and Xiaomi Mi 9 (\$470.00), with a 'Total (USD)' of \$1620.00. A 'Promo code' field and a 'Redeem' button are also present. A blue 'Checkout Order' button is at the bottom.

**Рисунок 1.21 – Форма для оформлення замовлення**

Сторінка «Contacts» має форму для відправки повідомлення до адміністратора та блок з контактними даними (рис. 1.22).

					РП 05.22.001 ДП ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		



### Рисунок 1.22 – Сторінка контактів

Таким чином, додаток з застосуванням мікросервісної архітектури протестовано, він працює та функціонує без помилок.

					РП 05.22.001 ДП ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ЕКОНОМІЧНИЙ РОЗРАХУНОК

Метою даних розрахунків є обчислення вартості виконання науково-дослідної роботи «Побудова серверної частини для інтернет-магазину з використанням технологій мікросервісної архітектури». Проведений аналіз показав, що для досягнення результату необхідно впровадити безперервні практики, такі як безперервна інтеграція та безперервне розгортання, а також перейти від традиційної віртуалізації до контейнеризації мікросервісів за допомогою механізму оркестрування контейнерів. Даний вид проекту відноситься до науково-дослідницької розробки. Оцінка якості розробленого проекту включає визначення трудомісткості і вартості його створення.

Розрахунок трудомісткості НДР здійснений в наступній послідовності:

1) Складений перелік всіх етапів і видів робіт, які необхідно виконати в ході даної НДР. Після узгодження з керівником проекту допущено виключення, доповнення, об'єднання окремих етапів і видів робіт;

2) По кожному виду робіт визначений кваліфікаційний рівень виконавців. Перелік етапів і робіт, що виконуються при проведенні НДР, приведений в таблиці 3.1.

### Розподіл робіт по етапах і видах виконавців.

Таблиця 3.1.

Етап проведення НДР	Вигляд робіт	Посада виконавця
Розробка технічного	1.Складання і затвердження ТЗ для НДР по розробці «Побудова серверної частини для інтернет-	Дипломник, керівник

					РП 05.22.002 ДП ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

завдання (ТЗ)	магазину з використанням технологій мікросервісної архітектури»	
Вибір напрямку дослідження	<p>1. Збір і вивчення науково-технічної літератури.</p> <p>2. Формулювання можливих напрямів вирішення завдань, поставлених в технічному завданні НДР і їх порівняльна оцінка.</p> <p>3. Вибір напрямку проведення досліджень</p> <p>4. Розробка плану проведення досліджень для подальшої розробки.</p>	Дипломник керівник
Теоретичні і експериментальні дослідження	<p>1. Серверне представлення структури мікросервісної архітектури. Обмін даними між сервісами</p> <p>2. Реалізація серверної архітектури</p>	Дипломник керівник консультанти
Узагальнення і оцінка результатів досліджень	<p>1. Узагальнення результатів</p> <p>2. Висновки.</p> <p>3. Складання і оформлення звіту.</p>	Дипломник керівник консультанти

*Оцінка тривалості виконання робіт* розраховується на основі вірогідних оцінок робіт, що задаються виконавцями.

**Очікувана трудомісткість робіт.**

Таблиця 3.2.

					РП 05.22.002 ДП ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Вигляд роботи	Очікуваний час виконання (дні)
1. Складання і затвердження ТЗ для НДР «Побудова серверної частини для інтернет-магазину з використанням технологій мікросервісної архітектури»	1
2. Збір і вивчення науково – технічної літератури, технічної документації і інших матеріалів.	2
3. Формулювання можливих напрямів вирішення завдань, поставлених в технічному завданні НДР і їх порівняльна оцінка.	2
4. Розробка плану проведення досліджень для подальшої розробки.	2
5. Серверне представлення структури мікросервісної архітектури. Обмін даними між сервісами	6
6. Реалізація серверної архітектури	8
Всього:	21

**Розрахунок собівартості і ціни виконання НДР.** Виходячи з особливостей створення науково – технічної продукції і її залежності від інтелектуальної праці, розрахунок собівартості і ціни виконання НДР включає наступні статті витрат: витрати на матеріали, основна і додаткова заробітна плата, відрахування до єдиного соціального фонду страхування, витрати на роботи, що виконуються сторонніми організаціями, і деякі інші.

1) Витрати на матеріали складають 190 грн. (Папір А4)

2) До витрат «Основна заробітна плата» відносяться оплата праці виконавців, безпосередньо притягнених до її виконання. Розмір основної зарплати встановлюється виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а

					РП 05.22.002 ДП ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

також їх середньої заробітної плати (ставки) за один робочий день. Відповідно до статті 8 «Закону про Державний бюджет України на 2021» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку – 39,26 грн.

Середня зарплата за один робочий день для кожного виконавця визначена по формулі:

$$Зден = п.т.с. * 8;$$

де п.т.с – погодинна тарифна ставка, грн.;

8 – тривалість робочого дня, год.

$$Зден дипломника = 39.26 * 8 = 314,08 \text{ грн.}$$

$$Зден керівника = 60.00 * 8 = 480 \text{ грн.}$$

$$Зден консультантів = 55.00 * 8 = 440 \text{ грн.}$$

Витрати на основну заробітну плату, НДР, що включаються в собівартість, приведені в таблиці 3.4.

### Витрати на основну заробітну плату.

Таблиця 3.4.

Виконавець	Погодинна тарифна ставка, грн	Денна ставка, грн	Трудомісткість у робочих днів	Сума основної зарплати, грн
Дипломник	40,00	320,00	21	6720,00
Керівник	60,00	480,00	1	480,00
Консультант по економічній частині	55,00	440,00	0,25	110,00

Консультант по охороні праці	55,00	440,0 0	0,25	110,00
Нормоконтроль	55,00	440,0 0	0,25	110,00
Всього (Зо)				7560,0 0

3) Витрати на додаткову заробітну плату визначаються у відсотках від основної. У наукових закладах додаткова заробітна плата складає 10-12% від основної заробітної плати.

$$Зд=10\% *Зо;$$

$$Зд= 0,1 *7560,00 =756,00 \text{ грн}$$

4) До складу собівартості НДР включаються податки, збори і інші обов'язкові платежі, встановлені системою оподаткування що діє. Відрахування до єдиного соціального внеску складає:

$$Зєсв=0,22*(Зо+Зд);$$

$$Зєсв= 0,22* (7560,00+756,00) = 1829,52 \text{ грн.}$$

5) До накладних витрат відносять витрати на управління і господарське обслуговування, що відноситься до всіх виконуваних НДР. У наукових закладах накладні витрати складають 40 -120% від основної і додаткової заробітної плати.

$$Рнакл= (Зо+Зд)*0,42;$$

$$Рнакл= (7560,00+756,00)*0.44 = 3659,04 \text{ грн.}$$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому НДР за формою, приведеною в таблиці 4.4.

### Калькуляція планової собівартості

Таблиця 4.4.

					РП 05.22.002 ДП ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

Статті витрат	Сума, грн.
1. Матеріали	190,00
2. Основна заробітна плата	7560,00
3. Додаткова заробітна плата	756,00
4. Відрахування до єдиного соціального внеску	1829,52
5. Накладні витрати	3659,04
Планова собівартість (Спл)	13994,56

Плановий прибуток визначений по формулі:

$$\text{Ппл} = 0,1 * \text{Спл} = 0,1 * 13994,56 = 1399,45 \text{ грн}$$

Де 0,1 – норматив, який враховує граничний рівень рентабельності, встановлений чинним законодавством для науково-технічної продукції.

Договірна ціна визначається по формулі

$$\text{Цнір} = \text{Спл} + \text{Ппл} = 13994,56 + 1399,45 = 15394,02 \text{ грн.}$$

Ціну реалізації встановлюємо з урахуванням ПДВ

$$\text{ПДВ} = 0,2 * \text{Цнір} = 0,2 * 15394,02 = 3078,80 \text{ грн.}$$

Звідси ціна реалізації становить:

$$\text{Цр} = \text{Цнір} + \text{ПДВ} \quad \text{Цр} = 15394,02 + 3078,80 = 18472,82 \text{ грн.}$$

					РП 05.22.002 ДП ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ОХОРОНА ПРАЦІ

#### Вступ

Соціальне значення охорони праці полягає у сприянні росту ефективності суспільного виробництва шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і профзахворювань і проявляється у зростанні продуктивності праці, збереженні трудових ресурсів та збільшенні сукупного національного продукту

Економічне значення охорони праці визначається ефективністю заходів з покращення умов і підвищення безпеки праці та є економічним виразом соціальної значущості охорони праці.

У результаті поліпшення умов праці нормалізується психологічний клімат в трудовому колективі, підвищується налагодженість в роботі, зростає продуктивність праці.

У розділі охорона праці дипломного проекту розглядаються шляхи створення безпечних умов праці для працівників.. До розгляду взято робоче місце програміста (оператора ЕОМ).

#### **3.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника.**

На користувачів ПК під час роботи з комп'ютерною технікою можуть діяти такі види небезпек: ураження електричним струмом; енергетична небезпека (виникає через коротке замикання: опіки, електрична дуга, викид розплавленого металу); небезпека загоряння; термонебезпека (дія високих температур через нагрівання конструктивних елементів); механічна небезпека (травми через падіння, дію рухомих частин, поріз за гострі частини конструктивних елементів); небезпека випромінювання (дія звукового (акустичного), високочастотного, інфрачервоного, ультрафіолетового й іонізуючого випромінювання,

					РП 05.22.003 ДП ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

## **3.2 Розробка заходів з охорони праці**

### **3.2.1 Виробничі приміщення**

Об'ємно-планувальні рішення будівель та приміщень для роботи з ПК мають відповідати вимогам ДСанПіН 3.3.2.007–98.

Щоб виключити попадання відбитих відблисків в очі користувачів поверхні в приміщенні повинні мати матову чи напівматову фактуру. Коефіцієнт відбиття має становити: для стелі 0,7— 0,8; стін 0,5—0,6; підлоги 0,3—0,5; інших поверхонь 0,4—0,5.

Площа, на якій розташовується одне робоче місце з ПК, повинна становити не менше ніж 6,0 м<sup>2</sup>, а об'єм приміщення — не менше ніж 20,0 м<sup>3</sup>. У приміщеннях з ПК слід щоденно робити вологе прибирання. Приміщення із ПК мають бути оснащені аптечками першої медичної допомоги.

Виробничі приміщення повинні обладнуватись шафами для зберігання документів, магнітних дисків, полицями, стелажми, тумбами тощо, з урахуванням вимог до площі приміщень.

Відповідне приміщення повинно бути укомплектоване системами центрального або індивідуального опалення, кондиціонування чи вентиляції повітря. Але при установці зазначених систем, необхідно переконатись, що батареї опалення, водопровідні труби, вентиляційні кабелі тощо, надійно сховані під захисними щитками, які перешкоджатимуть можливому потраплянню робітника під напругу.

### **3.2.2 Мікроклімат робочої зони працівників, вентиляція.**

Гігієнічні вимоги до параметрів виробничого середовища включають вимоги до параметрів мікроклімату, освітлення, шуму й вібрації, рівнів електромагнітного та іонізуючого випромінювання.

У виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря (ГОСТ 12.1.005-88, СН 4088-86).

					РП 05.22.003 ДП ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Табл1. Оптимальні величини температури, відносної вологості та швидкості руху повітря в робочій зоні виробничих приміщень

Період року	Категорія робіт (ГОСТ 12. 1.005-88)	Температура повітря, °С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодної	Легка — 1а	22—24	40—60	0,1
	Легка — 1б	21—23	40—60	0,1
Теплий	Легка — 1а	23—25	40—60	0,1
	Легка — 1б	22—24	40—60	0,2

До категорії 1а належать роботи, що виконуються сидячи і не потребують фізичного напруження, при яких витрати енергії складають до 139 Вт, а до категорії 1б — роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням, при яких витрати енергії становлять від 140 до 174 Вт.

Рівні іонізації повітря приміщень при роботі за ПК

Рівні	Кількість іонів в 1см <sup>3</sup> повітря	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Макс.допустимі	50000	50000

Для підтримки допустимих значень мікроклімату та концентрації позитивних та негативних іонів необхідно передбачати установки або прилади зволоження та/або штучної іонізації, кондиціювання повітря.

### **3.2.3. Освітлення робочого місця, шум, вібрація**

Природні освітлення повинно бути боковим, бажано одностороннім

Для уникнення засліплюючої дії сонячних променів найкраще, коли світловіотвори (вікна) зорієнтовані на північ чи північний схід. Коефіцієнт природної освітленості (КПО) повинен бути не нижче 1,5%.

Штучне освітлення у приміщеннях з ПК необхідно здійснювати у вигляді загальної системи рівномірного освітлення. У приміщеннях, де переважають роботи з документами, допускається застосування комбінованого освітлення.

Значення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500 лк. Якщо це неможливо забезпечити системою загального освітлення, допускається використовувати місцеве освітлення. При цьому світильники місцевого освітлення слід встановлювати таким чином, щоб не створювати бликів на поверхні екрана, а освітленість екрана має не перевищувати 300 лк.

Як джерела світла для штучного освітлення мають застосовуватись переважно люмінесцентні лампи типу ЛБ. У разі влаштування відбитого освітлення у виробничих та адміністративно-громадських приміщеннях допускається застосування металогалогенних ламп потужністю 250 Вт.

На комп'ютеризованих робочих місцях основними джерелами шуму є вентилятори системного блоку, накопичувачі, принтери ударної дії. Для зниження рівнів шуму на робочих місцях рекомендується розмістити друкувальні пристрої ударної дії (матричні, шрифтові принтери тощо) в іншому приміщенні, або огородити їх звукоізолюючими екранами.

### **3.2.4 Організація робочого місця користувача ПК**

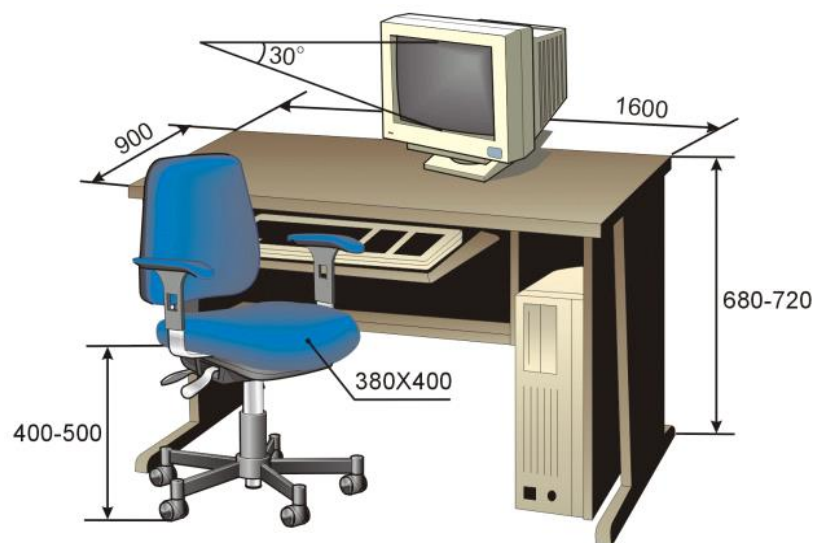
					РП 05.22.003 ДП ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Роботодавець, який використовує найману працю робітників, повинен забезпечити відповідність їхніх робочих місць комфортним та безпечним умовам. Розмір одного робочого місця має становити не менше 6 квадратних метрів. При необхідності, суміжні робочі місця співробітників, що працюють з комп'ютером, слід розділити перегородками висотою до 2 метрів.

При визначенні достатнього розміру приміщення і робочого місця на одну особу необхідно додатково враховувати шафи, сейфи, тумби або інші предмети меблів чи обладнання, які знаходяться в кімнаті. На столі працівника можливо розмістити допоміжні для роботи пристрої (принтери, колонки, сканери), а також місця для зберігання документів, за умови, що це не обмежуватиме видимість екрану і не заважатиме працівнику.

У разі надмірного шуму чи вібрації технічного обладнання, роботодавець повинен забезпечити працівників антивібраційними килимками.

Робочий стілець співробітника має бути підйомно-поворотним, легко регульованим за висотою та забезпечувати належну підтримку та зручне положення спини і хребта особи. Щодня необхідно проводити вологе прибирання приміщення, та очищати робоче місце та безпосередньо монітор комп'ютера від запиленості.



При організації праці, що пов'язана з використанням персональних комп'ютерів, для збереження здоров'я працюючих, запобігання професійним

					РП 05.22.003 ДП ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

захворювання і підтримки працездатності слід передбачити внутрішньозмінні регламентовані перерви для відпочинку. Внутрішньозмінні режими праці і відпочинку мають передбачати додаткові нетривалі перерви в періоди, що передують появі об'єктивних і суб'єктивних ознак стомлення і зниження працездатності

Протягом дня мають передбачатися:

- перерви для відпочинку і вживання їжі (обідні перерви);
- перерви для відпочинку і особистих потреб (згідно з трудовими нормами);
- додаткові перерви, що вводяться для окремих професій з урахуванням особливостей трудової діяльності

З метою зменшення негативного впливу монотонності є доцільним застосовувати чергування операцій усвідомленого тексту і числових даних (зміна змісту роботи), чередування вводу даних та редагування текстів. Для зниження нервово-емоційного напруження, стомлення зорового аналізатору, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільні деякі перерви використовувати для виконання комплексу вправ.

За умови високого рівня напруженості робіт з персональним комп'ютером показане психологічне розвантаження у спеціально обладнаних приміщеннях (в кімнатах психологічного розвантаження) під час регламентованих перерв або в кінці робочого дня.

### **3.3 Пожежна безпека**

Приміщення, де розміщені робочі місця, мають бути оснащені системою автоматичної пожежної сигналізації і вогнегасниками відповідно до вимог чинного законодавства України. Проходи до засобів пожежогасіння мають бути вільними.

					РП 05.22.003 ДП ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Для гасіння пожеж на робочому місці використовують вуглекислотні та порошкові вогнегасники.

Вуглекислотні вогнегасники випускаються як ручні (ВВК-5).

Порошкові вогнегасники ВП-2, ВП-5, ВП-10 та інші.

Наявність первинних засобів пожежогасіння і вогнегасників, їхня кількість і зміст відповідає вимогам ГОСТ 12.4.009-75 і ISO3941-77. У приміщенні виконуються усі вимоги по пожежній безпеці відповідно до вимог НАПБ А.0.001-95 “Правила пожежної безпеки в Україні”.

У приміщенні також мається план евакуації на випадок виникнення пожежі. Час евакуації відповідає вимозі СНиП 2.01.02-85, а максимальне видалення робочих місць від евакуаційних виходів відповідає СНиП 2.09.02-85.

					РП 05.22.003 ДП ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У ході дослідження, проведеного в рамках даного дипломного проекту, були виявлені проблеми, що виникають під час використання мікросервісної архітектури для побудови високонавантаженого прикладного програмного забезпечення. Ці проблеми лежать як у частині процесу розробки та управління цим процесом, так і в частині інфраструктури та системної архітектури. Наявність цих проблем з часом розробки системи тягне у себе збільшення часових інтервалів з моменту надходження вимоги до, до введення розробленої функціональності в промислову експлуатацію, і навіть є причиною неоптимального використання обчислювальних ресурсів. Це призводить до значного збільшення часу запиту на серверну частину і, як наслідок, відчутного зростання вартості кінцевого рішення. Аналіз, проведений у рамках даної роботи, показав, що для зниження впливу проблем, описаних вище, на вартість кінцевого рішення та на час постачання нової функціональності необхідно автоматизувати сам процес розробки, процес виділення обчислювальних потужностей та процес постачання. У ході дослідження був проведений аналіз підходів та способів автоматизації зазначених вище процесів. Аналіз показав, що для досягнення результату необхідно впровадити безперервні практики, такі як безперервна інтеграція та безперервне розгортання, а також перейти від традиційної віртуалізації до контейнеризації мікросервісів за допомогою механізму оркестрування контейнерів.

					РП 05.22.000 ДП ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Martin L. Abbott. The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise. –Addison-Wesley Professional, 2015.–624 с.
2. Vikram Murugesan, Microservices Deployment Cookbook. –Packt Publishing, 2017. –378 с.
3. Sam Newman. Building Microservices: Designing Fine-Grained Systems. –O'Reilly Media, 2015. –280 с.
4. Sourabh Sharma, Rajesh RV, David Gonzalez. Microservices: Building Scalable Software. –Packt Publishing Limited, 2017. –919 с.
5. Rajesh RV, Spring Microservices. –Packt Publishing, 2016. –436 с.
6. Pethuru Raj. Docker: Creating Structured Containers. –Packt Publishing, 2016. –320 с.
7. Web Call Server 5: [Электронный ресурс]. URL: <https://flashphoner.com/>
8. 40 open source, free and top UML tools: [Электронный ресурс]. URL: <https://www.predictiveanalyticstoday.com/open-source-free-unified-modeling-language-uml-tools/>
9. Что такое ASP.NET: [Электронный ресурс]. URL: <http://www.internet-technologies.ru/articles/lekciya-1-chto-takoe-asp-net-installyaciya-i-testovyy-proekt.html>
10. Web Call Server - Руководство Разработчика: [Электронный ресурс]. URL: [https://flashphoner.com/docs/wcs5/wcs\\_docs/html/ru/wcs-developer-guide-2/](https://flashphoner.com/docs/wcs5/wcs_docs/html/ru/wcs-developer-guide-2/)
11. Which is Best for Web Application Development—Dot Net, PHP, Python, Ruby, or Java: [Электронный ресурс]. URL: <https://www.addonsolutions.com/blog/which-is-best-for-web-application-development-dot-net-php-python-ruby-or-java.html/> (Дата обращения 15.04.2018);

					РП 05.22.000 ДП ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44