МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ПРИРОДНИЧО-ГУМАНІТАРНИЙ ФАХОВИЙ КОЛЕДЖ

ДВНЗ «Ужгородський національний університет»

Реєстраційний номер \_\_\_\_\_\_\_\_\_\_\_

Дата реєстрації \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тимко Олександр Андрійович

студента 4 курсу

групи КН-41

денної форми навчання

залікова книжка №253006

**ЗВIT**

ПРОХОДЖЕННЯ ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ

Галузь знань 12 «Інформаційні технології»,

спеціальність 121 «Інженерія програмного забезпечення»

**«Сервіс конвертування файлів»**

**Допущено до захисту** Науковий керівник:

Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ Нажмієва Н.М. \_\_\_\_\_\_\_\_\_\_\_\_

Підпис \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Консультант з охорони праці:

Щербан І. І. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Консультант з економіки:

Поп І. М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Практика захищена «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 р. з оцінкою «\_\_\_\_\_\_\_\_\_\_\_\_»**

**Підписи членів комісії\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Ужгород 2023

Автор проєкту: Тимко Олександр Андрійович

(прізвище, ім’я, по батькові)

Тема дипломного проєкту: Сервіс конвертування файлів

Науковий керівник: викл. Владимир Ульяна Миколаївна

(науковий ступінь, посада, прізвище, ім’я, по батькові)

Рецензент:

(науковий ступінь, посада, прізвище, ім’я, по батькові)

Дата захисту: «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2023 року

Анотація

Переддипломний проєкт складається з чотирьох розділів, вступу, висновків, переліку використаних джерел та додатків.

У першому розділі розглянуто загальні відомості про технології, що були використані під час написання проєкту.

У другому розділі описано роботу проєкту, таблиці бази даних.

У розділі «Економічна частина» прораховано витрати на створення проєкту, повну собівартість і гуртову ціну програмного продукту.

У розділі «Охорона праці» описано аналіз можливих небезпек на робочому місці, в робочій зоні та у виробничому середовищі відповідно виду робіт, що розглядаються в проекті. Також зроблений аналіз кількості вогнегасників виходячи з площі приміщення.

Обсяг пояснювальної записки 49

Перелік додатків: лістинг, документація програмного коду, схема зв’язків між таблицями.

Project author: Tymko Oleksandr Andriyivych

Project subject: File conversion service

Supervisor: lect. Vladimir Ulyana Nikolaevna

Reviewer: Date: «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2023

Annotation

The diploma project consists of four sections, introduction, conclusions, list of used sources and appendices.

The first section discusses general information about the technologies used when writing the project.

The second section describes the work of the project, database tables.

The section «Economic part» calculates the cost of creating a project, the total cost and wholesale price of the software product.

The section «Labor protection» section describes the analysis of possible hazards at the workplace, in the work area and in the production environment, according to the type of work considered in the project. It also analyzes the number of fire extinguishers based on the floor space.

Length of the explanatory note 49

List of applications: listing, software code documentation, diagram of relationships between tables.

### ЗМІСТ

ВСТУП 7

[1 ОГЛЯД ТЕОРЕТИЧНОЇ ТА ПРАКТИЧНОЇ ЧАСТИНИ ДИПЛОМНОГО ПРОЄКТУ 8](#_Toc135386654)

[1.1 Сервіс конвертування файлів 8](#_Toc135386655)

[1.2 Огляд аналогів 10](#_Toc135386656)

[1.3 Загальні відомості про використані технології 11](#_Toc135386657)

[1.3.1 Мова розмітки HTML 11](#_Toc135386658)

[1.3.2 Таблиці стилів CSS 12](#_Toc135386659)

[1.3.3 Мова програмування JavaScript 12](#_Toc135386660)

[1.3.4 Фреймворк Django 14](#_Toc135386661)

[1.3.5 TypeScript 15](#_Toc135386662)

[1.4 Опис алгоритму 15](#_Toc135386663)

[1.5 Інтерфейс та керівництво користувача 17](#_Toc135386664)

[1.6 Огляд роботи проєкту 17](#_Toc135386665)

[2 ЕКОНОМІЧНА ЧАСТИНА 20](#_Toc135386666)

[2.1 Характеристика та призначення роботи 20](#_Toc135386667)

[2.2 Витрати на створення програмного продукту 20](#_Toc135386668)

[2.2.1 Розрахунок вартості витрат на матеріали 20](#_Toc135386669)

[2.2.2 Розрахунок витрат на оплату праці працівника 21](#_Toc135386670)

[2.3 Витрати на утримання та експлуатацію ПЕОМ 22](#_Toc135386671)

[2.4 Загальновиробничі витрати 24](#_Toc135386672)

[2.5 Обчислення собівартості програмного продукту 24](#_Toc135386673)

[2.6 Обчислення гуртової ціни програмного продукту 25](#_Toc135386674)

[2.7 Висновки економічної доцільності проєкту 25](#_Toc135386675)

[3 ОХОРОНА ПРАЦІ 26](#_Toc135386676)

[3.1 Загальні вимоги охорони праці при роботі з ЕОМ 26](#_Toc135386677)

[3.2 Заходи щодо поліпшення умов з охорони праці 30](#_Toc135386678)

[3.3 Інструкція з охорони праці при роботі з ПЕОМ та відео-дисплейними терміналами 31](#_Toc135386679)

[3.3.1 Загальні положення 31](#_Toc135386680)

[3.3.2 Вимоги безпеки перед початком роботи 35](#_Toc135386681)

[3.3.3 Вимоги безпеки під час роботи 37](#_Toc135386682)

[3.3.4 Вимоги безпеки після закінчення роботи 38](#_Toc135386683)

[3.3.5 Вимоги безпеки в аварійних ситуаціяї 38](#_Toc135386684)

[ВИСНОВКИ 39](#_Toc135386685)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 41](#_Toc135386686)

[Додаток. Лістинг програми 43](#_Toc135386687)

Зараз у сучасному цифровому світі, коли інформація стала найціннішим ресурсом, обмін і перетворення файлів мають велике значення для ефективного взаємодії між людьми та комп’ютерами. У цьому контексті сервіси конвертування файлів набувають все більшої популярності та стають незамінними інструментами для широкого кола користувачів.

Тема проєкту - сервіс конвертування файлів, є актуальною та цікавою, оскільки вона стикається з проблемою, що виникає у багатьох користувачів щодо несумісності форматів файлів. Перетворення файлів з одного формату в інший може бути непростим завданням, особливо якщо врахувати різноманітність форматів і специфічні особливості кожного з них.

Метою проєкту є розробка та імплементація сервісу конвертування файлів, який забезпечить зручний та швидкий обмін і перетворенням даних у різних форматах. Цей сервіс буде призначений для широкого спектру користувачів, включаючи студентів, професіоналів, бізнес-користувачів та інших осіб, які потребують зручних і надійних засобів конвертації файлів.

# 1 ОГЛЯД ТЕОРЕТИЧНОЇ ТА ПРАКТИЧНОЇ ЧАСТИН ДИПЛОМНОГО ПРОЄКТУ

## **1.1 Сервіс конвертування файлів**

Сервіс конвертування файлів включає в себе огляд основних понять, принципів роботи та функцій, пов’язаних з конвертацією різних типів файлів. Нижче подано загальний огляд цієї теми:

* Конвертація файлів: Конвертація файлів означає перетворення файлів з одного формату в інший. Це може бути необхідно для забезпечення сумісності з різними програмами, пристроями або операційними системами. Конвертація може включати зміну формату зображень, відео, аудіо, документів, електронних книг та інших типів файлів.
* Формати файлів: Формат файлу визначає структуру та спосіб організації даних в файлі. Кожен тип файлу має свій унікальний формат, який визначається розширенням файлу (наприклад, .docx для документів Microsoft Word, .jpg для зображень у форматі JPEG тощо). Різні програми можуть підтримувати різні формати файлів, тому конвертація може бути потрібна для зручності використання файлів в різних середовищах.
* Онлайн сервіси конвертування файлів: Існують різні онлайн сервіси, які надають можливість конвертувати файли через Інтернет без необхідності встановлення спеціального програмного забезпечення на комп’ютері. Користувачі можуть завантажити вихідний файл на сервер сервісу та отримати його конвертовану версію у вказаному форматі.
* Типи конвертації: Сервіси конвертування файлів можуть підтримувати різні типи конвертації залежно від категорії файлів, яку вони обробляють. Наприклад, існують сервіси, які спеціалізуються на конвертації зображень, відео, аудіо, документів, електронних книг тощо. Деякі сервіси можуть підтримувати конвертацію між різними категоріями файлів.
* Функціональні можливості: Онлайн сервіси конвертування файлів можуть надавати різні функції для забезпечення зручного та якісного конвертування. Це можуть бути опції зміни якості, розміру або роздільної здатності зображень та відео, обрізка аудіофайлів, збереження структури документів, розпізнавання тексту тощо. Деякі сервіси також можуть надавати можливість пакетної конвертації багатьох файлів одночасно.
* Конфіденційність та безпека: Оскільки конвертація файлів відбувається на сервері сервісу, користувачі повинні бути уважними щодо приватності та безпеки своїх файлів. Рекомендується користуватися надійними сервісами, які гарантують захист персональних даних та безпечне завантаження та збереження файлів.

Загальний огляд теоретичної частини сервісу конвертування файлів надає загальне розуміння цієї теми. При виборі сервісу конвертування файлів важливо дослідити функціонал, підтримувані формати та безпеку, щоб забезпечити задоволення своїх потреб у конвертації файлів[1].

## **1.2 Огляд аналогів**

За аналоги приведено декілька онлайн сервісів для конвертування файлів.

Convertio — це онлайн-сервіс для конвертації файлів з одного формату в інший[2]. (рисунок 1.1)

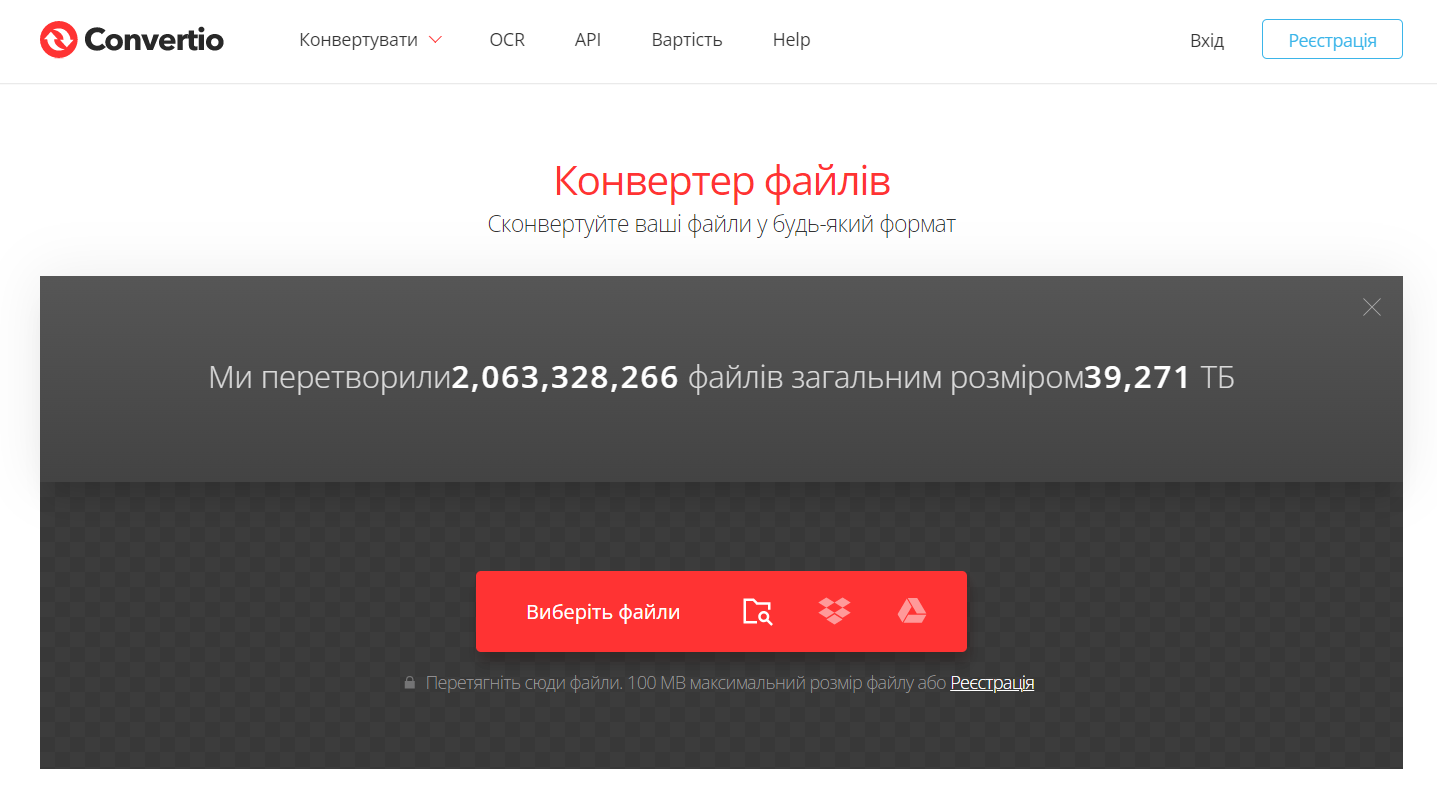


Рисунок 1.1 Онлайн сервіс https://convertio.co/

[www.online-convert.com](https://www.online-convert.com/) - це онлайн-інструмент для конвертації файлів з одного формату в інший[3].

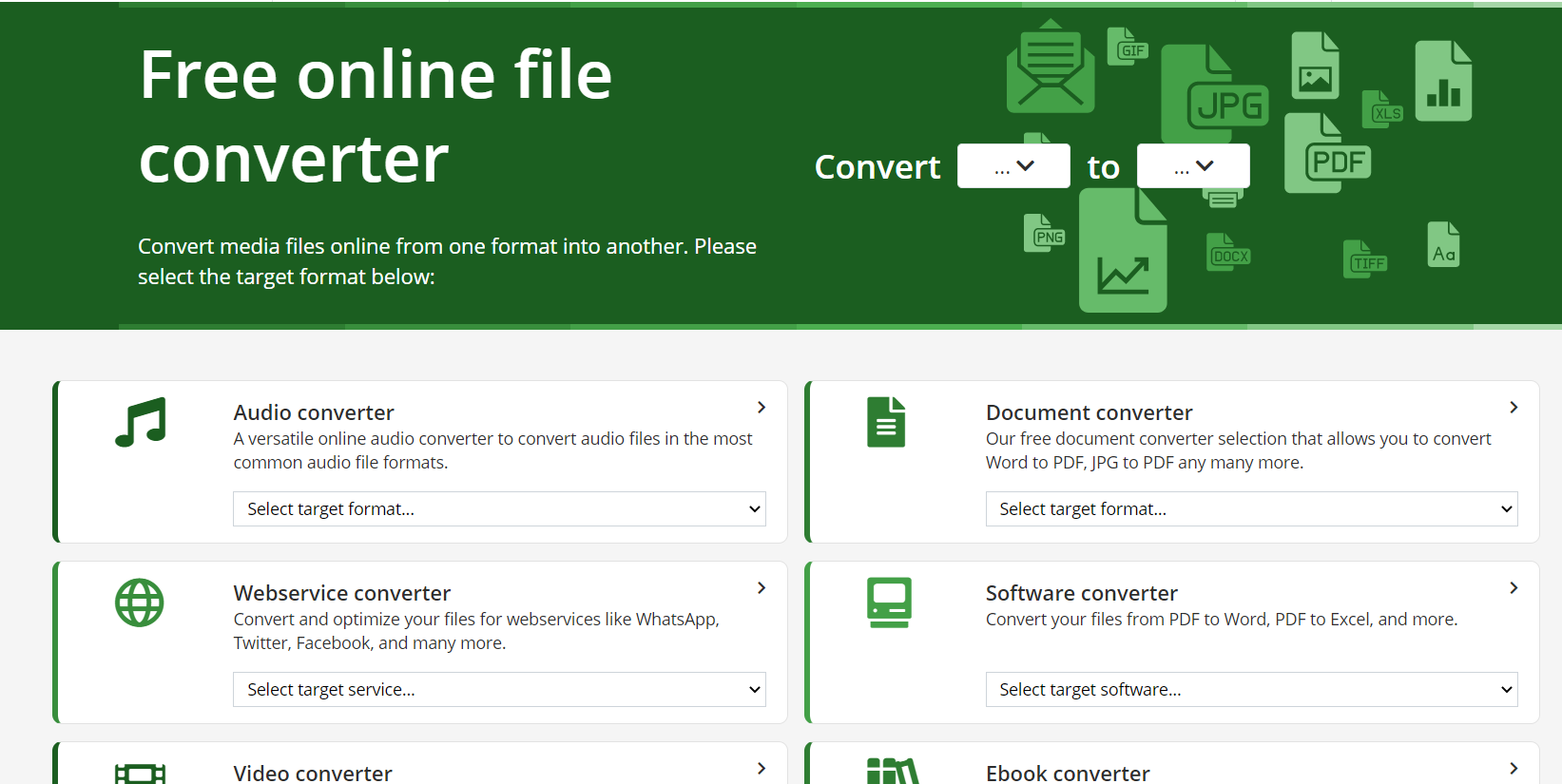


Рисунок 1.2 Онлайн сервіс https://www.online-convert.com/

## **1.3 Загальні відомості про використані технології**

Вебсайт магазину вторинного вжитку для компанії з розробки програмних продуктів розроблено на базі наступних технологій та засобів:

* мови розмітки HTML 5;
* таблиць стилів CSS 3
* скриптованої мови програмування JavaScript;
* TypeScript, засіб для розширення можливостей JavaScript;
* фреймворк для серверної частини сайту Django;
* засіб для реалізації авторизації JWT.

### 1.3.1 Мова розмітки HTML

HTML, або мова розмітки гіпертексту, — це стандартна мова розмітки, яка використовується для створення вебсторінок і вебдодатків. HTML використовується для структурування та відображення вмісту в Інтернеті, включаючи текст, зображення, відео, посилання та інші елементи.

Основними компонентами HTML є теги (також відомі як елементи), атрибути та вміст. Теги використовуються для визначення різних елементів на вебсторінці, таких як заголовки, абзаци, списки, зображення тощо. Атрибути використовуються для визначення додаткової інформації про елементи, такої як класи, ідентифікатори, розміри тощо. Вміст, наприклад, текст або медіа-елементи, розміщується між тегами.

HTML використовує вкладену розмітку, де елементи можуть бути вкладені один в одного для створення більш складних структур. Він також використовує блокову та вбудовану розмітку, яка визначає розташування та вигляд елементів на вебсторінці.

Однією з ключових особливостей HTML є те, що це мова розмітки, а не мова програмування. Вона описує структуру і зовнішній вигляд вебсторінок, але не включає логіку або функціональність, наприклад, взаємодію з користувачем або обробку даних. Для цього використовуються різні мови програмування, такі як JavaScript, які можуть бути вбудовані в HTML-документи для забезпечення додаткової функціональності[4].

### 1.3.2 Таблиці стилів CSS

CSS (Cascading Style Sheets — каскадні таблиці стилів) — це мова розмітки стилів, яка використовується веброзробниками для опису зовнішнього вигляду вебсторінок. CSS дозволяє розділити структуру (HTML) і зовнішній вигляд (стилі) вебдокумента, що дає можливість керувати відображенням вебсторінок на різних пристроях і браузерах.

Основна мета CSS — забезпечити стильову розмітку, яка визначає зовнішній вигляд елементів вебсторінки, таких як кольори, розміри, шрифти, розташування, відступи та інші властивості. CSS використовується разом з HTML для визначення зовнішнього вигляду вебдокумента, включаючи фон, межі, тіні, анімацію тощо.

CSS використовується для створення різноманітних стилів, від простих до складних, що дозволяє розробникам надавати вебсторінкам сучасний вигляд. Крім того, CSS дозволяє розробникам створювати адаптивні дизайни, які забезпечують коректне відображення вебсторінок на різних екранах і пристроях, таких як комп’ютери, планшети, смартфони та інші.

CSS також дозволяє використовувати різні концепції, такі як каскадування, успадкування, класи, ідентифікатори, псевдоелементи і псевдокласи, що робить його потужним інструментом для управління зовнішнім виглядом вебсторінок.

Однією з головних переваг CSS є можливість легко змінювати зовнішній вигляд вебсторінок без зміни їхньої структури. Це дозволяє розробникам ефективно контролювати відображення[5].

### 1.3.3 Мова програмування JavaScript

JavaScript є високорівневою, інтерпретованою мовою програмування, яка використовується для розробки динамічних вебдодатків та взаємодії з користувачем на стороні клієнта. Вперше випущений в 1995 році, JavaScript є однією з найпопулярніших мов програмування, використовуваних в розробці вебдодатків.

JavaScript дозволяє веброзробникам створювати різноманітні функції та взаємодії на стороні клієнта, такі як динамічні вебсторінки, валідацію форм, анімацію, обробку подій, взаємодію з сервером, взаємодію з додатками соціальних мереж, графічні ефекти, асинхронні запити до сервера та багато іншого. JavaScript також використовується в розробці серверних додатків, використовуючи платформи, такі як Node.js.

Однією з головних рис JavaScript є те, що він є мовою програмування, заснованою на об’єктно-орієнтованому підході, де всі елементи (змінні, функції, об’єкти) є об’єктами, з властивостями та методами. JavaScript використовує динамічну типізацію, що дозволяє змінювати типи даних під час виконання програми. JavaScript має велику кількість бібліотек та фреймворків, таких як jQuery, React, Angular, Vue та інші, які роблять розробку вебдодатків більш ефективною та продуктивною. JavaScript також є важливою частиною сучасного вебекосистеми, включаючи розробку мобільних додатків, розширень браузерів, розробку ігор, розробку серверних додатків та багато іншого.

JavaScript має синтаксис, схожий на C, що робить його відносно легким для вивчення та використання. Він може бути вбудований безпосередньо в HTML-сторінки або використовуватись зовнішніми файлами скриптів. JavaScript використовується для додавання взаємодії на вебсторінках, таких як валідація введених даних, взаємодія з елементами DOM (Document Object Model), асинхронна комунікація з сервером, маніпулювання даними на стороні клієнта та багато іншого.

JavaScript є мовою з відкритим кодом, що означає, що розробники можуть створювати свої власні функції та бібліотеки, вдосконалювати мову та співпрацювати в співтоваристві розробників. Він також має велику кількість ресурсів та документації, які допомагають розробникам розуміти та використовувати мову ефективно.

JavaScript використовується в широкому спектрі веброзробки, від простих вебсторінок до складних вебдодатків та додатків мобільних пристроїв. Він є однією з основних мов програмування, яка використовується в розробці фронтенду, тобто в розробці користувацького інтерфейсу вебдодатків. Вміння програмувати на JavaScript є важливим для багатьох веброзробників, що дозволяє їм створювати багатофункціональні та взаємодіючі вебдодатки для користувачів[6].

### 1.3.4 Фреймворк Django

Django — це високорівневий вебфреймворк на мові Python, який дозволяє розробникам швидко та ефективно створювати вебдодатки. Це безкоштовний фреймворк з відкритим вихідним кодом, який працює за архітектурною схемою Model-View-Controller (MVC).

Деякі з ключових особливостей Django включають:

* об’єктно-реляційне відображення (ORM): Django надає високорівневе ORM, яке дозволяє розробникам взаємодіяти з базою даних за допомогою класів Python замість SQL.
* інтерфейс адміністратора: Django постачається з вбудованим інтерфейсом адміністратора, який дозволяє розробникам легко керувати даними свого додатку.
* маршрутизація URL-адрес: Django дозволяє розробникам зіставляти URL-адреси з представленнями за допомогою простої та гнучкої системи маршрутизації URL-адрес.
* движок шаблонів: Django надає механізм шаблонів, який дозволяє розробникам створювати динамічні HTML-сторінки.
* засоби безпеки: Django надає декілька вбудованих функцій безпеки, включаючи захист від поширених вразливостей вебдодатків, таких як Cross-Site Scripting (XSS) та Cross-Site Request Forgery (CSRF).
* загалом, Django є популярним вибором для веброзробки, оскільки він надає потужний набір інструментів і добре документований, що дозволяє розробникам швидко розпочати роботу[7].

### 1.3.5 TypeScript

TypeScript — це відкрита надбудова мови JavaScript, яка надає можливості статичної типізації, розширення синтаксису та компіляції в JavaScript. Вона була розроблена компанією Microsoft і набула значної популярності серед розробників вебдодатків.

Однією з ключових особливостей TypeScript є можливість використання статичної типізації, яка дозволяє визначати типи змінних, параметрів функцій та значень, що повертаються. Це допомагає зменшити кількість помилок на етапі розробки та покращує розуміння коду. Крім того, TypeScript надає розширення синтаксису JavaScript, такі як введення інтерфейсів, класів, зчислень та інших функцій, які допомагають створювати більш чистий, структурований і розширюваний код.

Ще однією важливою особливістю TypeScript є компіляція в JavaScript. Код на TypeScript можна скомпілювати в коректний JavaScript, що дозволяє запускати його в будь-якому веббраузері або середовищі виконання JavaScript. Це також означає, що розробники можуть використовувати TypeScript у будь-якому проєкті, який використовує JavaScript, без необхідності переписувати весь код[8].

## **1.4 Опис алгоритму**

Основний алгоритм роботи сервісу конвертування файлів може включати наступні кроки:

* Користувач вибирає файл, який бажає конвертувати. Це може бути файл з його комп’ютера.
* Сервіс перевіряє формат вихідного файлу та формат, у який користувач хоче його конвертувати. Наприклад, це можуть бути формати документів (наприклад, PDF, DOCX, TXT), зображень (наприклад, JPEG, PNG, GIF), відео (наприклад, MP4), аудіо (наприклад, MP3) тощо.
* Сервіс перевіряє наявність необхідних ресурсів і програм, необхідних для конвертування файлу в запитаний формат. Це може включати перевірку наявності відповідних конвертерів, кодеків або бібліотек.
* Якщо необхідні ресурси доступні, сервіс обробляє вихідний файл за допомогою відповідного конвертера або обробника формату. Цей крок може включати відкриття вихідного файлу, зчитування його вмісту, виконання необхідних операцій (наприклад, зміна кодеку, розмірів, компресія, редакція, тощо) та створення нового файлу у форматі, який бажає отримати користувач.
* Коли конвертування завершено, сервіс повідомляє користувача про готовий файл у бажаному форматі. Це може бути відображення посилання на завантаження файлу, відправлення електронного листа з файлом або збереження файлу в особистому обліковому записі користувача.
* У випадку, якщо необхідні ресурси недоступні або конвертування неможливе з певних причин (наприклад, несумісність форматів, пошкоджений файл тощо), сервіс повідомляє користувача про помилку.

Це загальний опис алгоритму роботи сервісу конвертування файлів, і реальна реалізація може відрізнятися залежно від конкретного сервісу або програмного забезпечення.

## **1.5 Інтерфейс та керівництво користувача**

Для запуску сервісу конвертування файлів потрібно написати декілька команд.(рисунок 1.3)

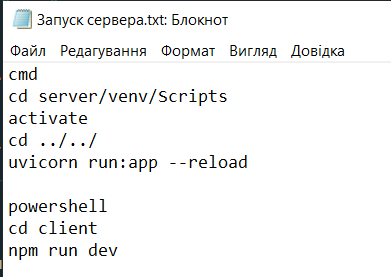


Рисунок 1.3 Конанди для запуску серверу

Після запуску серверу відкривається вікно, де можна буде конвертувати файли. Інтервейс який бачить користувач(рисунок 1.4)

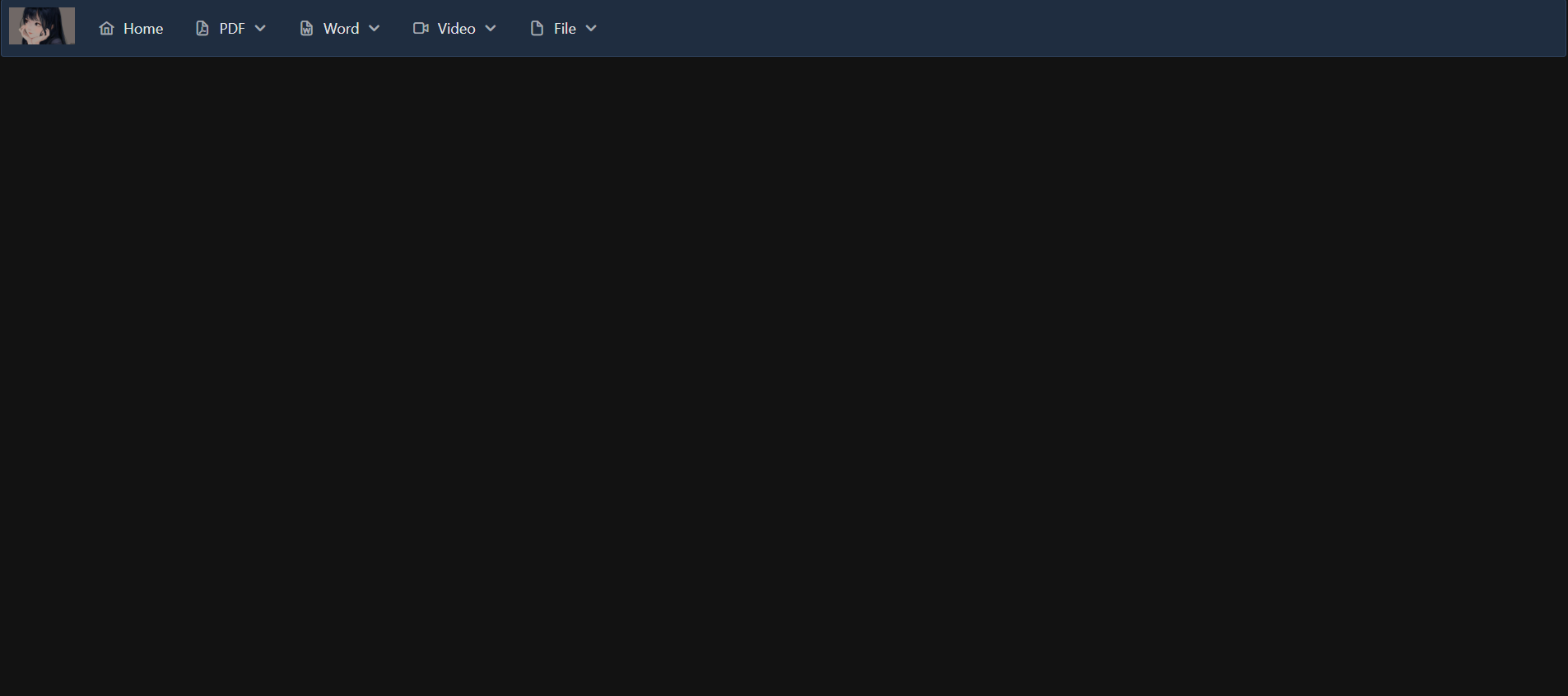


Рисунок 1.4 Головна сторінка

## **1.6 Огляд роботи проєкту**

Детальний опис роботи програми: (рисунок 1.5-1.8)

1. Вибір типу файлу:

* Після входу в систему користувач має можливість вибрати тип файлу, який він бажає конвертувати.
* Сервіс підтримує різні типи файлів, такі як документи (наприклад, DOCX, PDF), зображення (наприклад, JPG, PNG), відео (наприклад, MP4).

1. Вибір формату конвертації:

* Користувач обирає формат, у який він бажає конвертувати свій файл.
* Наприклад, якщо вихідний файл є документом у форматі DOCX, користувач може обрати, наприклад, формат PDF або TXT для конвертації.

1. Завантаження та конвертація файлу:

* Користувач має можливість завантажити вихідний файл, який він бажає конвертувати.
* Після завантаження файлу сервіс починає процес конвертації вибраного типу файлу у вибраний формат.

1. Завантаження конвертованого файлу:

* Після успішної конвертації користувач може завантажити новий файл у обраному форматі.
* Після завершення конвертації, користувач отримує архів з конвертованим файлом

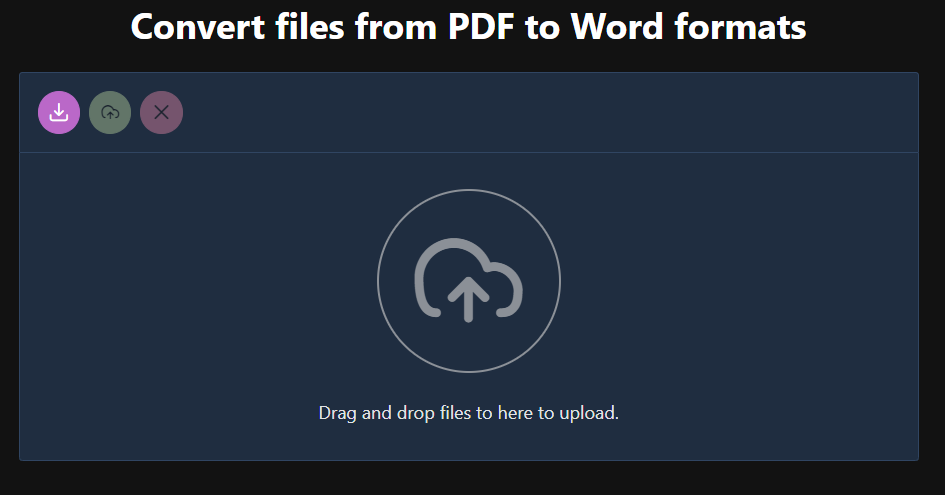


Рисунок 1.5 Вікно для конвертації

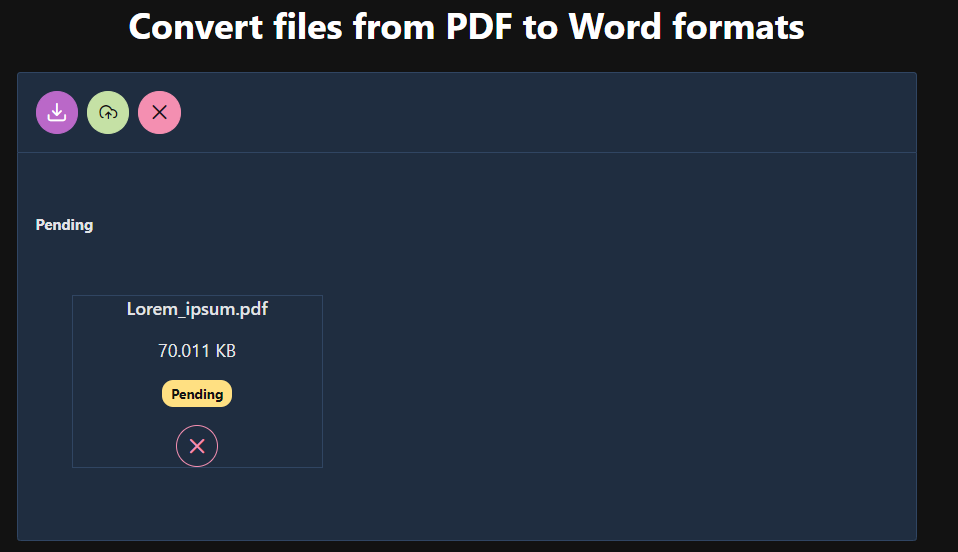


Рисунок 1.6 Файл завантажений для конвертації

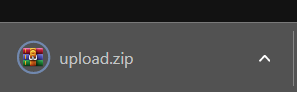


Рисунок 1.7 Заархівований файл

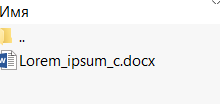


Рисунок 1.8 Конвертований файл

# 2 ЕКОНОМІЧНА ЧАСТИНА

## **2.1 Характеристика та призначення роботи**

Завданням переддипломного проєкту є розробка «Сервіс конвертування файлів». Цей сервіс надає можливість конвертування більшості розширень.

## **2.2 Витрати на створення програмного продукту**

### 2.2.1 Розрахунок вартості витрат на матеріали

Розрахунок виконуємо виходячи з витрат на допоміжні матеріали, перелік яких приведений в таблиці 2.1.

*Таблиця 2.1 Допоміжні матеріали*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  п/п | Найменування  матеріалу | Виробник і  марка | Одиниця  виміру | Норма  затрат | Ціна за одиницю,  грн. | Вартість,  грн. |
| 1 | Картридж | Canon PG 460 (3711C001) | Шт. | 1 | 750 | 750 |
| 2 | Лазерний диск CD–R | HP CD-R | Шт. | 2 | 8 | 16 |
| 3 | Офісний папір формату А4 | Buromax Euromax А4 80 г/м2 100 листів | Упак. | 1 | 74 | 74 |
| 4 | Конверт | Master | Шт. | 1 | 5 | 5 |
| 5 | Папка швидкозшивач А4 | Economix | Шт. | 1 | 10 | 10 |
|  | Всього | | | | | 855 |

Загальна вартість допоміжних матеріалів складає Звм = 855 грн.

### 2.2.2 Розрахунок витрат на оплату праці працівника

Для нарахування заробітної плати працівника приймаємо погодинну форму оплати.

|  |  |
| --- | --- |
| ЗПпог= t × ТЗПГОД , |  |

де t — час на створення програмного продукту, год.;

ТЗПГОД — тарифна ставка працівника певного розряду за годину праці, грн./год.;

Дані про трудомісткість операцій, виконуваних в процесі створення програмного продукту, приведені в таблиці 2.2.

*Таблиця 2.2 Трудомісткість операцій*

|  |  |  |
| --- | --- | --- |
| №  п/п | Найменування операції | Витрати, год |
| 1 | Постановка задачі | 10 |
| 2 | Проєктування програмного інтерфейсу | 30 |
| 3 | Реалізація програмних модулів | 100 |
| 4 | Відлагодження | 40 |
| 5 | Тестування | 30 |
| 6 | Оформлення документації, інструкції користувача | 20 |
|  | Всього | 230 |

Тарифну ставку працівника (Тзпгод) за годину праці приймаємо виходячи з Єдиної тарифної сітки 2022 року, яка для програміста 16 розряду становить:

Тзпгод = 8 071 / (8×22) = 45,86 грн/год.

Виходячи з цього, витрати на оплату праці програміста згідно з формулою дорівнюватимуть:

 = 45,86 × 230 = 10547.8 грн.

Нарахування на заробітну плату складають 22%

Знарах= × 0,22 = 10547.8 × 0,22 = 2320.51 грн

Тоді загальні витрати на оплату праці складають

ЗПзаг=+ Знарах = 10547.8 + 2320.51 = 13271,59 грн

## **2.3 Витрати на утримання та експлуатацію ПЕОМ**

**У період дії воєнного стану працюють інші правила.** Зокрема, не застосовуються норми статей 71–73 КЗпП(святкові і неробочі дні). На практиці це означає, що офіційних святкових та неробочих днів в Україні немає. При цьому графік роботи та час відпочинку встановлює роботодавець. Тому, якщо є необхідність, вивести людей у святкові дні на роботу — кожен керівник підприємства чи фізособа-роботодавець вирішують у 2023 році самостійно.

Норма тривалості робочого часу у годинах при 40-годинному робочому тижні триває — 2080 годин.

Таким чином, річний фонд часу роботи ПЕОМ складає:

ТЕОМ = 8 × (365 - 105) = 2 080 год.

|  |  |
| --- | --- |
| Балансову вартість ПЕОМ обчислимо за формулою:  СБАЛ = СРИН + ЗвСТ , |  |

де СРИН — ринкова вартість ПЕОМ, грн.;

ЗвСТ — витрати на доставку й встановлення ПЕОМ, грн.

Витрати на доставку й встановлення ПЕОМ складають 10 % від ринкової вартості ПЕОМ. Покладаємо ринкову вартість ПЕОМ рівною СРИН = 25 000 грн і отримуємо балансову вартість ПЕОМ:

СБАЛ = СРИН + ЗвСТ = 25 000 + (25 000×0,1) = 27 500 грн.

Для нарахування амортизації використовуємо прямолінійний метод:

|  |  |
| --- | --- |
| ЗАМ = СБАЛ / Текспл. , |  |

де СБАЛ — балансова вартість ПЕОМ, грн.;

Текспл.—термін експлуатації (років).

Текспл. приймаємо 5 років

Таким чином річні амортизаційні відрахування складають:

ЗАМ = 27 500 / 5= 5 500 грн.

Вартість електроенергії, споживаної за рік, визначаємо за формулою:

|  |  |
| --- | --- |
| ЗЕЛ = PПЕОМ × ТПЕОМ × СЕЛ × А , |  |

де PПЕОМ — сумарна потужність ПЕОМ, кВт;

ТПЕОМ — річний час роботи ПЕОМ, год.;

СЕЛ — вартість 1 кВт×год. електроенергії, 1,68грн;

А — коефіцієнт інтенсивного використання потужності ПЕОМ.

СЕЛ=1,68грн;

Сумарна потужність ПЕОМ PПЕОМ = 0,2 кВт. Приймаємо коефіцієнт інтенсивного використання потужності ПЕОМ рівним А = 0,9 і обчислюємо вартість електроенергії, споживаної за 1986год:

ЗЕЛ = 0,2 × 2 080 × 1,68× 0,9 = 629 грн.

Інші витрати на експлуатацію ПЕОМ (в тому числі й витрати на поточний та профілактичний ремонт) складають 10 % від балансової вартості ПЕОМ і дорівнюють:

ЗПР = 0,1× СБАЛ = 0,1×27 500 = 2 750 грн.

Повні витрати на експлуатацію ПЕОМ визначаємо за формулою:

|  |  |
| --- | --- |
| ЗПЕОМ = ЗЗП + ЗАМ + ЗЕЛ + ЗПР , |  |

де ЗЗП — витрати на заробітну плату, грн;

ЗАМ — амортизаційні відрахування, грн;

ЗЕЛ — витрати на електроенергію, грн;

ЗПР — інші витрати на експлуатацію ПЕОМ, грн.;

тоді:

ЗПЕОМ = 13271,59 + 5 500 + 629 + 2 750 = 22150,59 грн.

Вартість машинної години визначаємо за формулою:

|  |  |
| --- | --- |
| СГОД = ЗПЕОМ / ТПЕОМ , |  |

де ЗПЕОМ — повні витрати на експлуатацію ПЕОМ, грн.;

ТПЕОМ — час роботи ПЕОМ, год.

СГОД = 22150,59 / 270 = 82,03 грн./год.

Витрати на оплату машинного часу при відлагодженні програми визначаємо за формулою:

|  |  |
| --- | --- |
| = СГОД × ТВ , |  |

де СГОД — вартість машинної години, грн./год.;

ТВ — фактичний час відлагодження програмного продукту на ПЕОМ, год.

З таблиці 3.2 видно, що ТВ = 50 год., тоді:

 = 82,03 × 50 = 4 101,85 грн.

## **2.4 Загальновиробничі витрати**

Витрати на освітлення, опалення, комунальні послуги та інші в нашому випадку приймаємо 30% від витрат на оплату праці

|  |  |
| --- | --- |
| ЗЗАГ = × 30%/100, |  |

де  – витрати на оплату праці програміста, грн.

ЗЗАГ = 13271,59 × 0,3 = 3981.3 грн.

## **2.5 Обчислення собівартості програмного продукту**

Для обчислення складаємо калькуляцію собівартості продукту за статтями витрат в таблиці 2.3

*Таблиця 2.3 Собівартість програмного продукту за статтями витрат*

|  |  |  |
| --- | --- | --- |
| № п/п | **Статті витрат** | Сума, грн. |
| 1 | Сировина та матеріали | 855 |
| 2 | Основна заробітна плата | 10547.8 |
| 3 | Нарахування на зарплату | 2320.51 |
| 4 | Витрати на оплату машинного часу | 4101,85 |
| 5 | Загальновиробничі витрати | 3981.3 |
| 6 | Повна собівартість | 21 804,14 |

## **2.6 Обчислення гуртової ціни програмного продукту**

Гуртову ціну програмного продукту визначаємо за формулою

|  |
| --- |
| ЦРЕАЛ = СПОВ +Рн , |

де, СПОВ — повна собівартість виробу, грн.;

РН — нормативний рівень рентабельності, %.

В нашому випадку РН приймаємо в розмірі 25 % від повної собівартості програмного продукту.

Рн= СПОВ × (РН / 100)

Рн= 21 804,14 × 0,25 = 5451,14 грн.

ЦРЕАЛ = 21 804,14 + 5451 = 27255 грн.

## **2.7 Висновки економічної доцільності проєкту**

В результаті впровадження програмний продукт забезпечить можливість застосування ефективних методів і засобів роботи на підприємстві.

За економічними підрахунками повна собівартість проєкту складає 21 804,14 грн., а гуртова ціна програмного продукту – 27255,14 грн.

# 3 ОХОРОНА ПРАЦІ

## **Загальні вимоги охорони праці при роботі з ЕОМ**

**Потенційні небезпеки фізичного характеру:**

* Перевантаження очей: Яскравий інтенсивний світло екрану може спричинити втомленість та напругу очей. Це може призвести до головного болю, блювоти, запаморочення та інших проблем.
* Біль у спині: Поодинокі позиції протягом тривалого часу можуть спричинити біль у спині. Наприклад, довгий час сидіння без опори для спини може призвести до болю та напруги в області спини та шиї.
* Варикозна хвороба: Сидячий спосіб життя, пов’язаний з роботою за комп’ютером, може спричинити варикозну хворобу - хворобу, яка характеризується поширеними венами в нижній частині ніг[9].

**Потенційні небезпеки хімічного характеру:**

* Використання хімічних засобів для прибирання комп’ютерних пристроїв може містити токсичні речовини, які можуть бути шкідливі для здоров’я.
* Тонер для лазерних принтерів може містити токсичні речовини, які можуть бути вдихнуті під час заміни картриджів або викидатися в повітря під час роботи принтера.
* Розбирання комп’ютерних компонентів може виділяти пил та інші частки в повітря, які можуть бути шкідливі для здоров’я[10].

**Потенційні небезпеки психофізіологічного характеру:**

* Стрес, що виникає внаслідок високого темпу роботи, незавершення завдань вчасно, тисків з боку керівництва або співробітників.
* Втома від тривалої роботи за ПК, що може призвести до зниження продуктивності та негативно позначитися на здоров’ї.
* Проблеми зі здоров’ям очей, які можуть виникнути внаслідок напруження очей, невірної позиції голови та незручного освітлення.
* Зниження якості сну та невідповідний режим дня, що може виникнути внаслідок роботи в нічний час або нерегулярного розподілу робочого часу[11].

**Потенційні небезпеки, що пов’язані з порушеннями санітарно-гігієнічних умов:**

* Підвищений рівень пилу та інших частинок у повітрі, що може викликати алергічні реакції, проблеми зі здоров’ям дихальних шляхів та інші захворювання.
* Недостатня вентиляція приміщення, що може призвести до накопичення шкідливих речовин, таких як вуглекислий газ, формальдегід та інші.
* Неправильна організація робочого місця, яка може призвести до невірної позиції тіла під час роботи за ПК, що може призвести до захворювань, таких як біль у спині, затяжні головні болі, м’язові напруги та інші[11].

**Потенційні небезпеки, що пов’язані з порушеннями правил пожежної безпеки:**

* Перегрів блоку живлення - може призвести до загоряння кабелів та елементів блоку живлення.
* Перегрів процесора - може призвести до аварії комп’ютера та виникнення пожежі.
* Неправильне підключення та використання електричних розеток - може призвести до перевантаження електричної мережі та виникнення пожежі[12].

### Характеристика приміщення

Значною частиною виробничої структури підприємства є робоче місце або робоча зона, і від її чіткої та зрозумілої організації залежить ефективність та безпека працівників на робочому місці. Саме тому організація та стан робочих місць є важливими елементами, які повинні забезпечувати зручність і безпеку роботи з матеріалами та пристроями.

Дано приміщення з параметрами a = 9,5 м, b = 5,6 м, h = 4,2 м. Перед визначенням кількості робочих місць, які можна розмістити у приміщенні, слід визначити його площу та об’єм:

Sпр= a \* b = 9,5 \* 5,6 = 53,2 м2

Vпр =a \* b \* h =9,5\* 5,6 \* 4,2 = 223,44 м3

Також, потрібно враховувати розміри меблів на робочих місцях, зокрема робочого столу (див. таблицю 1.1).

Відповідно до НПАОП 0.00.1.28-10 рекомендовані розміри столу для робочого місця з ВДТ становлять:

* висота — 725 мм;
* ширина — 600-1400 мм;
* глибина — 800-1000 мм.

Згідно із зазначеними рекомендаціями необхідно вказати розміри робочих столів.

Таблиця 3.1 Параметри устаткування

|  |  |  |  |
| --- | --- | --- | --- |
| Параметри устаткування | Стіл | Крісло | Шафа |
| Довжина, (м) | 1,2 | 0,45 | 0,6 |
| Ширина, (м) | 0,9 | 0,45 | 0,4 |
| Висота, (м) | 0,725 | 0,4 | 1,8 |
| Займана площа, (м2) | 1,08 | 0,2025 | 0,24 |
| Займаний обсяг (м3) | 0,783 | 0,081 | 0,432 |
| Площа устаткування, яке розташоване на одному робочому місці Sуст. (м2) | Sуст. = Sстола+ Sкрісла + ½·Sшафи = | 1,4025 | |
| Обсяг, устаткування, яке розташоване на одному робочому місці Vуст. (м3) | Vуст. = Vстола+ Vкрісла + ½·Vшафи = | 1,08 | |

Знайдемо максимально можливу кількість комп’ютеризованих робочих місць, яку можна розмістити в заданому приміщенні за умови, що на одне таке робоче місце повинно доводитися не менш ніж 6 м2 вільної площі та 20 м3 за обсягом, не зайнятим устаткуванням:

Nр.м. по S = S / (Sуст. + 6) = 53,2 / (1,4025 + 6) = 7,19 (р.м.);

Nр.м. по V = V / (Vуст. + 20) = 223,44 / (1,08 + 20) = 10,6 (р.м.).

З отриманих результатів приймаємо мінімальне значення, округляючи його до найближчого меншого цілого числа, а саме Nр.м. по S = 7 (р.м.).

На рисунку 1.9 наведено схему розташування семи комп’ютеризованих робочих місць з параметрами приміщення а = 9,5 м, b = 5,6 м, h = 4,2 м із врахуванням наступних вимог:

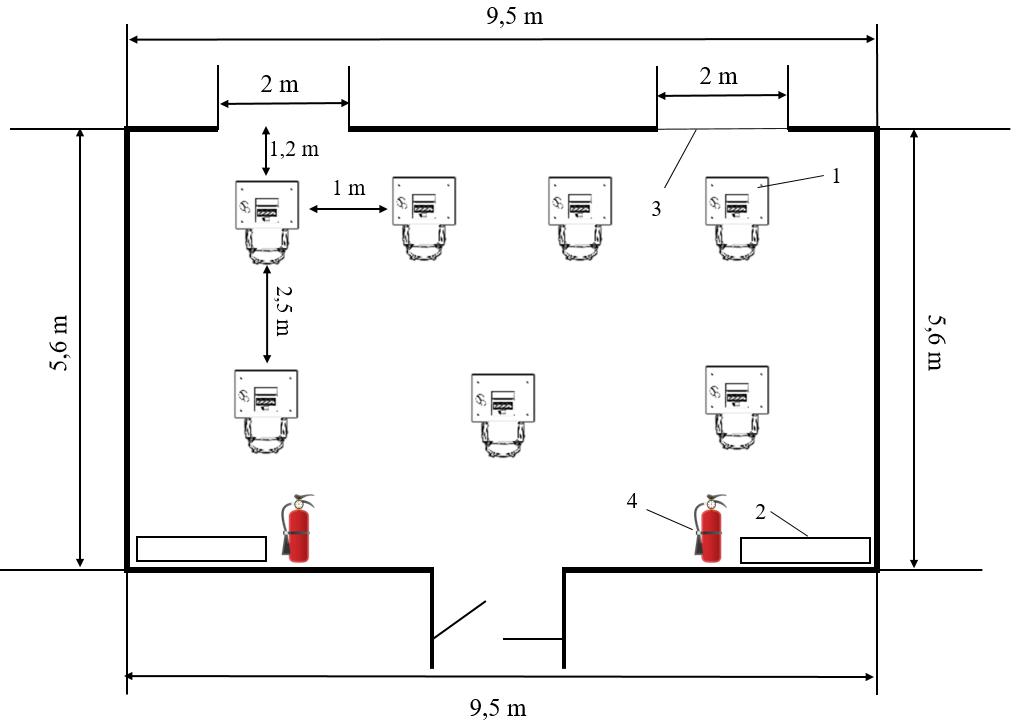


Рисунок 3.1 План виробничого приміщення з комп’ютеризованими робочими місцями: — комп’ютеризоване робоче місце з ВДТ, 2 — шафи для зберігання документації, 3 – сонцезахисні жалюзі, 4 — вогнегасник.

Найкраще розмістити комп’ютеризовані робочі місця рядами вздовж стіни з вікнами. Це дасть змогу виключити дзеркальне відбиття на екрані ВДТ джерел природного світла та їх потрапляння у поле зору операторів, що погіршує умови їх зорової роботи. Але треба при цьому брати до уваги ширину приміщення, при достатньому її значенні можливе розташування комп’ютеризованих робочих місць, окрім зазначеного вище, і посередині приміщення з метою оптимального використання його площі[13].

## **Заходи щодо поліпшення умов з охорони праці**

**Організаційні заходи:**

* Розробити та впровадити програму охорони праці та виробничої санітарії, що включає в себе перелік дій для забезпечення безпеки працівників та виробничого середовища.
* Проводити регулярні інструктажі для працівників з питань охорони праці та виробничої санітарії, щоб забезпечити їхню свідомість щодо правил та процедур безпеки.
* Встановити систему контролю та відслідковування дотримання правил безпеки на робочому місці. Це може бути здійснено шляхом використання відеокамер, інших систем моніторингу та звітів про безпеку працівників.
* Проводити регулярні перевірки умов роботи та виробничого середовища для виявлення можливих небезпек та проблем з охороною праці.
* Забезпечити необхідні засоби індивідуального захисту для працівників, наприклад, захисні маски, рукавиці, окуляри тощо.
* Встановити правила та процедури щодо збору, зберігання та видалення відходів та інших матеріалів, які можуть створювати небезпеку для здоров’я працівників.
* Запровадити систему нагород та заохочень для працівників, які дотримуються правил безпеки та допомагають у забезпеченні безпеки на робочому місці[14].

**Технічні заходи:**

* Встановлення системи вентиляції з повітряним фільтром, що дозволить підтримувати належну якість повітря та уникнути забруднення виробничого середовища.
* Встановлення світлових приладів з найкращою якістю освітлення, щоб уникнути втоми очей та нещасних випадків.
* Установлення автоматичних датчиків для регулювання температури та вологості, що дозволить забезпечити комфортні умови для працівників.
* Встановлення сигналізації для швидкого виявлення небезпек, наприклад, вогню, викиду шкідливих речовин.
* Переведення виробничих процесів на безпечніші матеріали та технології з метою зменшення ризику виникнення травм та отруєнь.
* Установлення спеціального обладнання та приладів для забезпечення безпеки при здійсненні різних операцій, наприклад, системи захисту від ножових порізів або приладів для безпечного підйому важких предметів.
* Встановлення пристроїв для зменшення шуму та вібрації, що допоможе попередити пошкодження слуху та інші проблеми здоров’я у працівників[15].

## **Інструкція з охорони праці при роботі з ПЕОМ та відео-дисплейними терміналами**

### Загальні положення

3.4.1.1. Дія інструкції поширюється на всі підрозділи підприємства.

3.4.1.2. Інструкція розроблена на основі ДНАОП 0.00-8.03-93 «Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві», ДНАОП 0.00-4.15-98 «Положення про розробку інструкцій з охорони праці», ДНАОП 0.00-4.12-99 «Типове положення про навчання з питань охорони праці», ДНАОП 0.00-1.31-99 “Правила охорони праці під час експлуатації електронно-обчислювальної техніки”.

3.4.1.3. За даною інструкцією оператор (користувач) персонального комп’ютеру (відеодисплейного терміналу) інструктується перед початком роботи (первинний інструктаж), а потім через кожні 6 місяців (повторний інструктаж).

Результати інструктажу заносяться в «Журнал реєстрації інструктажів з питань охорони праці», в журналі після проходження інструктажу повинен бути підпис особи, яка інструктує, та оператора (користувача).

3.4.1.4. Власник повинен застрахувати оператора від нещасних випадків та професійних захворювань.

В разі пошкодження здоров’я оператора (користувача) з вини власника, він (оператор) має право на відшкодування заподіяної йому шкоди.

3.4.1.5. За невиконання даної інструкції оператор несе дисциплінарну, матеріальну, адміністративну та кримінальну відповідальність.

3.4.1.6. До роботи на персональної електронно-обчислювальної машині (ПЕОМ) або відеодисплейному терміналі (ВДТ) допускаються особи, які пройшли спеціальне навчання, медичний огляд, вступний інструктаж з охорони праці, інструктаж на робочому місці та інструктаж з питань пожежної безпеки.

3.4.1.7. Оператор (користувач) повинен:

3.4.1.7.1. Виконувати правила внутрішнього трудового розпо­рядку.

3.4.1.7.2. Не допускати в робочу зону сторонніх осіб.

3.4.1.7.3. Не виконувати вказівок, які суперечать правилам охорони праці.

3.4.1.7.4. Пам’ятати про особисту відповідальність за виконання правил охорони праці та безпеку товаришів по роботі.

3.4.1.7.5. Вміти надавати першу медичну допомогу потерпілим від нещасних випадків.

3.4.1.7.6. Вміти користуватись первинними засобами пожежегасіння.

3.4.1.7.7. Виконувати правила особистої гігієни.

3.4.1.8. Основні небезпечні і шкідливі виробничі фактори, що можуть впливати на оператора (користувача):

а) фізичні:

* підвищений рівень електромагнітного випромінювання;
* підвищений рівень рентгенівського випромінювання;
* підвищений рівень ультрафіолетового випромінювання;
* підвищений рівень інфрачервоного випромінювання;
* підвищений рівень статичної електрики;
* підвищений рівень запиленість повітря робочої зони;
* підвищений зміст позитивних аероіонів у повітрі робочої зони;
* знижений вміст негативних аероіонів у повітрі робочої зони;
* знижена чи підвищена вологість повітря робочої зони;
* знижена чи підвищена рухомість повітря робочої зони;
* підвищений рівень шуму на робочому місці (від вентиляторів, процесорів, аудіоплат, принтерів);
* підвищений чи знижений рівень освітленості;
* підвищений рівень прямої і відбитої блискісті;
* підвищений рівень засліпленості;
* нерівномірність розподілу яскравості в поле зору;
* підвищена яскравість світлового зображення;
* підвищений рівень пульсації світлового потоку;
* ураження електричним струмом;

б) хімічні:

підвищений зміст у повітрі робочої зони двоокису вуглецю, озону, аміаку, фенолу, формальдегіду;

в) психофізіологічні:

* напруга зору;
* напруга уваги;
* інтелектуальні навантаження;
* емоційні навантаження;
* тривалі статичні навантаження;
* монотонність праці;
* великий обсяг інформації, оброблюваної в одиницю часу;
* нераціональна організація робочого місця;

г) біологічні:

підвищений вміст у повітрі робочої зони мікроорганізмів.

3.4.1.9. Під час вибору приміщення для розміщення робочих місць ПЕОМ або ВДТ необхідно враховувати, що вікна можуть давати блискість на екранах дисплеїв і викликати значне осліплення в тих, хто сидить перед ними, особливо влітку та в сонячні дні.

3.4.1.10. Приміщення з ПЕОМ та ВДТ повинні мати природне і штучне освітлення. При незадовільному освітленні знижується продуктивність праці оператора ПЕОМ (ВДТ), можливі короткозорість, швидка втомленість.

3.4.1.11. Організація робочого місця користувача ПЕОМ та ВДТ повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ГОСТ 12.2.032 “ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования”; характеру та особливостям трудової діяльності.

3.4.1.12. Не допускається розташування робочих місць ПЕОМ в підвальних приміщеннях і цокольних поверхах.(рисунок 3.10)

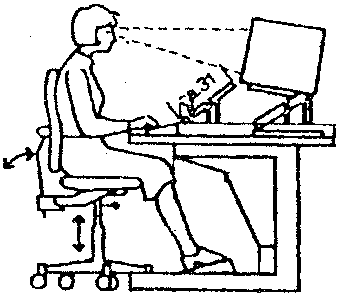


Рисунок 3.10 Правильне розташування робочих місць

3.4.1.13. Робочі місця з ВДТ і ПЕОМ під час виконання творчої роботи, яка потребує значної розумової напруги чи великої концентрації уваги, слід ізолювати одне від одного перегородкою висотою 1,5 2,0 м.

3.4.1.14. Неприпустимим є розташування приміщень категорій А і Б (ОНТП 24 86), а також виробництв з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ЕОМ, виконується їх обслуговування, налагодження і ремонт, а також над такими приміщеннями або під ними. Виробничі приміщення, в яких розташовані ЕОМ, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо).

3.4.1.15. Робочі місця з ВДТ рекомендується розміщувати в окремих приміщеннях. В разі розміщення робочих місць з ВДТ в залах або приміщеннях з джерелами небезпечних і шкідливих факторів вони повинні розташовуватись у повністю ізольованих кабінетах з природним освітленням та організованим повітрообміном[16].

### Вимоги безпеки перед початком роботи

3.4.2.1. Увімкнути систему кондиціювання повітря в приміщенні.

3.4.2.2. Оглянути робоче місце і привести його в порядок; впевнитись, що на ньому відсутні сторонні предмети; все обладнання і блоки ПЕОМ з’єднані з системним блоком за допомогою з’єднувальних шнурів.

3.4.2.3. Перевірити надійність встановлення апаратури на робочому столі. ВДТ має стояти не на краю стола. Повернути ВДТ так, щоб було зручно дивитися на екран під прямим кутом (а не збоку) і трохи зверху вниз; при цьому екран має бути трохи нахиленим нижній його край ближче до оператора (користувача).

3.4.2.4. Перевірити загальний стан апаратури, перевірити справність електропроводки, з’єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана.

3.4.2.5. Забороняється:

* експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією; залишення під напругою кабелів та проводів з неізольованими провідниками;
* застосування саморобних подовжувачів, які не відповідають вимогам Правил будови електроустановок до переносних електропроводок;
* застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарювання;
* користування пошкодженими розетками, з’єднувальними коробками, вимикачами, а також лампами, скло яких має сліди затемнення або випинання;
* підвішування світильників безпосередньо на струмопровідних проводах, обгортання електроламп і світильників папером, тканиною та іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);
* використання електроапаратури та приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств виготовлювачів.

3.4.2.6. Відрегулювати освітленість робочого місця.

3.4.2.7. Відрегулювати та зафіксувати висоту крісла, зручний для оператора (користувача) нахил його спинки.

3.4.2.8. У разі необхідності приєднати до процесора необхідну апаратуру (принтер, сканер тощо).

Усі кабелі, що з’єднують системний блок (процесор) з іншими пристроями, слід вмикати та вимикати тільки при вимкненому комп’ютері.

3.4.2.9. Ввімкнути апаратуру комп’ютера вимикачами на корпусах в послідовності: стабілізатор напруги, ВДТ, процесор, принтер (якщо передбачається друкування).

3.4.2.10. Відрегулювати яскравість свічення екрана ВДТ, мінімальний розмір світної точки, фокусування, контрастність.

Не слід робити зображення занадто яскравим, щоб не втомлювати очей.

Рекомендується:

 яскравість свічення екрана не менше 100 кд/м;

 відношення яскравості екрана ВДТ до яскравості оточуючих його поверхонь в робочій зоні не більше 3:1;

 мінімальний розмір точки свічення не менше 0,4 мм для монохромного ВДТ і не менше 0,6 мм для кольорового;

 контрастність зображення знаку не менше 0,8.

3.4.2.11. У разі виявлення будь яких несправностей роботу не розпочинати, повідомити про це керівника робіт[16].

### Вимоги безпеки під час роботи

3.4.1.1. Необхідно стійко розташувати клавіатуру на робочому столі, не допускаючи її хитання.

Разом з тим має бути передбачена можливість її поворотів та переміщень.

Положення клавіатури та кут її нахилу мають відповідати побажанням оператора (користувача).

Якщо в конструкції клавіатури не передбачено простору для опори долонь, то їх слід розташовувати на відстані не менше 100 мм від краю столу в оптимальній зоні моторного поля.

Під час роботи на клавіатурі сидіти прямо, не напружуватися.

3.4.1.2. Для зменшення несприятливого впливу на оператора (користувача) пристроїв типу «миша» (вимушена поза, необхідність постійного контролю за якістю дій) належить забезпечити вільну велику поверхню столу для переміщення «миші» і зручного упору ліктьового суглоба.

3.4.1.3. Не дозволяються сторонні розмови, подразнюючі шуми.

3.4.1.4. Періодично при вимкненому комп’ютері прибирати ледь змоченою мильним розчином бавовняною ганчіркою пил з поверхонь апаратури.

**Забороняється:**

3.4.5.1. Самостійно ремонтувати апаратуру, тим більше ВДТ, в якому кінескоп знаходиться під високою напругою (близько 25 кВ); ремонт апаратури виконують тільки спеціалісти з технічного обслуговування комп’ютерів, вони ж раз на півроку повинні відкривати процесор і вилучати пилососом пил і бруд, що там накопились.

3.4.5.2. Класти будь які предмети на апаратуру комп’ютера, напої на клавіатуру або поруч з нею це може вивести їх з ладу.

3.4.1.6. Для зняття статичної електрики рекомендується час від часу, доторкатися до металевих поверхонь (батарея центрального опалення тощо).

3.4.1.7. Для зниження напруженості праці на ПЕОМ необхідно рівномірно розподіляти і чергувати характер робіт відповідно до їх складності.

З метою зменшення негативного впливу монотонії доцільно застосувати чергування операцій введення тексту та числових даних (зміна змісту робіт), чергування редагування текстів та введення даних (зміна змісту та темпу роботи) і т. ін.[16].

### Вимоги безпеки після закінчення роботи

3.4.4.1. Закінчити та записати у пам’ять комп’ютера файл, що знаходиться в роботі. Вийти з програмної оболонки і повернутися в середовище MS DOS.

3.4.4.2. Вимкнути принтер, інші периферійні пристрої, вимкнути ВДТ і процесор. Вимкнути стабілізатор, якщо комп’ютер підключений до мережі через нього. Штепсельні вилки витягнути з розеток. Накрити клавіатуру кришкою для запобігання попаданню в неї пилу.

3.4.4.3. Прибрати робоче місце. Оригінали та інші документи покласти в ящик стола.

3.4.4.4. Ретельно вимити руки теплою водою з милом.

3.4.4.5. Вимкнути кондиціонер, освітлення і загальне електроживлення підрозділу.

3.4.4.6. Рекомендується в спеціально обладнаному приміщенні провести сеанс психофізіологічного розвантаження і зняття втоми з виконанням спеціальних вправ аутотренінгу.

### Вимоги безпеки в аварійних ситуаціяї

3.4.5.1. Аварійна ситуація чи нещасний випадок можуть статися в разі: ураження електричним струмом, загорання апаратури тощо.

3.4.5.2. У разі раптового припинення подавання електроенергії вимкнути комп’ютер в такій послідовності: периферійні пристрої, ВДТ, процесор, стабілізатор напруги, витягнути штепсельні вилки з розеток.

3.4.5.3. При виявленні ознак горіння (дим, запах гару), вимкнути апаратуру, знайти джерело займання і вжити заходів щодо його ліквідації, повідомити керівника робіт, не допускати в небезпечну зону сторонніх осіб.

3.4.5.4. Якщо є потерпілі, надавати їм першу медичну допомогу, при необхідності, викликати швидку медичну допомогу.

# ВИСНОВКИ

Під час проєкту з сервісу конвертування файлів я здобув значний досвід та набув розуміння важливості такого сервісу в сучасному світі. Ось деякі зроблені мною висновки:

Зростання популярності: З огляду на зростання кількості електронних документів та різноманітних форматів, я зрозумів, що сервіс конвертування файлів є вельми актуальним і вимаганим. Користувачі постійно шукають зручний спосіб конвертувати файли з одного формату в інший, зберігаючи при цьому їх якість та форматування.

Різноманітність форматів: Процес конвертації документів включає в себе багато різних форматів, таких як PDF, DOCX, JPG, PNG та багато інших. Це означає, що сервіс конвертування файлів повинен підтримувати широкий спектр форматів і забезпечувати надійну та точну конвертацію між ними.

Зручність та доступність: Часто користувачам потрібно швидко та легко змінити формат файлу. Важливо мати простий та інтуїтивно зрозумілий інтерфейс для завантаження та конвертації файлів без зайвих зусиль. Крім того, сервіс повинен бути доступний з будь якого пристрою з Інтернет підключенням, щоб користувачі могли користуватися ним у будь який час та з будь якого місця.

Збереження якості та форматування: Під час конвертації важливо забезпечити збереження якості та форматування файлів. Наприклад, при конвертації текстових документів важливо зберегти оригінальну структуру, шрифти та розмітку, щоб отриманий файл був легким у використанні та читанні.

Безпека та конфіденційність: У світі, де безпека даних є пріоритетом, важливо, щоб сервіс конвертування файлів гарантував захист конфіденційної інформації користувачів. Це може включати захист від несанкціонованого доступу, шифрування даних та безпечну передачу файлів через захищені протоколи.

Постійне оновлення та розвиток: Технологічний прогрес швидко змінюється, і нові формати файлів виникають. Тому сервіс конвертування файлів повинен постійно оновлюватися та розвиватися, щоб підтримувати нові формати і задовольняти зростаючі потреби користувачів.

Загалом, переддипломна практика з сервісу конвертування файлів підтвердила важливість такого сервісу в цифровому світі. Виконання всіх цих аспектів допоможе розробити ефективний сервіс, який задовольнить потреби користувачів та забезпечить зручну та надійну конвертацію файлів у різних форматах.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке конвертер файлів . URL: http://hi news.pp.ua/kompyuteri/14018 scho take konverter faylv viznachennya priznachennya ta vikoristannya v ofsny robot.html (дата звернення: 17.05.2023)
2. Онлайн сервіс https://convertio.co/ (дата звернення: 17.05.2023)
3. Онлайн сервіс https://www.online-convert.com/ (дата звернення: 17.05.2023)
4. Мова розмітки HTML. URL: <http://www.znannya.org/?view=html>. (Дата звернення: 17.05.2023);
5. Таблиці стилів CSS URL: < https://w3schoolsua.github.io/html/html\_css.html> (дата звернення 17.05.2023)
6. javascript мова програмування. URL: <https://uk.wikipedia.org/wiki/JavaScript>. (дата звернення:17.05.2023);
7. Фреймворк Django. URL: < https://uk.wikipedia.org/wiki/Django >. (дата звернення: 17.05.2023);
8. Typescript URL:< https://www.w3schools.com/typescript/typescript\_intro.php>. (Дата звернення 17.05.2023); (дата звернення: 17.05.2023);
9. Керівник з охорони праці та безпеки. (2017). Робота з екранним обладнанням: Правила охорони здоров’я та безпеки (екранне обладнання) 1992 року. [онлайн] Доступно за посиланням URL : https://www.hse.gov.uk/pubns/indg36.pdf (дата звернення: 17.05.2023);
10. Правила охорони праці України ПОП 01 14.1 15 URL: https://zakon.rada.gov.ua/laws/show/z1419 11#Text (дата звернення: 17.05.2023);
11. Occupational Safety and Health Administration. (2004).Режим доступу:<https://www.osha.gov/SLTC/etools/computerworkstations/evaluation_checklist.html>. (дата звернення: 17.05.2023);
12. Рекомендації з охорони праці при роботі за комп’ютером від Міністерства охорони здоров’я та соціальної політики України [https://moz.gov.ua/article/health/ohorona praci pri roboti za kompjuterom](https://moz.gov.ua/article/health/ohorona-praci-pri-roboti-za-kompjuterom) (дата звернення: 17.05.2023);
13. Наказ Держпраці України «Про затвердження типових норм витрат на виробництво продукції (виконання робіт, надання послуг)»: https://sfm.illinois.gov/(дата звернення: 17.05.2023);
14. Щербан І.І. Методичні вказівки до виконання розділу «Охорона праці» в дипломних роботах (проєктах) для студентів галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення» за освітньо кваліфікаційним рівнем «молодший спеціаліст». Ужгород, 2023 27с. (дата звернення: 17.05.2023);
15. Державний санітарний нагляд за дотриманням вимог санітарних правил та норм на виробництві (<http://www.dsns.gov.ua/>) (дата звернення: 17.05.2023);
16. Державні норми та правила з охорони праці ці норми та правила містять детальну інформацію про заходи з поліпшення умов праці. (дата звернення: 17.05.2023);
17. Інструкція з охорони праці під час робіт на персональному комп’ютері і відеодисплейних терміналах URL: https://dnaop.com/html/31947/doc instrukcijaz ohoroni pracipid chas robit na personalynomu kompjuterii videodisplejnih terminalah (дата звернення: 17.05.2023).

Додаток. Лістинг програми

**Front**

**Base.css**

/\*colorpalettefrom<https://github.com/vuejs/theme>\*/

:root{

--vt-c-white:#ffffff;

--vt-c-white-soft:#f8f8f8;

--vt-c-white-mute:#f2f2f2;

--vt-c-black:#181818;

--vt-c-black-soft:#222222;

--vt-c-black-mute:#282828;

--vt-c-indigo:#2c3e50;

--vt-c-divider-light-1:rgba(60,60,60,0.29);

--vt-c-divider-light-2:rgba(60,60,60,0.12);

--vt-c-divider-dark-1:rgba(84,84,84,0.65);

--vt-c-divider-dark-2:rgba(84,84,84,0.48);

--vt-c-text-light-1:var(--vt-c-indigo);

--vt-c-text-light-2:rgba(60,60,60,0.66);

--vt-c-text-dark-1:var(--vt-c-white);

--vt-c-text-dark-2:rgba(235,235,235,0.64);

}

/\*semanticcolorvariablesforthisproject\*/

:root{

--color-background:var(--vt-c-white);

--color-background-soft:var(--vt-c-white-soft);

--color-background-mute:var(--vt-c-white-mute);

--color-border:var(--vt-c-divider-light-2);

--color-border-hover:var(--vt-c-divider-light-1);

--color-heading:var(--vt-c-text-light-1);

--color-text:var(--vt-c-text-light-1);

--section-gap:160px;

}

@media(prefers-color-scheme:dark){

:root{

--color-background:var(--vt-c-black);

--color-background-soft:var(--vt-c-black-soft);

--color-background-mute:var(--vt-c-black-mute);

--color-border:var(--vt-c-divider-dark-2);

--color-border-hover:var(--vt-c-divider-dark-1);

--color-heading:var(--vt-c-text-dark-1);

--color-text:var(--vt-c-text-dark-2);

}

}

\*,

\*::before,

\*::after{

box-sizing:border-box;

margin:0;

position:relative;

font-weight:normal;}

body{

min-height:100vh;

color:var(--color-text);

background:var(--color-background);

transition:color0.5s,background-color0.5s;

line-height:1.6;

font-family:Inter,-apple-system,BlinkMacSystemFont,’SegoeUI’,Roboto,Oxygen,Ubuntu,

Cantarell,’FiraSans’,’DroidSans’,’HelveticaNeue’,sans-serif;

font-size:15px;

text-rendering:optimizeLegibility;

-webkit-font-smoothing:antialiased;

-moz-osx-font-smoothing:grayscale;

}

**Logo.svg**

<svgxmlns=«http://www.w3.org/2000/svg»viewBox=«00261.76226.69»><pathd=«M161.096.001l-30.22552.351L100.647.001H-.005l130.877226.688L261.749.001z»fill=«#41b883»/><pathd=«M161.096.001l-30.22552.351L100.647.001H52.346l78.526136.01L209.398.001z»fill=«#34495e»/></svg>

**Main.css**

@import’./base.css’;

#app{

max-width:1280px;

margin:0auto;

padding:2rem;

font-weight:normal;

}

a,

.green{

text-decoration:none;

color:hsla(160,100%,37%,1);

transition:0.4s;

}

@media(hover:hover){

a:hover{

background-color:hsla(160,100%,37%,0.2);

}

}

@media(min-width:1024px){

body{

display:flex;

place-items:center;

}

#app{

display:grid;

grid-template-columns:1fr1fr;

padding:02rem;

}

}

**NavBar.vue**

<script lang=«ts»>

import { ref } from «vue»;

import { generateTabNavigation } from ‘../core/navigation’;

export default {

  setup() {

    const items = ref(generateTabNavigation());

    return { items };

  },

};

</script>

<template>

  <div>

    <MenuBar :model=«items»>

      <template #start>

        <img

          alt=«logo»

          src=«https://c4.wallpaperflare.com/wallpaper/206/351/660/anime-girls-black-hair-long-hair-purple-eyes-cute-little-girl-hd-wallpaper-preview.jpg»

          height=«40»

          class=«mr-2»

        />

      </template>

    </MenuBar>

  </div>

</template>

<style lang=«css»>

</style>

**UploadFile.vue**

<script lang=«js»>

import { UPLOAD\_URI } from ‘../constants/server.ts’;

import { mapStores } from ‘pinia’

import { userStore } from ‘../stores/userstore.ts’

import axios from ‘axios’;

import { saveAs } from ‘file-saver’

export default {

  props: [‘fromFormat’, ‘toFormat’],

  computed: {

    ...mapStores(userStore)

  },

  data() {

    return {

      UPLOAD\_URI: UPLOAD\_URI,

      uploadedFiles: [],

      files: [],

      totalSize: 0,

      totalSizePercent: 0,

    };

  },

  methods: {

    uploader(event) {

      const files = event.files;

      const formData = new FormData();

      for (const file of files) {

        formData.append(«files», file);

      }

      axios({

        method: ‘post’,

        url: UPLOAD\_URI,

        data: formData,

        headers: {

          ‘Accept’: ‘application/json’,

          ‘Content-Type’: ‘multipart/form-data’,

          ‘x-User-UUID’: this.userStore.getUUID,

          ‘x-from-format’: this.fromFormat,

          ‘x-to-format’: this.toFormat,

          ‘Access-Control-Allow-Origin’: ‘\*’

        },

        responseType: ‘arraybuffer’

      }).then(response => {

        const blob = new Blob([response.data], {type: ‘application/x-zip-compressed’});

        saveAs(blob, «upload.zip»)

        this.$emit(‘uploadedEvent’)

      })

    },

    onRemoveTemplatingFile(file, onFileRemove, index) {

      onFileRemove(index);

      this.totalSize -= parseInt(this.formatSize(file.size));

      this.totalSizePercent = this.totalSize / 10;

    },

    onClearTemplatingUpload(clear) {

      clear();

      this.totalSize = 0;

      this.totalSizePercent = 0;

    },

    onSelectedFiles(event) {

      this.files = event.files;

      this.files.forEach((file) => {

        this.totalSize += parseInt(this.formatSize(file.size));

      });

    },

    onAdvancedUpload() {

      this.$toast.add({

        severity: «info»,

        summary: «Success»,

        detail: «File Uploaded»,

        life: 3000,

      });

    },

    uploadEvent(callback) {

      this.totalSizePercent = this.totalSize / 10;

      callback();

    },

    onTemplatedUpload(event) {

      console.log(event);

      this.totalSize = 0;

      this.totalSizePercent = 0;

      this.$toast.add({

        severity: «info»,

        summary: «Success»,

        detail: «File Uploaded»,

        life: 3000,

      });

    },

    onUpload() {

      this.$toast.add({

        severity: «info»,

        summary: «Success»,

        detail: «File Uploaded»,

        life: 3000,

      });

    },

    formatSize(bytes) {

      if (bytes === 0) {

        return «0 B»;

      }

      let k = 1000,

        dm = 3,

        sizes = [«B», «KB», «MB», «GB», «TB», «PB», «EB», «ZB», «YB»],

        i = Math.floor(Math.log(bytes) / Math.log(k));

      return parseFloat((bytes / Math.pow(k, i)).toFixed(dm)) + « « + sizes[i];

    },

  },

};

</script>

<template>

  <Toast />

  <FileUpload

    name=«demo[]»

    :customUpload=«true»

    :url=«UPLOAD\_URI»

    :multiple=«true»

    accept=««

    :maxFileSize=«1000000000»

    @uploader=«uploader»

    @upload=«onTemplatedUpload($event)»

    @select=«onSelectedFiles»

  >

    <template

      #header=«{ chooseCallback, uploadCallback, clearCallback, files }»

    >

      <div

        class=«flex flex-wrap justify-content-between align-items-center flex-1 gap-2»

      >

        <div class=«flex gap-2»>

          <Button

            @click=«chooseCallback()»

            icon=«pi pi-download»

            class=«p-button-rounded»

          ></Button>

          <Button

            @click=«uploadEvent(uploadCallback)»

            icon=«pi pi-cloud-upload»

            class=«p-button-rounded p-button-success»

            :disabled=«!files || files.length === 0»

          ></Button>

          <Button

            @click=«clearCallback()»

            icon=«pi pi-times»

            class=«p-button-rounded p-button-danger»

            :disabled=«!files || files.length === 0»

          ></Button>

        </div>

      </div>

    </template>

    <template

      #content=«{

        files,

        uploadedFiles,

        removeUploadedFileCallback,

        fileRemoveCallback,

      }»

    >

      <div v-if=«files.length > 0»>

        <h5>Pending</h5>

        <div class=«flex flex-wrap p-5 gap-5»>

          <div

            v-for=«(file, index) of files»

            :key=«file.name + file.type + file.size»

            class=«card m-0 px-6 flex flex-column border-1 surface-border align-items-center gap-3»

          >

            <span class=«font-semibold»>{{ file.name }}</span>

            <div>{{ formatSize(file.size) }}</div>

            <Badge value=«Pending» severity=«warning» />

            <Button

              icon=«pi pi-times»

              @click=«onRemoveTemplatingFile(file, fileRemoveCallback, index)»

              class=«p-button-outlined p-button-danger p-button-rounded»

            />

          </div>

        </div>

      </div>

      <div v-if=«uploadedFiles.length > 0»>

        <h5>Completed</h5>

        <div class=«flex flex-wrap p-0 sm:p-5 sm:p-5 gap-5»>

          <div

            v-for=«(file, index) of uploadedFiles»

            :key=«file.name + file.type + file.size»

            class=«card m-0 px-6 flex flex-column border-1 surface-border align-items-center gap-3»

          >

            <div>

              <img

                role=«presentation»

                :alt=«file.name»

                :src=«file.objectURL»

                width=«100»

                height=«50»

                class=«shadow-2»

              />

            </div>

            <span class=«font-semibold»>{{ file.name }}</span>

            <div>{{ formatSize(file.size) }}</div>

            <Badge value=«Completed» class=«mt-3» severity=«success» />

            <Button

              icon=«pi pi-times»

              @click=«removeUploadedFileCallback(index)»

              class=«p-button-outlined p-button-danger p-button-rounded»

            />

          </div>

        </div>

      </div>

    </template>

    <template #empty>

      <div class=«flex align-items-center justify-content-center flex-column»>

        <i

          class=«pi pi-cloud-upload border-2 border-circle p-5 text-8xl text-400 border-400»

          style=«font-size: 20rem»

        ></i>

        <p class=«mt-4 mb-0»>Drag and drop files to here to upload.</p>

      </div>

    </template>

  </FileUpload>

</template>

<style lang=«css» scoped></style>

**Helpers.ts**

export function str2bytes (str: string) {

    const  bytes = new Uint8Array(str.length);

    for (let i=0; i<str.length; i++) {

        bytes[i] = str.charCodeAt(i);

    }

    return bytes;

}

**navigation.ts**

import type { ITabNavigation } from «@/interfaces/navigation/ITabNavigation»;

export function generateTabNavigation(): Array<ITabNavigation> {

  return [

      {

        label: «Home»,

        icon: «pi pi-fw pi-home»,

        to: { path: ‘/’}

      },

      {

        label: «PDF»,

        icon: «pi pi-fw pi-file-pdf»,

        items: [

          {

            label: «PDF to Word»,

            icon: «pi pi-fw pi-file»,

            to: { name: «pdf.word» }

          },

          {

            label: «PDF to Power Point»,

            icon: «pi pi-fw pi-file»,

            to: { name: «pdf.pptx» }

          },

          {

            label: «PDF to JPG»,

            icon: «pi pi-fw pi-file»,

            to: { name: «pdf.jpg» }

          },

          {

            label: «PDF to JPEG»,

            icon: «pi pi-fw pi-file»,

            to: { name: «pdf.jpeg» }

          },

          {

            label: «PDF to PNG»,

            icon: «pi pi-fw pi-file»,

            to: { name: «pdf.png» }

          },

        ],

      },

      {

        label: «Word»,

        icon: «pi pi-fw pi-file-word»,

        items: [

          {

            label: «PDF»,

            icon: «pi pi-fw pi-file»,

            to: { name: ‘word.pdf’ }

          },

          {

            label: «TXT»,

            icon: «pi pi-fw pi-file»,

            to: { name: ‘word.txt’ }

          }

        ],

      },

      {

        label: «Video»,

        icon: «pi pi-fw pi-video»,

        items: [

          {

                label: «MP3»,

                icon: «pi pi-volume-up»,

                to: { name: «mp4.mp3»}

              },

          {

            label: «Compress»,

            icon: «pi pi-fw pi-window-minimize»,

            to: { name: «mp4.mp4» }

          },

        ],

      },

      {

        label: «File»,

        icon: «pi pi-fw pi-file»,

        items: [

                    {

            label: «JSON»,

            icon: «pi pi-fw pi-file-edit»,

            items: [

              {

                label: «CSV»,

                icon: «pi pi-fw pi-file-edit»,

                to: { name: ‘json.csv’ }

              },

              {

                label: «XML»,

                icon: «pi pi-fw pi-file-edit»,

                to: { name: ‘json.xml’ }

              }

            ],

          },

          {

            label: «XML»,

            icon: «pi pi-fw pi-file-edit»,

            items: [

              {

                label: «CSV»,

                icon: «pi pi-fw pi-file-edit»,

                to: { name: ‘xml.csv’ }

              },

              {

                label: «JSON»,

                icon: «pi pi-fw pi-file-edit»,

                to: { name: ‘xml.json’ }

              },

            ],

          },

          {

            label: «CSV»,

            icon: «pi pi-fw pi-file-edit»,

            items: [

              {

                label: «JSON»,

                icon: «pi pi-fw pi-file-edit»,

                to: { name: ‘csv.json’ }

              },

              {

                label: «XML»,

                icon: «pi pi-fw pi-file-edit»,

                to: { name: ‘csv.xml’ }

              }

            ],

          },

          {

            label: «HTML»,

            icon: «pi pi-fw pi-file-edit»,

            items: [

              {

                label: «TXT»,

                icon: «pi pi-fw pi-file-edit»,

                to: { name: ‘html.txt’ }

              },

            ],

          }

        ],

      },

    ]

}

**toast.ts**

import type { ToastServiceMethods } from ‘primevue/toastservice’;

class Toast {

    toast: ToastServiceMethods

    constructor(toast: ToastServiceMethods) {

        this.toast = toast;

    }

    info(msg: string, summary=‘Info’) {

        this.toast.add({ severity: ‘info’, summary: summary, detail: msg, life: 3000 });

    }

    error(msg: string, summary=‘Error’) {

        this.toast.add({ severity: ‘error’, summary: summary, detail: msg, life: 3000 });

    }

    success(msg: string, summary=‘Success’) {

        this.toast.add({ severity: ‘success’, summary: summary, detail: msg, life: 3000 });

    }

    warning(msg: string, summary=‘Warning’) {

        this.toast.add({ severity: ‘warn’, summary: summary, detail: msg, life: 3000 });

    }

}

export function getToastService(toastService: ToastServiceMethods) {

    return new Toast(toastService);

}

**index.ts**

import { createRouter, createWebHistory } from «vue-router»;

import HomeView from «../views/HomeView.vue»;

import ConvertFileView from «../views/ConvertFileView.vue»;

import PickConverterView from «../views/PickConverterView.vue»;

import { FileFormats} from «@/constants/file-formats»;

const router = createRouter({

  history: createWebHistory(import.meta.env.BASE\_URL),

  routes: [

    {

      path: «/»,

      name: «home»,

      component: HomeView,

    },

    {

      path: «/upload/pdf/docx»,

      name: «pdf.word»,

      component: ConvertFileView,

      props: () => ({

        fromFormat: FileFormats.PDF,

        fromFormatLabel: «PDF»,

        toFormat: FileFormats.WORD,

        toFormatLabel: «Word»,

        description: ««

      })

    },

    {

      path: «/upload/pdf/jpg»,

      name: «pdf.jpg»,

      component: ConvertFileView,

      props: () => ({

        fromFormat: FileFormats.PDF,

        fromFormatLabel: «PDF»,

        toFormat: FileFormats.JPG,

        toFormatLabel: «JPG»,

        description: ««

      })

    },

    {

      path: «/upload/pdf/png»,

      name: «pdf.png»,

      component: ConvertFileView,

      props: () => ({

        fromFormat: FileFormats.PDF,

        fromFormatLabel: «PDF»,

        toFormat: FileFormats.PNG,

        toFormatLabel: «PNG»,

        description: ««

      })

    },

    {

      path: «/upload/pdf/jpeg»,

      name: «pdf.jpeg»,

      component: ConvertFileView,

      props: () => ({

        fromFormat: FileFormats.PDF,

        fromFormatLabel: «PDF»,

        toFormat: FileFormats.JPEG,

        toFormatLabel: «JPEG»,

        description: ««

      })

    },

    {

      path: «/upload/pdf/pptx»,

      name: «pdf.pptx»,

      component: ConvertFileView,

      props: () => ({

        fromFormat: FileFormats.PDF,

        fromFormatLabel: «PDF»,

        toFormat: FileFormats.POWER\_POINT,

        toFormatLabel: «Power Point»,

        description: ««

      })

    },

    {

      path: «/upload/json/csv»,

      name: «json.csv»,

      component: ConvertFileView,

      props: () => ({

        fromFormat: FileFormats.JSON,

        fromFormatLabel: «JSON»,

        toFormat: FileFormats.CSV,

        toFormatLabel: «CSV»,

        description: ««

      })

    },

    {

      path: «/upload/csv/json»,

      name: «csv.json»,

      component: ConvertFileView,

      props: () => ({

        fromFormat: FileFormats.CSV,

        fromFormatLabel: «CSV»,

        toFormat: FileFormats.JSON,

        toFormatLabel: «JSON»,

        description: ««

      })

    }.

AboutView.vue

<template>

  <div class=«about»>

    <h1>This is an about page</h1>

  </div>

</template>

<style>

@media (min-width: 1024px) {

  .about {

    min-height: 100vh;

    display: flex;

    align-items: center;

  }

}

</style>

**ConvertFileView.vue**

<script lang=«ts» setup>

import UploadFile from «../components/UploadFile.vue»;

const props = defineProps<{ // TODO make usage of own interfaces

  fromFormat: string

  fromFormatLabel: string

  toFormat: string

  toFormatLabel: string

  description: string

}>()

</script>

<template>

  <div>

    <div class=«text-box»>

      <h1 class=««>

        Convert files from {{ fromFormatLabel }} to {{ toFormatLabel }} formats

      </h1>

      <p>{{ description }}</p>

    </div>

    <div class=«upload-box»>

      <UploadFile :from-format=«fromFormat» :to-format=«toFormat»/>

    </div>

  </div>

</template>

<style lang=«css» scoped>

.text-box {

  text-align: center;

  font-family: -apple-system, BlinkMacSystemFont, «Segoe UI», Roboto, Helvetica, Arial, sans-serif, «Apple Color Emoji», «Segoe UI Emoji», «Segoe UI Symbol»;

}

.text-box h1 {

  font-weight: bold;

}

.upload-box {

  max-width: 50rem;

  margin: 0 auto;

}

</style>

**run.py**

import uvicorn

from typing import List

from fastapi import FastAPI, File, UploadFile

from fastapi.responses import FileResponse, StreamingResponse

from fastapi.requests import Request

from fastapi.middleware.cors import CORSMiddleware

import io

from backend.apps.converter.services.converter\_service import ConverterService

app = FastAPI()

app.add\_middleware(

    CORSMiddleware,

    allow\_origins=[«\*»],

    allow\_credentials=True,

    allow\_methods=[«\*»],

    allow\_headers=[«\*»],

    expose\_headers=[«\*»]

)

@app.post(«/upload/»)

async def resolve\_convert\_files(request: Request, files: List[UploadFile] = File(...)):

    zipped\_files: io.BytesIO = await ConverterService(files, request.headers[‘x-user-uuid’], request.headers[‘x-from-format’], request.headers[‘x-to-format’]).convert()

    return StreamingResponse(iter([zipped\_files.getvalue()]),

                             media\_type=«application/x-zip-compressed»,

                             headers={«Content-Disposition»: «attachment; filename=upload.zip»})

if \_\_name\_\_ == «\_\_main\_\_»:

    uvicorn.run(app, host=«127.0.0.1», port=8000, log\_level=«debug»)