

Інститут фізико-технічних та комп'ютерних наук  
Відділ комп'ютерних технологій  
Кафедра програмного забезпечення комп'ютерних систем

## КУРСОВИЙ ПРОЕКТ

на тему:

# ІНФОРМАЦІЙНА СИСТЕМА ПІДПРИЄМСТВА

Студента 3 курсу 343ск. групи  
напряму підготовки  
«Програмна інженерія»  
Шпильки А. А.

Керівник  
Доцент кафедри ПЗКС,  
Прохоров Г.В.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Нормоконтролер  
\_\_\_\_\_ В.В. Комісарчук  
” ” \_\_\_\_\_ 2018 р.

Члени комісії _____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

Чернівецький національний університет імені Юрія Федьковича  
Інститут фізико-технічних та комп'ютерних наук  
Кафедра Програмного забезпечення комп'ютерних систем  
Напрямок підготовки Програмна інженерія

**ЗАТВЕРДЖУЮ**

зав. кафедрою \_\_\_\_\_

«\_\_» \_\_\_\_\_ 2018 р.

**ЗАВДАННЯ**  
**на курсовий проект студента**  
Шпильки Анатолія Анатолійовича  
(прізвище, ім'я, по-батькові)

1. Тема проекту «Створення інформаційної системи підприємства»  
Затверджено наказом по університету від «\_\_» \_\_\_\_\_ 2018р. № \_\_\_\_\_
2. Термін подачі студентом завершеного проекту «\_\_» \_\_\_\_\_ червня 2018 р.
3. Вихідні дані до проекту:  
Створити програму для підприємства, яка містить базу даних різного роду виробів (вантажних, легкових автомобілів, автобусів, мотоциклів і т.д.) та робітників (інженерів, технологів, збирачів, токарів й т.д.). Кожна з перерахованих категорій виробів має характеристики, властиві тільки цієї або декількох категорій.  
Створити html сторінку з виводом даних та можливістю їх редагувати (видаляти, додавати нові дані).
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити):
  - розробити загальні вимоги до програми \_\_\_\_\_
  - описати методи \_\_\_\_\_
  - описати алгоритми програми \_\_\_\_\_
  - описати користувацький інтерфейс \_\_\_\_\_
  - розробити програму «Інформаційна система підприємства» \_\_\_\_\_
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
  - скріншоти роботи з програмою. \_\_\_\_\_

6. Дата видачі завдання “\_\_\_” \_\_\_\_\_ 2018 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної роботи	Термін виконання етапів роботи
1	Одержання технічного завдання	21/02/2018
2	Опрацювання літературних джерел	28/02/2018
3	Проектування структури	07/03/2018
4	Написання програмного коду	14/03/2018
5	Розробка баз даних	04/04/2018
6	Тестування та відлагодження програми	25/04/2018
7	Оформлення програмної документації	09/05/2018
8	Представлення готової роботи	25/05/2018
9	Захист роботи	Згідно графіку

Завдання прийняла до виконання

\_\_\_\_\_

(підпис студента)

Керівник проекту

\_\_\_\_\_

(підпис)

## РЕФЕРАТ

В курсовому проекті розроблено програмний проект інформаційної системи підприємства.

Програма розрахована для перегляду БД об'єктів системи та внесення певних змін у неї.

Область застосування – програма призначена для підприємства, щоб мати можливість переглядати БД створюваних об'єктів та звітів по виконаній роботі організації.

Розробка реалізована засобами середовища PhpStorm на мові PHP. Дане середовище є зручним у використанні для швидкого та якісного створення додатків.

Дана розробка у майбутньому може бути розширена із додаванням нового функціоналу і видозміненою логікою обробки.

Курсовий проект містить: \_\_26\_\_ ст., \_\_1\_\_ табл., \_\_3\_\_ джерела.

*ПІДПРИЄМСТВО, MODEL, ORM, REST, БАЗА ДАНИХ, PHP.*

## SUMMARY

## ЗМІСТ

СПЕЦИФІКАЦІЯ ПРОЕКТУ.....	7
1. ТЕХНІЧНЕ ЗАВДАННЯ.....	8
1.1. Підстави для розробки.....	8
1.2. Призначення розробки.....	8
1.3. Аналіз вимог до програмного забезпечення.....	8
1.3.1. Функціональні вимоги.....	8
1.3.2. Вимоги до складу та параметрів технічних засобів.....	9
1.3.2.1. Вимоги до інтерфейсу.....	9
1.3.2.2. Вимоги до інформаційної та програмної сумісності.....	11
1.3.2.3. Вимоги до тестування програмного забезпечення.....	11
1.4. Стадії та етапи розробки.....	13
1.5. Порядок контролю і приймання.....	13
2. АРХІТЕКТУРА ТА ФУНКЦІОНАЛЬНІ ПОКАЗНИКИ.....	15
2.1. Призначення та область застосування.....	15
2.2. Опис та обґрунтування обраної архітектури.....	15
3. ОПИС ПРОГРАМИ.....	16
3.1. Опис та обґрунтування обраних програмних засобів.....	16
3.2. Опис логічної структури.....	18
3.3. Опис модулів та класів.....	18
3.4. Опис і обґрунтування вхідних і вихідних даних.....	20
3.5. Опис інтерфейсу користувача.....	20
3.6. Використані технічні засоби.....	22
4. ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ.....	23
4.1. Об'єкт випробувань.....	23
4.2. Порядок та методика випробувань.....	23
4.3. Результати випробувань та використані технічні засоби.....	23
ВИСНОВКИ.....	25
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	26

## СПЕЦИЦІКАЦІЯ ПРОЕКТУ

Призначенням даного проекту являється створення інформаційної бази даних для підприємства. Програма призначена для перегляду розроблених виробів підприємством: вантажних і легкових автомобілів, автобусів, мотоциклів і т.д. та персоналу: інженерів, технологів, техніків, збираців, токарів і т. д. Кожна з перерахованих категорій виробів має характеристики, властиві тільки цієї або декільком категоріям: наприклад, до характеристик автобусів ставиться марка автобуса, кількість пасажирських місць, двигун, колір, коробка передач для вантажівок унікальними характеристиками є колесна база, тип самоскиду, вантажопідйомність. Для розробки даного проекту ми використовуємо БД.

Бази даних – це частина інформаційних систем – програмно-апаратних комплексів, що здійснюють зберігання та обробку величезних інформаційних масивів.

База даних являє собою певним чином структуровану сукупність даних, що зберігаються разом і обробляються відповідно до деякими правилами. Як правило, база даних моделює деяку предметну область або її фрагмент. Дуже часто в якості постійного сховища інформації баз даних виступають файли.

Інформація, яка зберігається в базі даних, може постійно поповнюватися. Від того, як часто це робиться, залежить її актуальність. Інформацію про об'єкти також можна змінювати і доповнювати.

Даний проект, що розробляється – інформаційна система (ІС) підприємства, що дозволяє автоматизувати роботу даної організації. Основна ціль підприємства – створювати вироби, які роблять замовники за допомогою договорів, а також структурно розділити організацію на категорії.

## **1. ТЕХНІЧНЕ ЗАВДАННЯ**

### **1.1 Підстави для розробки**

Розробка програмного забезпечення виконується на підставі рішення засідання кафедри ПЗКС про затвердження тем курсових робіт.

*Повна назва розробки:* «ІС підприємства» для ПК на основі ОС Windows».

*Коротка назва:* ІСП (Інформаційна система підприємства), програма.

### **1.2 Призначення розробки**

Призначенням даної розробки є створення інформаційної бази підприємства.

Реалізована версія програми призначена для робітників організації, щоб вони мали можливість переглядати структуру організації розподілену по таблицях.

### **1.3 Аналіз вимог до програмного забезпечення**

#### **1.3.1 Функціональні вимоги**

До програмного забезпечення висуваються наступні вимоги:

- Створення бази даних.
- Редагування таблиць бази даних.
- Перегляд інформації по таблицях.
- Забезпечити пошук згідно критеріїв.
- Створення інтерфейсу.

#### **1.3.2 Вимоги до складу та параметрів технічних засобів**



Для нормальної роботи розробленого програмного продукту необхідний персональний комп'ютер з наступними мінімальними характеристиками:

Параметр	Значення
Процесор	Intel Core 2 Duo 1.8 Ghz
Пам'ять	1 GB RAM
Жорсткий диск	1 GB

### 1.3.2.1 Вимоги до інтерфейсу

Програма повинна мати зрозумілий, зручний та візуально приємний інтерфейс.

Програмне забезпечення повинно бути оформлене у вигляді html-сторінки, що відображається в браузері з приємно підібраною кольоровою гамою.

Додатково передбачено кнопки переходу між таблицями.

Доступ до програмних засобів має бути забезпечено за допомогою web-браузера Google Chrome.

Програма повинна складатися з:

- Системи вибору будь-якої таблиці.
- Системи запитів.
- Системи додавання та редагування нових полів.

Основними властивостями, яким повинні задовольняти інтерфейси, є такі:

1. Адаптованість означає, що інтерфейс повинен бути:
  - сумісним з потребами та можливостями користувача;
  - забезпечувати простоту переходу від виконання однієї функції до іншої;

- забезпечувати користувача на високому рівні вказівками стосовно його можливих дій, а також генерувати належний зворотний зв'язок на його запити;
- надавати користувачу можливість відчувати себе повноправним керівником ситуації при розв'язанні всіх типів задач, тобто, забезпечувати його всією необхідною інформацією;
- користувач повинен бути впевненим, що він сам розв'язує поставлену задачу;
- забезпечувати користувача різними, взаємно доповнюючими формами представлення результатів в залежності від типу запиту або від характеру отриманого рішення;
- враховувати особливості користувачів різних рівнів.

2. Достатність інтерфейса означає таке:

- допустимі запити користувача повинні бути чіткими і однозначними для користувачів всіх рівнів, а також для прикладних задач всіх типів;
- реакція системи на всі типи запитів також повинна бути однозначною і зрозумілою і, по можливості, простою.

3. Дружність інтерфейсу – це максимальна простота його використання і готовність в повній мірі задовольнити запити користувача при розв'язанні визначеного класу задач.

4. Гнучкість інтерфейсу. Інтерфейс повинен бути максимально простим навіть у випадку, коли розв'язується дуже складна задача. При цьому простота означає таке:

- інтерфейс не повинен бути перевантажений деталями щодо представлення розв'язку поставленої задачі;
- користувач може не охопити всіх подробиць;
- тобто нічого зайвого, крім того, що необхідно для розуміння результату;

- він не повинен містити зайвих декоративних деталей, які відволікають від головної задачі;
- інтерфейс повинен бути консистентним, тобто, ґрунтуватись на використанні відомих, загальноприйнятих методів і засобів представлення інформації;
- в ідеалі процес взаємодії користувача з системою не повинен представляти ніяких труднощів.

#### **1.3.2.2 Вимоги до інформаційної та програмної сумісності**

Необхідною вимогою до програмного забезпечення є встановлена операційна система класу Windows 7/8/10, Linux, MacOS.

Налаштування таблиць зберігається у таблиці БД.

Окремих вимог не висувається.

#### **1.3.2.3 Вимоги до тестування програмного забезпечення**

Для тестування програмного забезпечення необхідно виконати наступні дії:

##### **1) Дії CRUD:**

- запустити програму;
- додати новий виріб;
- заповнити всі необхідні поля;
- редагування полів таблиці;
- видалення полів;
- закрити програму.

##### **2) Реалізація специфічних запитів:**

1. Одержати перелік видів виробів вказаної категорії та загалом, які були виготовлені вказаним цехом або підприємством.
2. Одержати кількість та список виробів окремої категорії та

загалом, які виготовлені вказаним цехом або підприємством протягом вказаного періоду.

3. Одержати дані про кадровий склад цеху або підприємства загалом та окремо за категоріями інженерно-технічного персоналу.
4. Одержати кількість та список ділянок вказаного цеху або підприємства загалом та їх керівників.
5. Одержати перелік робіт, які виконуються над вказаним виробом.
6. Одержати склад бригад вказаного цеху.
7. Одержати список майстрів вказаного цеху.
8. Одержати список виробів загалом та виробів вказаної категорії, що виготовляються на поточний момент вказаним цехом та підприємством загалом.
9. Одержати склад бригад, які приймають участь у виготовленні вказаного виробу.
10. Одержати список лабораторій, які приймають участь у випробуванні вказаного виробу.
11. Одержати перелік виробів загалом та вказаної категорії, які проходили випробування у вказаній лабораторії протягом вказаного періоду.
12. Одержати список працівників вказаної лабораторії, які брали участь у випробуванні вказаного виробу, виробів загалом та окремої категорії у вказаній лабораторії протягом вказаного періоду.
13. Одержати список устаткування, яке ви користувалося для випробування вказаного виробу, виробів загалом та окремої категорії у вказаній лабораторії протягом вказаного періоду.
14. Одержати список та перелік виробів загалом та вказаної категорії, які виготовлені вказаних цехом загалом та протягом вказаного періоду.

Після успішного виконання цих дій тестування програми може вважатися завершеним.

#### 1.4 Стадії та етапи розробки

№ з/п	Назва етапів випускної роботи	Термін виконання етапів роботи
1	Одержання технічного завдання	21/02/2018
2	Опрацювання літературних джерел	28/02/2018
3	Проектування структури	07/03/2018
4	Написання програмного коду	14/03/2018
5	Розробка баз даних	04/04/2018
6	Тестування та відлагодження програми	25/04/2018
7	Оформлення програмної документації	09/05/2018
8	Представлення готової роботи	25/05/2018
9	Захист роботи	Згідно графіку

#### 1.5 Порядок контролю і приймання

Програмне забезпечення повинне бути випробуване та протестоване на наявність помилок.

Порядок контролю і прийому проекту полягає в наступному:

- перевірка правильного функціонування програми керівником проекту;
- проект має бути розгорнений на зовнішньому сервері (посилання на нього видається усім викладачам);
- перевірка відповідності стандартам оформлення документації (перевірка нормоконтролером);
- перевірка правильного оформлення програмної документації керівником проекту;

- документація повинна бути вложена в проект в папку «/documentation»;
- попередній захист курсової роботи.

## 2. АРХІТЕКТУРА, ФУНКЦІОНАЛЬНІ ПОКАЗНИКИ

### 2.1 Призначення та область застосування

Призначення даної програми є створення інформаційної бази даних для підприємства.

### 2.2 Опис та обґрунтування обраної архітектури

Для даного проекту була вибрана об'єктно-зорієнтована архітектура, оскільки вона є найпоширенішою та найкраще підходить для розробки програм складного рівня. Середовищем для розробки було обрано JetBrains PhpStorm, а мовою програмування було обрано PHP, так як це середовище є зручним, а мова PHP однією з найпопулярніших для використання мов.

Було проаналізовано декілька систем керування базами даних, таких як: MySQL, MongoDB, PostgreSQL, SQLite. На мою думку, для цілей, які виконуватиме ця програма, більш за все відповідає вимогам та підходить система керування базами даних MySQL.

Архітектура складається з таких підрівнів:

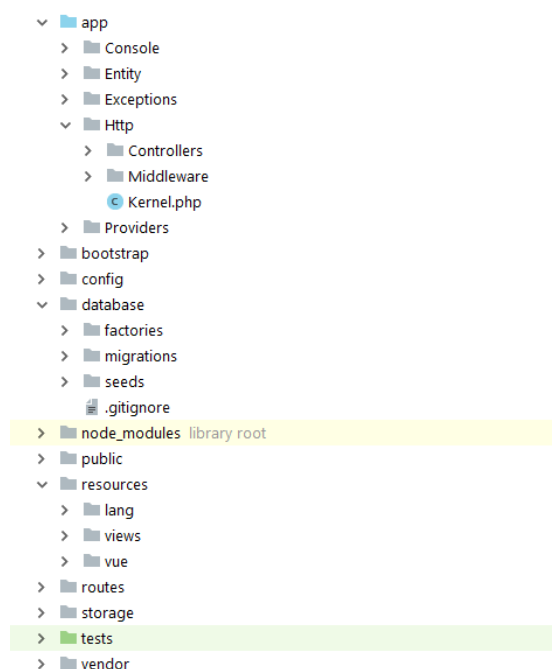


Рисунок 1 – Підрівні системи

### 3. ОПИС ПРОГРАМИ

#### 3.1 Опис і обґрунтування обраних програмних засобів

Розробка програмного забезпечення здійснена в середовищі JetBrains PhpStorm. Дане середовище розробки – зручний засіб розробки мови програмування PHP. JetBrains PhpStorm являє собою розширені рефакторингом і форматування для PHP, HTML, CSS, JavaScript, XML і багатьох інших мов. Присутні інструменти для роботи з базами даних і SQL файлами, включаючи зручний клієнт і редактор для схеми бази даних. В ньому поєднані потужні можливості мови програмування PHP які, в свою чергу, дозволяють розробнику легко, швидко та якісно створювати програмні продукти.

Я обрала мову програмування PHP тому, що:

1. Використання об'єктно-орієнтованого підходу дозволяє легко використовувати готові об'єкти в своїх програмах. Для запуску веб-додатку необхідний сервер з встановленими Apache/Nginx, MySQL та PHP.

2. Присутня динамічна типізація, C-подібний синтаксис та висока швидкодія.

3. Механізм автоматичного генерування документації на основі коментарів, розміщених в тексті програм.

4. Можливо реалізувати абсолютно різні за способом функціонування і сфері використання веб-додатки.

Framework Laravel:

1. Реалізує архітектурний шаблон MVC, який розділяє модель даних, інтерфейс користувача та модуль керування.

2. Забезпечує просту та ефективну роботу з БД, використовуючи Eloquent-ORM та міграції – систему керування версіями для БД, що спрощує розгортання і оновлення веб-додатку.



3. Забезпечує просту розробку інтерфейсу користувача за допомогою шаблонізатора Blade та інтеграції з Vue.js

4. Має вбудований інструмент для роботи в командному рядку Artisan який допомагає виконувати задачі які часто повторяються.

5. Має інтегровану систему модульного тестування phpunit, що робить розроблювані проекти більш надійними.

Базу даних MySQL, бо:

1. Може підтримувати роботу БД значних розмірів (берігають до 50 млн. записів).

2. Працює на різних платформах.

3. Має систему контролю доступу до даних, забезпечує шифрування даних при передаванні.

4. Швидкість функціонування.

ORM – Eloquent, бо:

1. Усуває безліч повторюваного коду.

2. Інтегрований в Laravel, можна за допомогою командного рядка створювати моделі та міграції.

3. Реалізує шаблон ActiveRecord для роботи з БД.

4. Підтримує кешування запитів до БД.

Front-end (Vue.js) бо:

1. Laravel офіційно підтримує і рекомендує використовувати Vue, він відразу інтегрований в проект.

2. Легший в використанні і супроводі ніж Angular і React, що зменшує час розробки та підтримки проекту.

3. Компоненти зберігаються в одному файлі, синтаксис шаблонів мінімальний, лаконічний та розширюваний.

4. Не потрібно використовувати TypeScript та JSX, розділення логіки та представлення, ні відміну від Angular і React.

5. Має хорошу документацію та розповсюджується по нормальній ліцензії, на відміну від React.

### 3.2 Опис логічної структури

Логічну структуру бази даних зручно подати у вигляді наступної діаграми варіантів використання (рис.2).

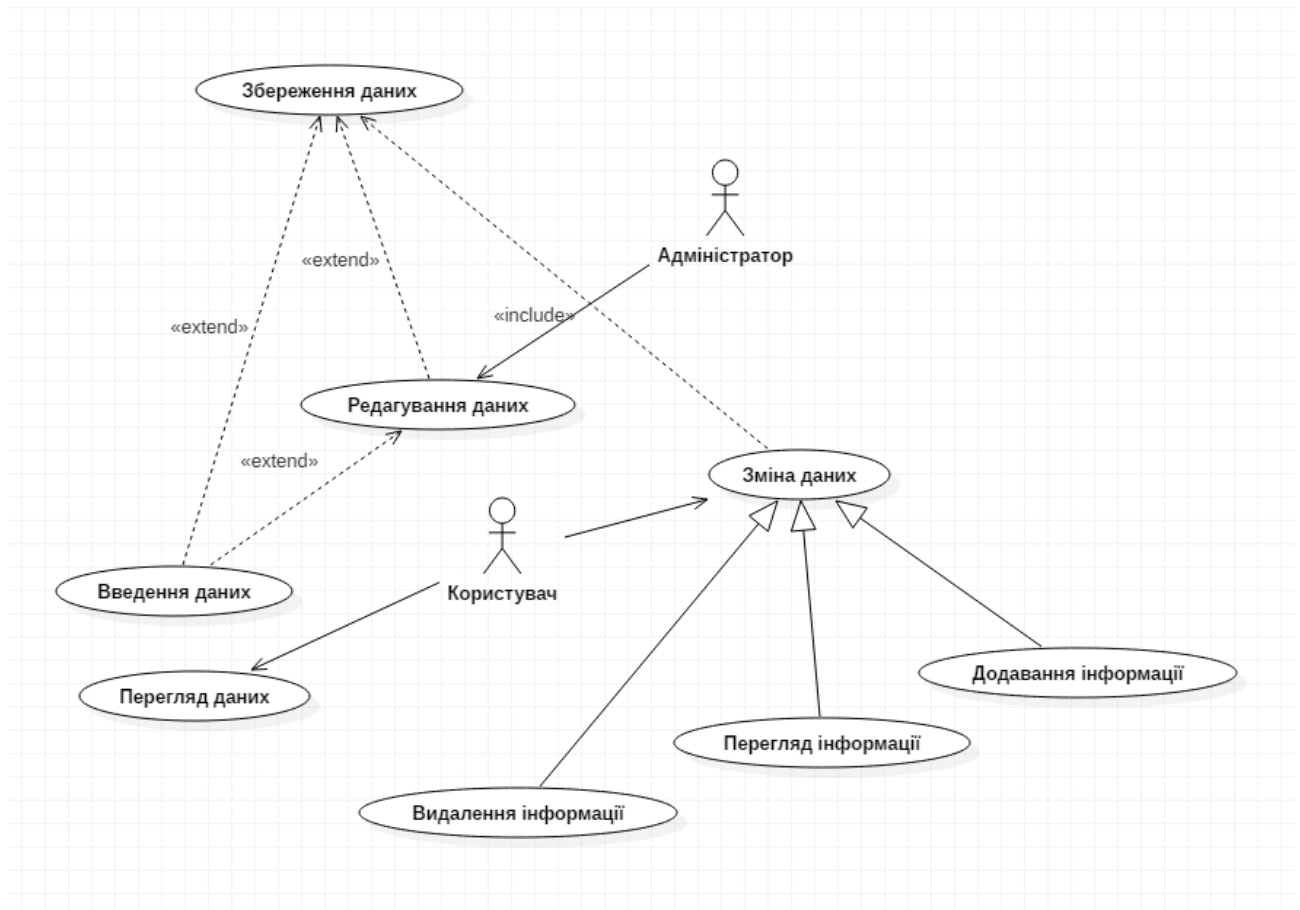


Рисунок 2 – Логічна структура бази

### 3.3 Опис модулів класів

Робота розробленого програмного забезпечення реалізується наступними файлами класами:

Клас Bridge – це клас для опису об’єкту бригади, що входить у склад підприємства:

```

class Brigade extends Model
{
    protected $fillable = [
        'brigade_name'
    ];

    public function workers()
    {
        return $this->hasMany(related: Worker::class);
    }
}

```

Клас Bus – це клас для опису об’єкту «автобус», що створює підприємство:

```

class Bus extends Model
{
    protected $fillable = [
        'brand',
        'engine',
        'color',
        'transmission',
        'production_year',
        'man_capacity',
        'make_now',
        'workshop_id',
    ];

    public function workshop()
    {
        return $this->belongsTo(related: Workshop::class);
    }

    public function testLabs()
    {
        return $this->belongsToMany(
            related: TestLab::class,
            table: 'buses_test_labs',
            foreignPivotKey: 'bus_id',
            relatedPivotKey: 'test_lab_id'
        );
    }

    public function setTestLabs($ids)
    {
        $this->testLabs()->sync($ids);
    }
}

```

Клас Car – це клас для опису об’єкту «автомобіль», що створює підприємство:

```

class Car extends Model
{
    protected $fillable = [
        'brand',
        'body_type',
        'engine',
        'color',
        'transmission',
        'production_year',
        'make_now',
        'workshop_id',
    ];

    public function workshop()
    {
        return $this->belongsTo(related: Workshop::class);
    }

    public function testLabs()
    {
        return $this->belongsToMany(
            related: TestLab::class,
            table: 'cars_test_labs',
            foreignPivotKey: 'car_id',
            relatedPivotKey: 'test_lab_id'
        );
    }

    public function setTestLabs($ids)
    {
        $this->testLabs()->sync($ids);
    }
}

```

Клас Engineer – це клас для опису об’єкту «інженери», що входять до складу підприємства:

```

class Engineer extends Model
{
    protected $fillable = [
        'first_name',
        'last_name',
        'middle_name',
        'phone_number',
        'personal_number',
        'home_address',
        'category',
        'position',
        'employment_date',
        'workshop_id'
    ];

    public function workshop()
    {
        return $this->belongsTo(related: Workshop::class);
    }
}

```

Клас Lorry – це клас для опису об'єкту «вантажівка», що створює підприємство:

```
class Lorry extends Model
{
    protected $fillable = [
        'brand',
        'engine',
        'color',
        'transmission',
        'production_year',
        'wheel',
        'dump_truck',
        'carrying',
        'make_now',
        'workshop_id',
    ];

    public function workshop()
    {
        return $this->belongsTo(related: Workshop::class);
    }

    public function testLabs()
    {
        return $this->belongsToMany(
            related: TestLab::class,
            table: 'lorries_test_labs',
            foreignPivotKey: 'lorry_id',
            relatedPivotKey: 'test_lab_id'
        );
    }
}
```

Клас Motorcycle – це клас для опису об'єкту «мотоцикл», що створює підприємство:

```
class Motorcycle extends Model
{
    protected $fillable = ['brand', 'engine', 'color', 'transmission', 'production_year',
        'stroller', 'make_now', 'workshop_id'];

    public function workshop()
    {
        return $this->belongsTo(related: Workshop::class);
    }

    public function testLabs()
    {
        return $this->belongsToMany(related: TestLab::class, table: 'motorcycles_test_labs',
            foreignPivotKey: 'motorcycle_id', relatedPivotKey: 'test_lab_id');
    }
}
```

Клас Worker – це клас для опису робітників, що працюють на підприємстві:

```
class Worker extends Model
{
    protected $fillable = [
        'first_name',
        'last_name',
        'middle_name',
        'phone_number',
        'personal_number',
        'home_address',
        'category',
        'is_brigadier',
        'employment_date',
        'brigade_id',
        'workshop_id'
    ];

    public function brigade()
    {
        return $this->belongsTo(related: Brigade::class);
    }

    public function workshop()
    {
        return $this->belongsTo(related: Workshop::class);
    }
}
```

Клас Workshop – це клас для опису об'єкту «цех».

```
class Workshop extends Model
{
    protected $fillable = ['workshop_name'];

    public function cars()
    {
        return $this->hasMany(related: Car::class);
    }

    public function engineers()
    {
        return $this->hasMany(related: Engineer::class);
    }

    public function workers()
    {
        return $this->hasMany(related: Worker::class);
    }

    public function testLabs()
    {
        return $this->belongsToMany(related: TestLab::class, table: 'workshop_test_labs',
            foreignPivotKey: 'workshop_id', relatedPivotKey: 'test_lab_id');
    }
}
```

### 3.4 Опис і обґрунтування вхідних і вихідних даних

Вхідними даними для розробленого продукту є таблиці в базі даних, такі як автомобілі, автобуси, мотоцикли, вантажівки, інженери, робітники, бригади, цехи. При відкритті кожної з таблиць можна виконувати різні маніпуляції над нею, а також повернутися назад за допомогою відповідної кнопки.

### 3.5 Опис інтерфейсу користувача

Робота з програмою вимагає лише базових знань та навичок роботи з web-браузером Google Chrome.

Розроблене програмне забезпечення – набір класів, які об'єднані з графічним інтерфейсом користувача, а також базою даних.

На рисунку 3 представлено головне вікно програми. Дана форма складається із головного меню та контентної області, на якій розміщуються вкладки, таблиця та елементи керування нею. При додаванні або редагуванні даних впливає модальне вікно з формою.

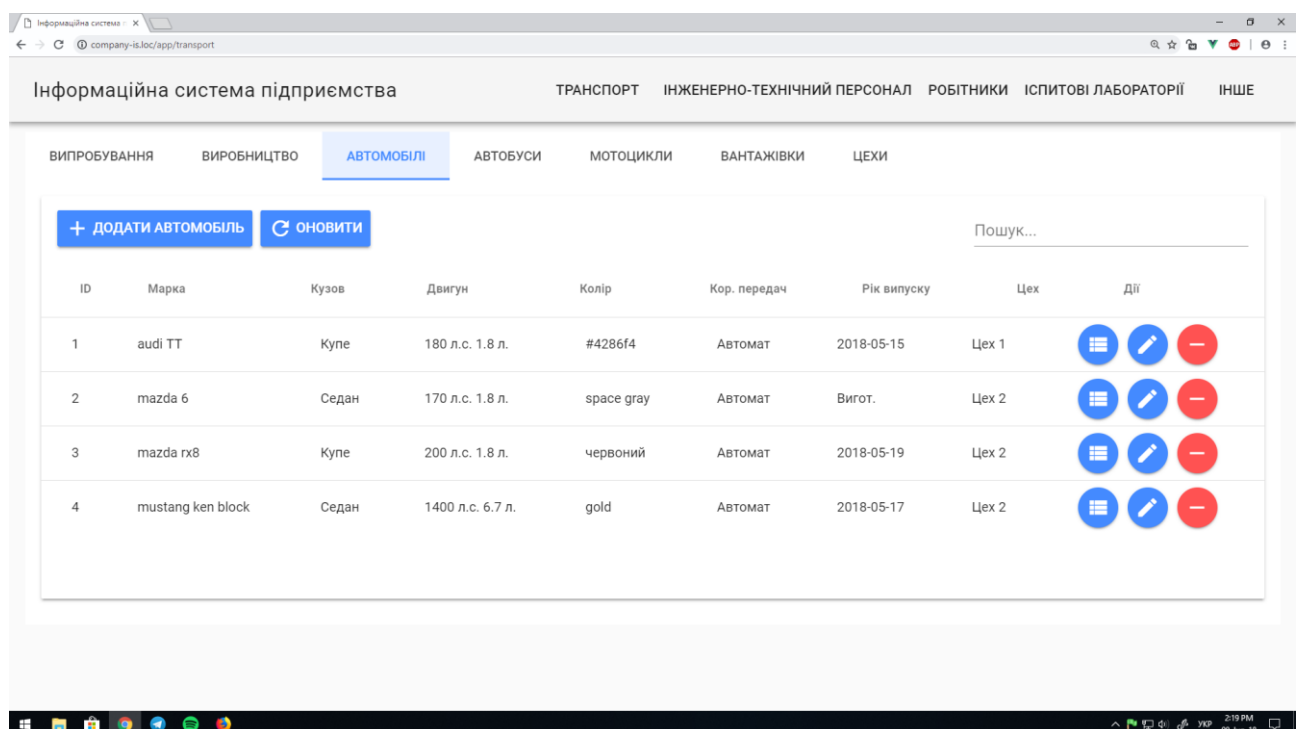


Рисунок 3 – Головне вікно програми

Тут ми бачимо п'ять основних елементів меню, а саме:

- Транспорт – кнопка для перегляду таблиць автомобілі, автобуси, мотоцикли, вантажівки, цехи та запитів до них.
- Інженерно-технічний персонал – кнопка для перегляду таблиці інженери та та запитів до неї.
- Робітники – кнопка для перегляду таблиці робітників та бригад та запитів до них.
- Іспитові лабораторії – кнопка для перегляду таблиці іспитових лібораторій.
- Інше – підменю з такими сторінками: використані технології, титульна сторінка та документація.

### **3.6 Використані технічні засоби**

Розробка програмного продукту здійснювалася на персональному комп'ютері з наступною конфігурацією:

- Процесор – Intel(R) Pentium G4560 CPU 3,5 GHz.
- ОЗП – 8 GB.
- Відеоадаптер – Nvidia Geforce 1050.
- Жорсткий диск – 1 TB.
- DVD-RW – PLDS DS8A8SH ATA Device.

Даний комп'ютер працює під управлінням ОС Windows 10 Pro.



## **4. ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ**

### **4.1 Об'єкт випробувань**

Об'єктом випробувань є створене в процесі виконання курсової роботи програмне забезпечення. Для перевірки коректності роботи програми було проведено ряд тестів. Мета випробувань – підтвердження задоволення усіх функціональних та не функціональних вимог до програмного забезпечення.

### **4.2 Порядок та методика випробувань**

Порядок проведення випробувань програмного забезпечення:

- запустити програму;
- додати новий запис;
- заповнити всі необхідні поля;
- редагування полів таблиці;
- видалення полів;
- перевірка реалізації потрібних запитів;
- закрити програму.

### **4.3 Результати випробувань та використані технічні засоби**

Випробування програмного продукту проводилось шляхом тестування різних введених функцій. При тестуванні помилок не виявлено. Програма працює правильно.

Тестування проводилось на пристроях з наступною конфігурацією:

Таблиця 1 – Характеристика пристроїв

Параметри	Пристрій №1	Пристрій №2
Процесор	Intel Pentium G4560 3,5 GHz.	Intel(R) Core™ i3-2365M
ОЗП	8 GB	4 GB
Версія ОС	Windows 10 Pro	Linux Mint 18.3
Пам'ять	1 TB	500 ГБ

Результати випробувань свідчать про те, що розроблене програмне забезпечення повністю задовольняє п.1.3.2.3 *Вимоги до тестування програмного забезпечення* технічного завдання.

## ВИСНОВОК

В даний час бази даних відіграють велику роль у веденні будь-якої документації. Майже кожна структура в наш час обладнана комп'ютером, який допомагає вести облік набагато зручніше ніж на папері. Автоматизація роботи з великими базами даних має велике значення в наш час.

Дана програма призначена для заповнення відповідних структур організації. На даному етапі програма містить в собі лише базові вимоги. Тобто, в базу заносяться лише дані про співробітників та виготовлені вироби. В подальшому планується розширення цієї програми.

Тестування показало працездатність програмного забезпечення та відповідність вимогам.

Дане програмне забезпечення має наступні можливості:

- додавання, редагування та видалення полів будь-якої із таблиць бази даних;
- перегляд результатів.

При розробці даної програми, використав на практиці наступні технології:

- проектування БД;
- моделювання БД;
- програмування мовою PHP;
- Laravel Framework;
- Eloquent ORM;
- REST API;
- Uml – діаграми;
- Git;
- MySQL;
- Material Design;
- Vue.js;
- HTML, CSS, JS.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Агальцов, В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. – 272 с.
2. Кириллов, В.В. Введение в реляционные базы данных. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб.: БХВ-Петербург, 2012. – 464 с.
3. Л. Веллингтон, Л. Томпсон «Разработка веб-приложений с помощью PHP и MySQL» 4-е изд. (2016)
4. Д. Складар «PHP 7. Руководство по созданию интерактивных веб-сайтов» (2017)
5. Флэнаган – JavaScript – Подробное руководство 6-ое издание
6. Фрейн – HTML5 CSS3 – разработка сайтов для любых браузеров и устройств