

ДОДАТОК А
Керівництво користувача

ЗМІСТ

| | |
|---|-----|
| Вступ..... | 81 |
| 1 Призначення та умови застосування..... | 82 |
| 2 Підготовка до роботи..... | 84 |
| 3 Опис операцій..... | 85 |
| 3.1 Авторизація та автентифікація..... | 85 |
| 3.2 Робочі області..... | 86 |
| 3.2.1 Робоча область менеджера з закупівель..... | 86 |
| 3.2.2 Робоча область менеджера з замовлень..... | 99 |
| 3.2.3 Робоча область планувальника виробництва..... | 103 |
| 4 Аварійні ситуації..... | 105 |
| 5 Рекомендації щодо освоєння..... | 106 |

ВСТУП

Документ Керівництво користувача допомагає ознайомитися з графічним інтерфейсом та функціональними можливостями розробленої системи. Також документ містить рекомендації по опануванню вебдодатку для підвищення ефективності роботи з ним.

Доступ до роботи з системою мають лише працівники підприємства через власні акаунти, створені адміністратором. Для працівників тих професій, інтерфейси яких передбачені в складі системи, робота з вебдодатком буде простою та інтуїтивно зрозумілою. Освоєння спеціальної технічної документації не вимагається.

Користувачам, які мають недостатній досвід роботи з персональним комп'ютером та браузерними додатками (робота зі сторінками, функціональними елементами вікон, кнопками, списками, таблицями, полями введення тощо), рекомендується ознайомитися з документом Посібник користувача в повній мірі.

1 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Розроблена система призначена для підтримки процесу обробки клієнтських замовлень та супроводжуючих його процесів.

Системою передбачено робочі області для трьох ролей користувачів: менеджер із закупівель, менеджер із замовлень, планувальник виробництва.

Менеджеру з закупівель доступний такий функціонал:

- моніторинг ресурсів компанії – набір функцій, який дозволяє відслідковувати запаси підприємства та маніпулювати записами про матеріали;

- робота з даними постачальників – набір функцій, що передбачає можливість додання нових постачальників до бази, внесення та редагування інформації у їхні картки;

- закупівля матеріалів – набір функцій, що дозволяє менеджеру створити заявку на закупівлю матеріалів у конкретного постачальника;

- перегляд статистики – набір функцій, що передбачає перегляд та маніпулювання даними про подані заявки на закупівлю матеріалів, а також аналіз витрат компанії за період.

Менеджеру з замовлень доступний такий функціонал:

- робота з даними про послуги – набір функцій, що дозволяє переглядати та редагувати наявні послуги з виготовлення продукції;

- обробка замовлень – набір функцій, що дозволяє приймати клієнтські замовлення та відправляти їх на виконання;

- перегляд статистики – набір функцій, що дозволяє переглядати та маніпулювати даними про подані на обробку та вже виконані замовлення, а також аналіз прибутку компанії за період.

Планувальнику виробництва доступний такий функціонал:

- обробка замовлень – набір функцій, що передбачає перегляд отриманих заявок від менеджера з замовлень, а також оновлення статусів замовлень.

– перегляд ресурсів компанії – функція, що дозволяє переглядати наявні матеріали, необхідні для виконання клієнтських замовлень.

2 ПІДГОТОВКА ДО РОБОТИ

Перед початком роботи необхідно переконатися, що виконані наступні умови:

а) користувач системи ознайомлений з правилами користування системою завдяки Посібнику користувача;

б) робочий комп'ютер підключений до мережі Інтернет;

в) здійснено з'єднання з сервером бази даних MongoDB Atlas;

г) відкритий сучасний веббраузер, що підтримує технології HTML5, CSS3 та JavaScript;

д) в пошуковому рядку прописана адреса вебсервісу <http://localhost:8080/> (адреса може змінитися, про що заздалегідь мають повідомити адміністратори) та натиснута клавіша Enter, після чого відбудеться автоматичне перенаправлення на сторінку /login та система буде готова до роботи.

3 ОПИС ОПЕРАЦІЙ

3.1 Авторизація та автентифікація

Якщо користувач попередньо зареєстрований в системі, він має отримати логін та пароль для доступу до своєї робочої області. Опинившись на сторінці логіну, користувач має внести ці дані до форми та натиснути кнопку Вхід. Сторінка входу зображена на рисунку А.1.



Рисунок А.1 – Сторінка входу

Якщо користувач ввів невірні дані, система повідомить йому про це. В такому випадку необхідно спробувати увійти знов та, у випадку невдачі, звернутися до адміністратора (рисунок А.2).

Вхід

User a@gmail.com not found

Email

Пароль

ВХІД

Рисунок А.2 – Невдала спроба входу

3.2 Робочі області

Після успішного входу в систему користувач, в залежності від його професії, потрапляє на головну сторінку своєї робочої області.

3.2.1 Робоча область менеджера з закупівель

Розглянемо робочу область менеджера з закупівель. Головна сторінка для цього працівника – склад матеріалів. Її вигляд представлено на рисунку А.3.

| Категорія | Матеріал | Кількість (шт) |
|----------------|-------------------------|----------------|
| Всі категорії | Введіть назву матеріалу | За зростанням |
| Метал | Лист | 100 |
| Дерево | Балка | 155 |
| Покриття | Лак епоксидний | 200 |
| Метал | Мідний дріт | 280 |
| Покриття | Фарба епоксидна | 440 |
| Гума | Гумовий ущільнювач | 540 |
| Оптика | Ліна | 720 |
| Ел. компоненти | Провідник | 980 |

Рисунок А.3 – Вигляд головної сторінки менеджера з закупівель

Якщо детально розібрати графічний інтерфейс менеджера з закупівель, то можна побачити два основних його елементи: панель навігації та блок контенту. Панель навігації необхідна для переміщення між сторінками в межах робочої області та виходу з системи. Вона завжди залишається статичною, лише підсвічуючи активні вкладки. При наведенні курсором на одну з вкладок, активується спадаючий список, розширюючий вибір доступних сторінок (рисунок А.4).

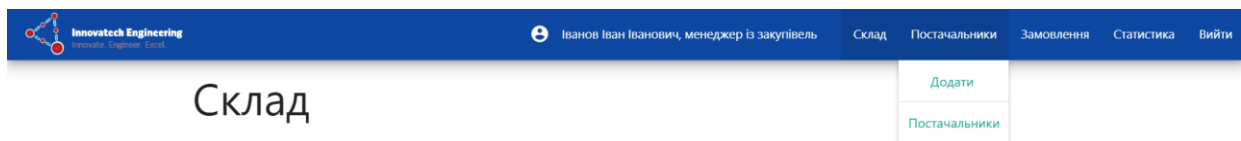


Рисунок А.4 – Активований спадаючий список


У блоці контенту й відбувається безпосереднє виконання робочих процесів. Наповнення блоку контенту змінюється в залежності від дій користувача та обраної вкладки.

На сторінці складу менеджер із закупівель може продивитися наявні матеріали та відслідкувати їх кількість. Передбачено можливість фільтрації за категорією, пошук за назвою та фільтрацію за кількістю за зростанням та спаданням (рисунок А.5)

| Категорія | Матеріал | Кількість (шт.) |
|-----------|-------------------------|-----------------|
| Гума | Введіть назву матеріалу | За зростанням |
| Гума | Гумовий ущільнювач | 540 |

Рисунок А.5 – Функції пошуку та фільтрації

Склад

| Категорія | Матеріал  | Кількість (шт.) |
|---------------|--|-----------------|
| Всі категорії | Дріт | За зростанням |
| Метал | Мідний дріт | 280 |

Склад


| Категорія | Матеріал  | Кількість (шт.) |
|-----------|--|-----------------|
| Покриття | Введіть назву матеріалу | За спаданням |
| Покриття | Фарба епоксидна | 440 |
| Покриття | Лак епоксидний | 200 |

Рисунок А.5, аркуш 2

Крім того, над даними складу можна проводити маніпуляції. Для цього на сторінці є елемент плаваючої бульбашки (рисунок А.6), який при наведенні мишею розкриє список можливих дій (рисунок А.7), які спочатку не будуть активними, оскільки користувач не знаходиться в режимі редагування.



Рисунок А.6 – Елемент Плаваюча бульбашка



Рисунок А.7 – Список можливих дій (неактивний)

Щоб перейти в режим редагування, необхідно натиснути на піктограму бульбашки, після чого всі рядки в таблиці стануть доступними для редагування (рисунок А.8). В них можна змінювати текст та потім натиснути на піктограму дискети для збереження змін.

Склад

| Категорія | Матеріал Q | Кількість (шт.) |
|----------------|-------------------------|-----------------|
| Всі категорії | Введіть назву матеріалу | За спаданням |
| Пластик | Пластикові гранули | 1270 |
| Ел. компоненти | Провідник | 980 |
| Оптика | Лінза | 720 |
| Гума | Гумовий ущільнювач | 540 |
| Покриття | Фарба епоксидна | 440 |










Рисунок А.8 – Увімкнений режим редагування

Якщо натиснути на піктограму плюс, створиться новий порожній рядок, який можна заповнити, а потім натиснути на піктограму дискети для збереження доданого рядку (рисунок А.9).

| Категорія | Назва матеріалу | Кількість |
|-----------|-----------------|-----------|
| | | |










Рисунок А.9 – Додавання нового рядку

Якщо натиснути на піктограму Корзина, відкриється режим видалення. Біля кожного рядку матеріалу з'явиться піктограма корзини (рисунок А.10), та, якщо натиснути на неї, відкриється модальне вікно з запитанням про підтвердження видалення рядку (рисунок А.11)



| | | | |
|--------|-------|-----|---|
| Дерево | Балка | 155 |  |
| Метал | Лист | 100 |  |

Рисунок А.10 – Увімкнений режим видалення

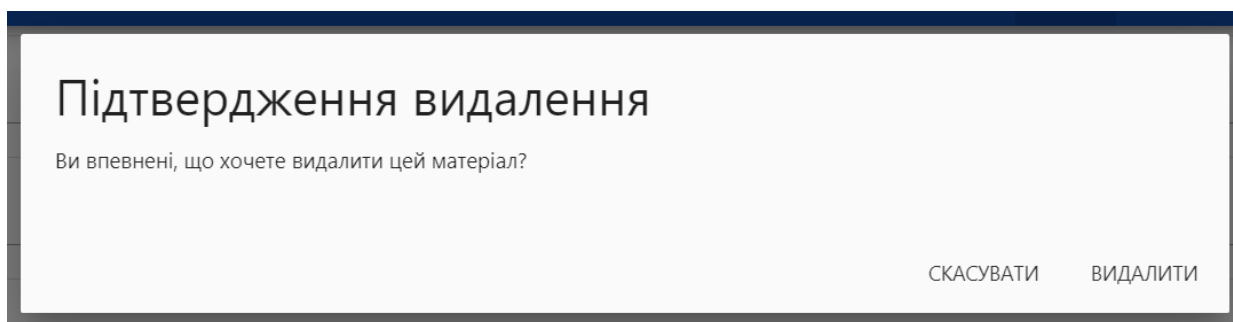


Рисунок А.11 – Вікно підтвердження видалення

Для того, щоб вийти з режиму редагування та скасувати всі незбережені зміни, повторно натисніть на плаваючу бульбашку.

Якщо переміститися на вкладку Додати зі спадаючого списку постачальників, відкриється сторінка додавання нового постачальника (рисунок А.12).

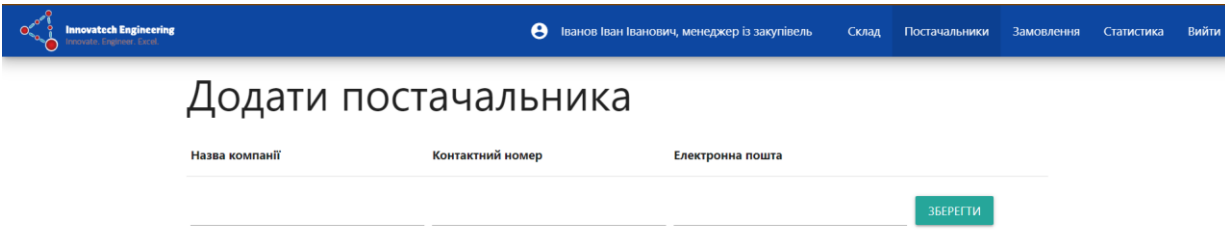


Рисунок А.12 – Сторінка додавання постачальника

Всі додані постачальники відображаються на вкладці Постачальники (рисунок А.13).

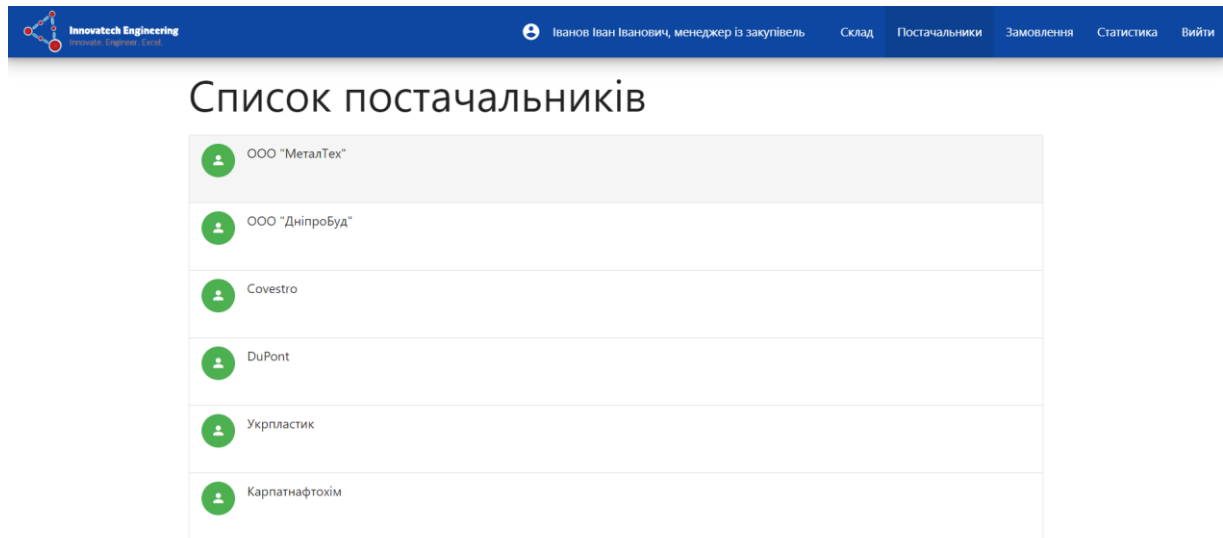


Рисунок А.13 – Сторінка списку постачальників

На цій сторінці можна через клік миші переміститися до сторінки з карткою постачальника (рисунок А.14).

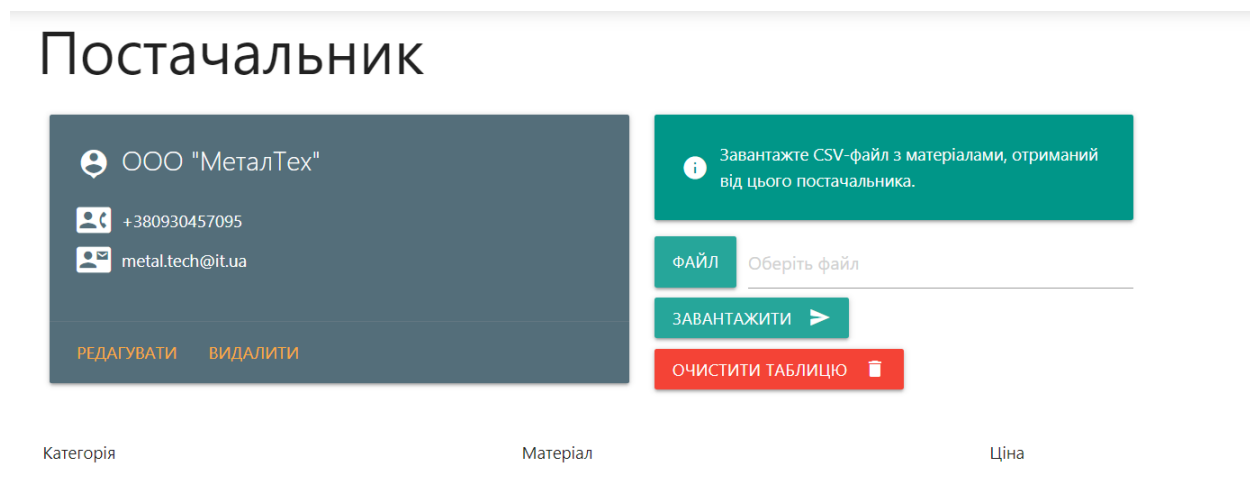
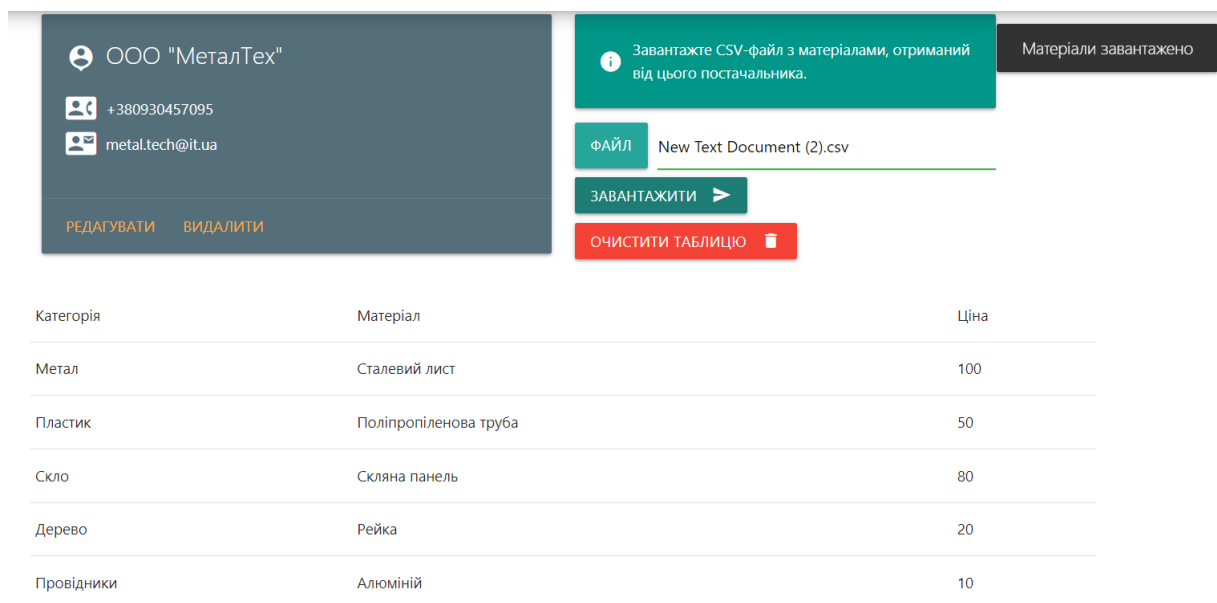


Рисунок А.14 – Сторінка з карткою постачальника

На даній сторінці можна переглянути дані постачальника, змінити їх та видалити постачальника з бази. Операція видалення супроводжується модальним вікном із підтвердженням. Також за допомогою кнопок ФАЙЛ та ЗАВАНТАЖИТИ є можливість завантажити список матеріалів у форматі .csv, отриманий від постачальника. Додані матеріали одразу відобразяться на сторінці (рисунок А.15). Якщо дані було завантажено некоректно, таблицю можна очистити за допомогою кнопки ОЧИСТИТИ ТАБЛИЦЮ.



Supplier Profile: ООО "МеталТех"
 Phone: +380930457095
 Email: metal.tech@it.ua
 Actions: РЕДАГУВАТИ, ВИДАЛИТИ

Message: Завантажте CSV-файл з матеріалами, отриманий від цього постачальника. Матеріали завантажено

File: New Text Document (2).csv
 Actions: ЗАВАНТАЖИТИ, ОЧИСТИТИ ТАБЛИЦЮ

| Категорія | Матеріал | Ціна |
|------------|-----------------------|------|
| Метал | Сталевий лист | 100 |
| Пластик | Поліпропіленова труба | 50 |
| Скло | Скляна панель | 80 |
| Дерево | Рейка | 20 |
| Провідники | Алюміній | 10 |



Рисунок А.15 – Процес завантаження списку матеріалів із .csv-файлу

Якщо переміститися на вкладку **Замовити** зі спадаючого списку замовлень, відкриється сторінка створення специфікації (рисунок А.16), яка дозволяє сформувати специфікацію з необхідних матеріалів конкретного постачальника, вказавши бажану дату доставки.

Нова специфікація

Бажана дата постачання
29.06.2024

Постачальник
ООО "Дніпробуд" ▼

| Категорія | Матеріал | Кількість | Вартість | |
|-----------|-----------------|-----------|-----------|---|
| Метал | ▼ Сталевий лист | ▼ 20 | 2000 грн. |  |
| Кераміка | ▼ Плитка | ▼ 30 | 1050 грн. |  |

Загальна кількість: 50 шт. Загальна вартість: 3050 грн.


 **СТВОРИТИ**

Рисунок А.16 – Сторінка створення специфікації

Як можна побачити з рисунку А.16, список матеріалів завантажується на підставі раніше заповненої картки постачальника. Вказавши категорію, матеріал та його кількість, вартість розраховується автоматично. Також автоматично розраховується сумарна кількість та вартість усіх матеріалів. Є функція додавання нових рядків та видалення вже наявних. Створення специфікації підтверджується натисканням кнопки **СТВОРИТИ**.

Створені специфікації відображаються на вкладці Специфікації спадаючого списку замовлень (рисунок А.17), у якому в кожному рядку специфікації вказана скорочена інформація про неї та її статус: проведена специфікація має зелену піктограму документу та галочку навпроти статусу, не проведена – червону піктограму та порожній чекбокс.







| | |
|---|--|
| ✓ Проведено | ✓ Виконано |
|  ООО "Woodland" 31.04.2024 | ✓ Проведено ✓ Виконано |
|  ООО "Woodland" 31.05.2024 | ✓ Проведено ✓ Виконано |
|  ООО "Woodland" 31.06.2024 | ✓ Проведено ✓ Виконано |
|  ООО "МеталТех" 31.05.2024 | ✓ Проведено <input type="checkbox"/> Не виконано |
|  ООО "МеталТех" 30.06.2024 | ✓ Проведено ✓ Виконано |
|  ООО "ДніпроБуд" 29.06.2024 | <input type="checkbox"/> Не проведено <input type="checkbox"/> Не виконано |

Рисунок А.17 – Список специфікацій


Якщо клікнути по специфікації, відкриється детальна інформація про неї (рисунок А.18).

Специфікація

Не проведено Не виконано

| Бажана дата доставки | Постачальник | Ел. пошта | Матеріал | Кількість (од.) | Вартість (UAH) |
|----------------------|-----------------|-------------------------------|-------------------------|-----------------|----------------|
| 29.06.2024 | ООО "ДніпроБуд" | olinkevich.yaroslav@gmail.com | Сталевий лист Плитка | 20 30 | 2000 1050 |

Сумарна вартість: 3050.00 UAH

 Коментар до замовлення





ОФОРМИТИ НА БЛАНКУ  ПРОВЕСТИ  ПРОВЕСТИ Й ВІДПРАВИТИ ЗАПИТ  ВИДАЛИТИ З БАЗИ ДАНИХ 

Рисунок А.18 – Сторінка детальної інформації про специфікацію

Як видно на рисунку А.18, заповнену раніше інформацію було занесено до таблиці. Однак ці дані поки ще не є сформованим замовленням, оскільки

специфікацію ще потрібно провести. Провести специфікацію можливо двома способами:

– вручну: натиснути на кнопку ПРОВЕСТИ, після чого статус специфікації зміниться на Проведено, в таблиці даних з'явиться колонка про дату проведення специфікації та деякі кнопки, пов'язані з проведенням, стануть неактивними. Після цього необхідно натиснути на кнопку ОФОРМИТИ НА БЛАНКУ, щоб отримати заявку в форматі PDF, та вручну надіслати її на пошту постачальника;

– автоматично: натиснути на кнопку ПРОВЕСТИ Й ВІДПРАВИТИ ЗВІТ, після чого система також відмітить специфікацію як проведено, додасть колонку з датою проведення до таблиці, зробить неактивними деякі кнопки, сформує заявку у форматі PDF та автоматично відправить його на пошту постачальника, яка вказана в таблиці.

Також є можливість написання коментаря, який додасться до листа з заявою у вигляді тексту. Також є можливість видалення специфікації з бази даних – ця дія супроводжується модальним вікном із попередженням про вплив на звітність.

Процес проведення специфікації зображено на рисунку А.19.

The screenshot shows a web interface for managing specifications. At the top, the title 'Специфікація' is displayed. A green notification box on the right states 'Специфікацію успішно відмічено як проведено'. Below the title is a table with columns: 'Заявку подано', 'Бажана дата доставки', 'Постачальник', 'Ел. пошта', 'Матеріал', 'Кількість (од.)', and 'Вартість (UAH)'. The table contains one row of data. Below the table, a grey box shows 'Сумарна вартість: 3050.00 UAH'. At the bottom, there are four buttons: 'ОФОРМИТИ НА БЛАНКУ' (with a PDF icon), 'ПРОВЕСТИ', 'ПРОВЕСТИ Й ВІДПРАВИТИ ЗАПИТ' (with an envelope icon), and 'ВИДАЛИТИ З БАЗИ ДАНИХ' (with a trash icon).

| Заявку подано | Бажана дата доставки | Постачальник | Ел. пошта | Матеріал | Кількість (од.) | Вартість (UAH) |
|---------------|----------------------|-----------------|-------------------------------|-------------------------|-----------------|----------------|
| 07.06.2024 | 29.06.2024 | ООО "Дніпробуд" | olinkevich.yaroslav@gmail.com | Сталевий лист Плитка | 20 30 | 2000 1050 |

Сумарна вартість: 3050.00 UAH

ОФОРМИТИ НА БЛАНКУ ПРОВЕСТИ ПРОВЕСТИ Й ВІДПРАВИТИ ЗАПИТ ВИДАЛИТИ З БАЗИ ДАНИХ

Рисунок А.19 – Процес проведення специфікації

Сформована заява представлена на рисунку А.20.

**Innovatech Engineering**вул. Набережна, 12
м. Київ, Україна**Замовлення: №6663232c2d076a634ffbc34a****Дата створення:** 07.06.2024**До сплати:** 71500 UAH**Дата поставки:** 29.06.2024**Постачальник:** ООО "ДніпроБуд"

ПП: Олінкевич Я.В.

Адреса: Україна, м. Київ, вул. Лісна 24

| Матеріал | Кількість (од.) | Вартість (UAH) |
|---------------|-----------------|----------------|
| Сталевий лист | 20 | 2000 |
| Плитка | 30 | 1050 |

Загальна сума 3050 UAH**Сплачено** 0 UAH**До сплати** 3050 UAH

Оплата здійснюється протягом 15 днів. Дякуємо за вашу співпрацю.

Рисунок А.20 – Заява на замовлення матеріалів

Якщо працівнику потупила інформація, що постачальник здійснив доставку замовлення, його можна відмітити як Виконане, натиснувши на порожній чекбокс (рисунок А.21).

| Специфікація | | | | | | | Замовлення успішно відмічене як виконане |
|--------------------|----------------------|-----------------|------------------------------|-----------------------------|-----------------|-------------------------------|--|
| Заявку подано | Бажана дата доставки | Постачальник | Ел. пошта | Матеріал | Кількість (од.) | Вартість (UAH) | |
| 07.06.2024 | 29.06.2024 | ООО "ДніпроБуд" | olinkевич.yaroslav@gmail.com | Сталевий лист | 20 | 2000 | |
| | | | | Плитка | 30 | 1050 | |
| | | | | | | Сумарна вартість: 3050.00 UAH | |
| ОФОРМИТИ НА БЛАНКУ | | ПРОВЕСТИ | | ПРОВЕСТИ Й ВІДПРАВИТИ ЗАПИТ | | ВИДАЛИТИ З БАЗИ ДАНИХ | |

Рисунок А.21 – Процес відмічання замовлення як виконаного

Якщо повернутися на сторінку списку специфікацій, можна побачити, що специфікація змінила свій вигляд, та, за бажанням, її можна видалити з цього списку для зручності роботи з іншими специфікаціями (дані не видаляються з бази даних) (рисунок А.21).

| | | |
|---|-----------------|------------|
| | ООО "ДніпроБуд" | |
| | 29.06.2024 | |
| ✓ | Проведено | ✓ Виконано |

Рисунок А.21 – Змінений вигляд виконаного замовлення

Для перегляду списку проведених специфікацій та пошуку за постачальником, датою подання заявки, бажаною датою доставки та статусом, необхідно перейти на вкладку Закупівлі зі спадаючого списку статистики (рисунок А.22).

Список заявок

| Постачальник <input type="text"/> | Дата подання заявки | Бажана дата доставки | Статус |
|-----------------------------------|---------------------|----------------------|-------------|
| Введіть постачальника | | Всі | Всі |
| ООО "Woodland" | 19.04.2024 | 31.05.2024 | Виконано |
| ООО "МеталТех" | 30.05.2024 | 25.06.2024 | Не виконано |
| ООО "МеталТех" | 01.03.2024 | 10.03.2024 | Виконано |
| ООО "Woodland" | 19.04.2024 | 31.04.2024 | Виконано |
| ООО "Woodland" | 19.05.2024 | 31.05.2024 | Виконано |
| ООО "Woodland" | 05.06.2024 | 31.06.2024 | Виконано |
| ООО "МеталТех" | 30.05.2024 | 31.05.2024 | Не виконано |
| ООО "МеталТех" | 03.06.2024 | 30.06.2024 | Виконано |

Рисунок А.22 – Список заявок на закупівлю матеріалів

Якщо клікнути по рядку, відкриється детальна інформація про специфікацію.

Щоб подивитися звітність із витрат компанії на закупівлю матеріалів у вигляді лінійної діаграми, необхідно переміститися на вкладку Звітність із того ж спадаючого списку (рисунок А.23).

Статистика закупівель за 2024 рік

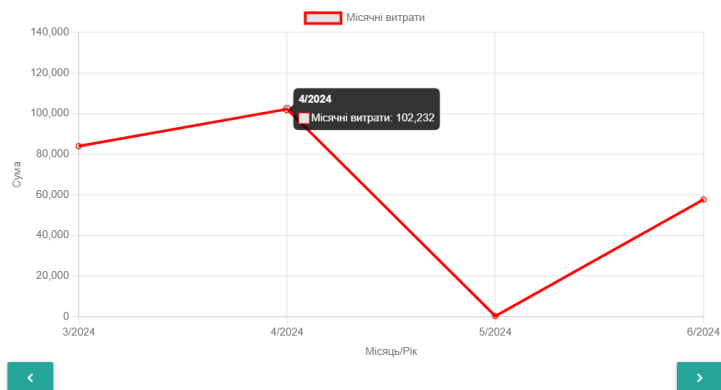


Рисунок А.23 – Графік статистики закупівель

Даний графік є інтерактивним. Можливим є перегляд витрат за кожен місяць, також можна перемикатися між роками.

3.2.2 Робоча область менеджера з замовлень

Увійшовши до системи як менеджер із замовлень, користувач потрапляє на головну сторінку – Список специфікацій (рисунок А.24).

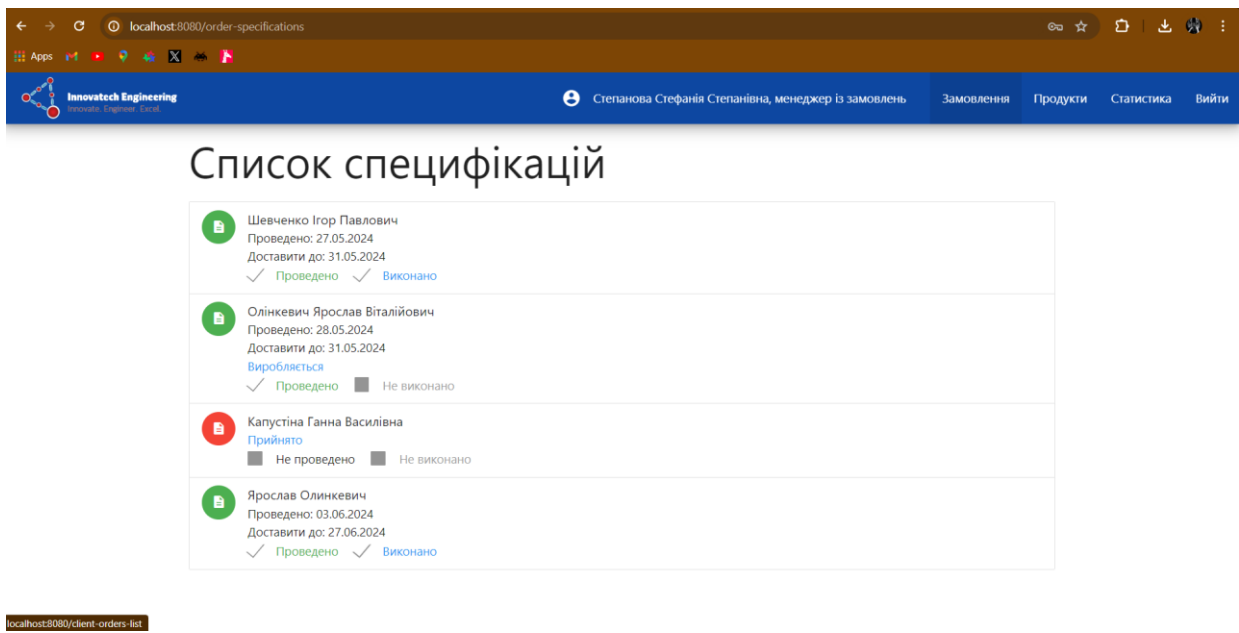


Рисунок А.24 – Головна сторінка менеджера з замовлень

На даній сторінці можна переглянути список специфікацій, які створюються менеджером із замовлень із короткою інформацією та статусами. Щоб створити специфікацію, необхідно переміститися на вкладку Створити спадаючого списку замовлень (рисунок А.25).

Нове замовлення

Бажана дата доставки

29.06.2024

| | | |
|-------------------|------------|-----------------------------|
| Замовник | Телефон | Email |
| Ярослав Олінкевич | 0290747095 | olinkevych.yaroslav@nure.ua |

| Категорія | Продукт | Кількість | Вартість |
|-----------|------------------|-----------|-------------|
| Дім | Телевізор МК-240 | 50 | 750000 грн. |

Загальна кількість: 50 шт.

Загальна вартість: 750000 грн.



Рисунок А.25 – Сторінка створення замовлення

На сторінці з рисунку А.25 можливо вказати дані, надані клієнтом при замовленні, а також обрати одну або декілька послуг із виготовлення продукції. Специфікація створюється шляхом натискання на кнопку СТВОРИТИ, після чого її можна побачити в списку, зображеному на рисунку А.24.

Натиснувши на потрібну специфікацію зі списку, відкриється сторінка з детальною інформацією про замовлення (рисунок А.25)

Замовлення

Номер в базі: 66632b6f2d076a634ffbc387 ■ Не проведено □ Не виконано

1 Прийнято
2 У плануванні
3 Виробляється
4 Контроль якості
5 Їде на склад
6 На складі

| Бажана дата доставки | Замовник | Ел. пошта | Продукція | Кількість (од.) | Вартість (UAH) |
|----------------------|-------------------|-----------------------------|------------------|-----------------|----------------|
| 29.06.2024 | Ярослав Олінкевич | olinkevych.yaroslav@nure.ua | Телевізор МК-240 | 50 | 750000 |

Сумарна вартість: 750000.00 UAH

ОФОРМИТИ НАКЛАДНУ
ПРОВЕСТИ
ПРОВЕСТИ Й ВІДПРАВИТИ НАКЛАДНУ
ВИДАЛИТИ З БАЗИ ДАНИХ

Рисунок А.25 – Сторінка детальної інформації про замовлення

Дану специфікацію можна також можна провести одним із двох способів, які було описано в п.п. 3.2.1, та або надіслати сформовану накладу вручну на пошту замовника або зробити це автоматично. Коли замовлення повністю виготовлене та доставлене клієнтові, його можна відмічати як виконане у відповідному чекбоксі. Прогрес виконання замовлення відображається у статусбарі. Нові статуси встановлює планувальник виробництва в залежності від поточного етапу виготовлення замовлення.

Список проведених специфікацій можна подивитися на вкладці Замовлення спадаючого списку замовлень (рисунок А.26), а також використати потрібні фільтри для зручного пошуку необхідного замовлення.

Список замовлень

| Замовник <input type="text"/> | Дата створення | Бажана дата доставки | Статус |
|-------------------------------|----------------|----------------------|-------------|
| Введіть замовника | Всі | Всі | Всі |
| Шевченко Ігор Павлович | 27.05.2024 | 31.05.2024 | Виконано |
| Олінкевич Ярослав Віталійович | 28.05.2024 | 31.05.2024 | Не виконано |
| Ярослав Олінкевич | 03.06.2024 | 27.06.2024 | Виконано |

Рисунок А.26 – Сторінка списку замовлень

За кліком по рядку відкриється сторінка детальної інформації про замовлення.

Якщо переміститися на вкладку Продукти, то можна отримати список продукції, послуги з виготовлення якої надає компанія, та здійснити операції з маніпуляції даним списком, які було описано в п.п. 3.2.1 (рисунок А.27).

Продукція

| Категорія | Продукт | Ціна (грн.) |
|---------------|-------------------------|--------------|
| Всі категорії | Введіть назву продукту | За спаданням |
| Дім | Телевізор МК-240 | 15000 |
| Кухня | Тостер АЗ-106-8 | 2500 |
| Кухня | Комбайн кухонний АЗУ-18 | 6400 |
| Дім | Пилосос БР-145 | 7000 |
| Сад | Газонокосарка ІК-209-45 | 9800 |



Рисунок А.27 – Сторінка списку продукції

Якщо переміститися на вкладку Статистика, то можна побачити лінійну діаграму, яка відображає прибуток на основі виконаних замовлень. Цей графік інтерактивний, можна подивитися детальну інформацію за кожен місяць, а також перемикатися між роками. Сторінку наведено на рисунку А.28.

Статистика продажів за 2024 рік

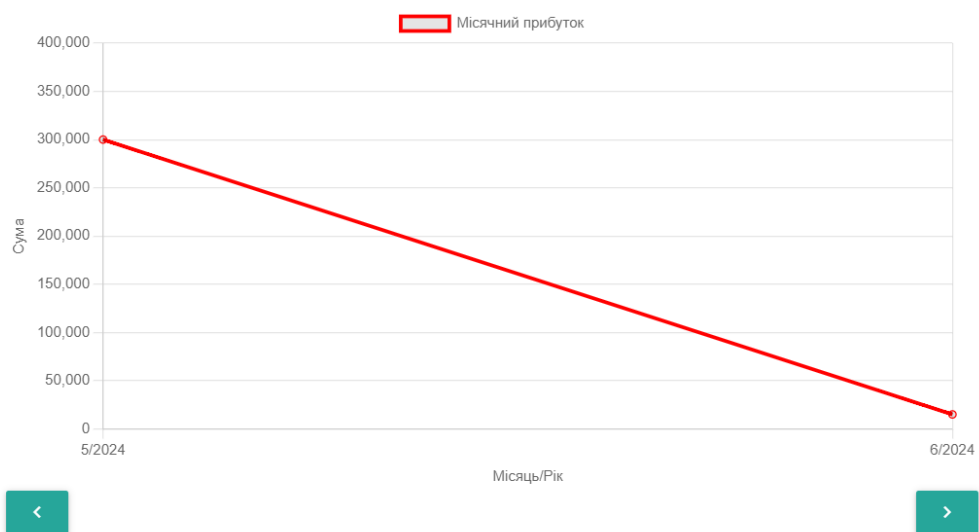


Рисунок А.28 – Сторінка статистики з продажів

3.2.3 Робоча область планувальника виробництва

Увійшовши до системи як планувальник виробництва, користувач потрапляє на головну сторінку – Список замовлень (рисунок А.29), яка відображає специфікації, проведені менеджером із замовлень.

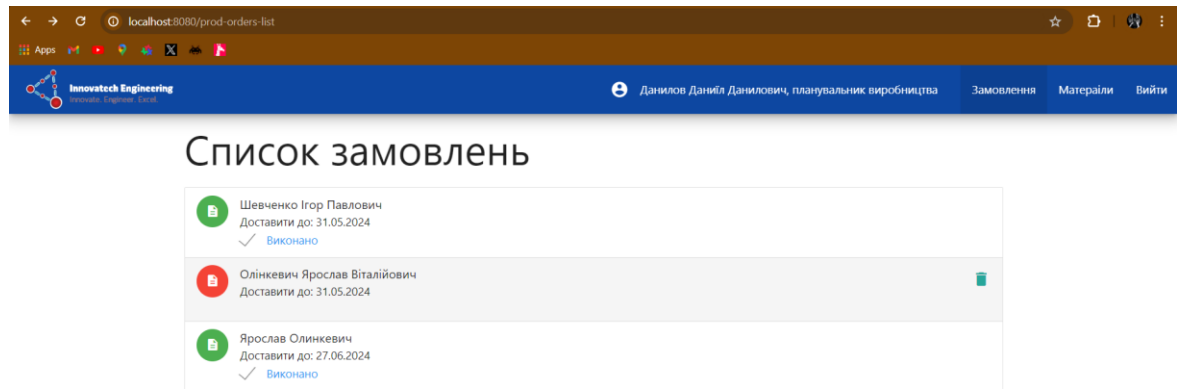


Рисунок А.29 – Сторінка зі списком замовлень

За кліком по замовленню відкриється детальна інформація про нього (рисунок А.30).

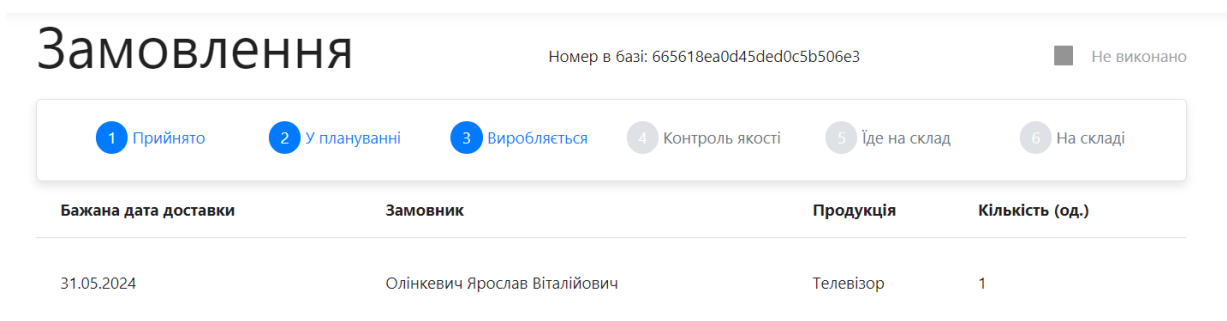


Рисунок А.30 – Сторінка деталей замовлення

Як видно з рисунку А.30, на сторінці відображається необхідна для планувальника інформація про замовлення, а також присутній інтерактивний блок зі статусами замовлення. Статус виконання замовлення змінюється за допомогою кліку на відповідну бульбашку етапу.

Якщо переміститися на вкладку Матеріали, то можна побачити список доступних ресурсів, виділених компанією для виконання клієнтських замовлень, у якому можна зайнятися пошуком матеріалів за допомогою відповідних фільтрів та полів (рисунок А.31).

Склад


| Категорія | Матеріал  | Кількість (шт.) |
|----------------|--|-----------------|
| Всі категорії | Введіть назву матеріалу | За зростанням |
| Метал | Лист | 100 |
| Дерево | Балка | 155 |
| Покриття | Лак епоксидний | 200 |
| Метал | Мідний дріт | 280 |
| Покриття | Фарба епоксидна | 440 |
| Гума | Гумовий ущільнювач | 540 |
| Оптика | Лінза | 720 |
| Ел. компоненти | Провідник | 980 |

Рисунок А.31 – Сторінка доступних матеріалів

4 АВАРІЙНІ СИТУАЦІЇ

Перелік можливих помилок, з якими може зіткнутися користувач при роботі з системою, наведено в таблиці 4.1.

Таблиця 4.1 – Можливі помилки при роботі з системою

| № | Опис помилки | Спосіб усунення |
|----|-------------------------|---|
| 1. | Сервер не відповідає | Перевірити, чи запущений сервер; перевірити з'єднання з інтернетом; звернутися до адміністратора. |
| 2. | Помилка занесення даних | Перевірити занесені дані на відповідність формату, що вимагається системою; звернутися до адміністратора. |

Якщо під час роботи в системі було втрачено інтернет-підключення, то браузер виведе вікно, що повідомляє про помилку. Система буде повторювати спроби відновити свою роботу, доки не встановиться стабільне з'єднання з Інтернетом.

5 РЕКОМЕНДАЦІЇ ЩОДО ОСВОЄННЯ

Для ефективної взаємодії користувача з системою необхідно мати базові навички роботи з персональним комп'ютером та браузерними додатками, а також ознайомитися з Посібником користувача.

ДОДАТОК Б
Текст програми

Index.js:

```

const express = require('express')
const mongoose = require('mongoose')
const path = require('path')
const exphbs = require('express-handlebars')
const config = require('./config');
const session = require('express-session')
const passport = require('passport')
const LocalStrategy = require('passport-local').Strategy

const materialRoutes = require('./routes/materials')
const providerRoutes = require('./routes/providers')
const specificationRoutes = require('./routes/specifications')
const reportRoutes = require('./routes/reports')
const salesReportRoutes = require('./routes/sales-reports')
const orderRoutes = require('./routes/orders')
const prodOrderRoutes = require('./routes/prod-orders')
const productRoutes = require('./routes/products')
const authenticationRoutes = require('./routes/authentication')
const auth = require('./middleware/auth')

const { formatDate } = require('./service/formatDate')
const User = require('./models/User')

const PORT = process.env.PORT || 8080
const app = express()

const hbs = exphbs.create({
  defaultLayout: 'main',
  extname: 'hbs',
  helpers: {
    formatDate,
    eq: function (a, b) {
      return a === b;
    }
  }
})

app.engine('hbs', hbs.engine)
app.set('view engine', 'hbs')
app.set('views', 'views')

app.use(express.urlencoded({ extended: true }))
app.use(express.static(path.join(__dirname, 'public')))

app.use(session({
  secret: config.session.secret,
  resave: false,
  saveUninitialized: true

```

```

app.use(passport.initialize())
app.use(passport.session())

passport.use(new LocalStrategy(
  async (username, password, done) => {
    try {
      const user = await User.findOne({ username })
      if (!user) {
        return done(null, false, { message: `User ${username} not found` })
      }
      if (user.password !== password) {
        return done(null, false, { message: 'Incorrect password' })
      }
      return done(null, user)
    } catch (err) {
      return done(err)
    }
  }
))

passport.serializeUser((user, done) => {
  done(null, user.id)
})

passport.deserializeUser(async (id, done) => {
  try {
    const user = await User.findById(id)
    done(null, user)
  } catch (err) {
    done(err)
  }
})

app.use((req, res, next) => {
  res.locals.isAuthenticated = req.isAuthenticated();
  next();
});

app.use(authenticationRoutes, materialRoutes, providerRoutes, specificationRoutes,
reportRoutes, orderRoutes, productRoutes, prodOrderRoutes, salesReportRoutes)

async function start() {
  try {
    await mongoose.connect(config.mongodb.uri);
    app.listen(PORT, () => {
      console.log('Server has been started...');
    });
  } catch (e) { console.log(e);} start()
}

```

Config.js:

```
module.exports = {
  mongodb: {
    uri:
      'mongodb+srv://Thomas_Howard:Yarik31082002@cluster0.ibqjpdg.mongodb.net/todos'
  },
  session: {
    secret: 'your-secret-key'
  }
}
```

AuthController.js:

```
const User = require('../models/User');

const passport = require('passport');

class authController {
  async login(req, res, next) {
    passport.authenticate('local', (err, user, info) => {
      if (err || !user) {
        return res.render('login', { layout: 'login-layout', error: info ?
info.message : 'Login failed' });
      }
      req.logIn(user, (err) => {
        if (err) {
          return next(err);
        }
        switch (user.role) {
          case 'order_manager':
            return res.redirect('/order-specifications');
          case 'production_planner':
            return res.redirect('/prod-orders-list');
          default:
            return res.redirect('/');
        }
      });
    })(req, res, next);
  }
  async getUsers(req, res) {
    try {
      const users = await User.find();
      res.json(users);
    } catch (e) {
      console.log(e)
    }
  }
}

module.exports = new authController()
```

ProviderController.js:

```

const Provider = require('../models/Provider');
var csv = require('csvtojson');

const importProvider = async (req, res) => {
  try {
    const providerId = req.params.id;
    const provider = await Provider.findById(providerId);
    if (!provider) {
      return res.status(404).json({ status: 404, success: false, msg: 'Provider
not found' });
    }

    const csvData = await csv().fromFile(req.file.path);
    const materialsData = csvData.map((row) => ({
      category: row.category,
      title: row.title,
      price: parseFloat(row.price),
    }));

    provider.materials = [...provider.materials, ...materialsData];
    await provider.save();
    res.status(200).json({ success: true, provider });
  } catch (error) {
    res.status(400).json({ success: false, msg: error.message });
  }
};

module.exports = {
  importProvider
}

```

Auth.js:

```

module.exports = function(req, res, next) {
  if (req.isAuthenticated()) {
    return next();
  }
  res.redirect('/login');
}

```

CheckRole.js:

```

module.exports = function(allowedRoles) {
  return function(req, res, next) {
    if (req.isAuthenticated() && allowedRoles.includes(req.user.role)) {
      return next();
    }
    res.status(403).send('Access denied');
  }
}

```

User.js:

```
const {Schema, model} = require('mongoose');

const User = new Schema({
  name: {
    type: String,
    required: true
  },

  username: {
    type: String,
    unique: true,
    required: true
  },

  password: {
    type: String,
    required: true
  },

  role: {
    type: String,
    required: true
  }
})

module.exports = model('User', User);
```

Index.css:

```
nav { position: fixed; top: 0; width: 100%; z-index: 1000; }
.user-info { margin-right: 30px; }
.dropdown-content li a { text-align: center; margin-left: 0; padding-left: 0.8rem; }
.container { margin-top: 80px; margin-bottom: 32px; }
.material { margin-bottom: 1rem; }
.material span.completed { text-decoration: line-through; }
.material label { width: 100%; display: flex; align-items: center; justify-content: space-between; cursor: pointer; background: #f3f3f3; padding: 1rem; }
.material i { z-index: 2; }
.chips { padding: 10px; }
.provider-list { list-style: none; padding: 0; }
.provider-item { display: flex; align-items: center; margin-bottom: 20px; }
.provider-logo { width: 100px; height: auto; margin-right: 20px; }
.provider-details { flex-grow: 1; }
```

```

.provider-title { font-size: 18px; font-weight: bold; }
.provider-description { margin-top: 5px; }
.provider-button { margin-top: 10px; }
.provider-title-wrapper { display: flex; align-items: center; margin-bottom: 10px; }
.provider-title { margin-left: 10px; }
.provider-detail { display: flex; align-items: center; margin-bottom: 5px; }
.provider-detail:last-child { margin-bottom: 0; }
.provider-detail span { margin-left: 10px; }
.card-panel span { display: flex; align-items: center; }
.card-panel i { margin-right: 10px; }
.logo { max-height: 60px; }
.dropdown-trigger { outline: none; }
.specifications li { margin-bottom: 20px; }
.collection-item { transition: background-color 0.3s ease; }
.collection-item:hover { background-color: #f5f5f5; }
.input-field { margin-bottom: 0; }
.disabled { background-color: #ccc !important; pointer-events: none; }
.specification-header { display: flex; justify-content: space-between; align-items: center; }
.specification-header h2 { margin-top: 0; }
.QuantityChip { float: right; margin-top: 10px; font-weight: bold; }
.orderRow { cursor: pointer; }
.orderRow:hover { background-color: #f5f5f5; }
.chartControls { display: flex; justify-content: space-between; align-items: center; }
.chart-container { max-width: 800px; margin: 0 auto; height: auto; }
.specificationCheckboxes { display: flex; gap: 20px; }
.orderSpecification-buttons { padding-top: 10px; display: inline-block; gap: 5px; }
.order-progress { background-color: #ffffff; border: 1px solid #dee2e6; border-radius: 5px; padding: 20px; box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); }
.progress-steps { list-style: none; padding: 0; display: flex; justify-content: space-between; margin: 0; }
.step { text-align: center; position: relative; flex: 1; }
.step:not(:last-child)::after { content: ""; position: absolute; top: 50%; right: 0; width: 100%; height: 2px; background-color: #dee2e6; transform: translateY(-50%); z-index: -1; }
.step.active::after { background-color: #007bff; }
.step .step-number { display: inline-block; width: 30px; height: 30px; line-height: 30px; border-radius: 50%; background-color: #dee2e6; color: #ffffff; margin-bottom: 5px; }
.step.active .step-number { background-color: #007bff; }

```

```
.step .step-label { color: #6c757d; }
.step.active .step-label { color: #007bff; }
.step.completed::after { background-color: #007bff; }
.step.completed .step-number { background-color: #007bff; }
.step.completed .step-label { color: #007bff; }
```

Authentication.js:

```
const Router = require('express');
const router = new Router();
const controller = require('../controllers/authController');

router.get('/authentication', async (req, res) => {
  res.render('authentication');
});

router.get('/login', (req, res) => {
  res.render('login', { layout: 'login-layout' });
});

router.post('/login', controller.login)

router.get('/users', controller.getUsers)

router.get('/logout', (req, res) => {
  req.logout((err) => {
    if (err) {
      return next(err);
    }
    res.redirect('/login');
  });
});

module.exports = router
```

Pdf-service.js:

```
const PDFDocument = require('pdfkit');
const fs = require('fs');
function generateHeader(doc) {
  doc.image('./public/images/pdf-logo.png', 50, 45, { height: 50, width: 50 })
  .font("DejavuSans")
  .fillColor('#444444')
  .fontSize(20)
  .text('Innovatech Engineering', 110, 63)
  .fontSize(10)|
  .text('вул. Набережна, 12', 200, 65, { align: 'right' })
  .text('м. Київ, Україна', 200, 80, { align: 'right' })
}
```

```

    .moveDown();
}
function generateFooter(doc) {
    doc.fontSize(
        10,
    ).text(
        'Оплата здійснюється протягом 15 днів. Дякуємо за вашу співпрацю.',
        50,
        700,
        { align: 'center', width: 500 },
    );
}
function generateOrderInformation(doc, specification) {

    var totalPrice = 0;
    doc
        .fillColor("#444444")
        .fontSize(15)
        .font("DejaVuSans-Bold")
        .text(`Замовлення: №${specification._id}`, 50, 160);
    generateHr(doc, 185);
    const orderInformationTop = 200;
    specification.materials.forEach(function (material) {
        totalPrice += material.price * material.quantity;
    });
    doc
        .fontSize(10)
        .font("DejaVuSans-Bold")
        .text("Дата створення:", 50, orderInformationTop)
        .font("helvetica_cyr_oblique")
        .text(formatDate(new Date()), 150, orderInformationTop)
        .font("DejaVuSans-Bold")
        .text("До сплати:", 50, orderInformationTop + 15)
        .font("helvetica_cyr_oblique")
        .text(totalPrice + " UAH", 150, orderInformationTop + 15)
        .font("DejaVuSans-Bold")
        .text("Дата поставки:", 50, orderInformationTop + 30)
        .font("helvetica_cyr_oblique")
        .text(`${specification.date}`, 150, orderInformationTop + 30)
        .font("DejaVuSans-Bold")
        .text("Постачальник:", 50, orderInformationTop + 45)
        .font("DejaVuSans")
        .text(`${specification.provider}`, 150, orderInformationTop + 45)
        .font("DejaVuSans")
        .text("ПП: Олінкевич Я.В.", 300, orderInformationTop)
        .font("DejaVuSans")
        .text("Адреса: Україна, м. Київ, вул. Лісна 24", 300, orderInformationTop +
15)

```

```

        .moveDown();

    generateHr(doc, 272);
}
function generateOrderTable(doc, specification) {
    var totalPrice = 0;
    var i;
    const orderTableTop = 330;

    specification.materials.forEach(function (material) {
        totalPrice += material.price;
    });
    doc.font("DejaVuSans-Bold");
    generateTableRow(
        doc,
        orderTableTop,
        "Матеріал",
        "Кількість (од.)",
        "Вартість (UAH)",
        "Всього (UAH)"
    );
    generateHr(doc, orderTableTop + 20);
    doc.font("DejaVuSans");
    for (i = 0; i < specification.materials.length; i++) {
        const material = specification.materials[i];
        const position = orderTableTop + (i + 1) * 30;
        generateTableRow(
            doc,
            position,
            material.title,
            material.quantity,
            material.price
        );
        generateHr(doc, position + 20);
    }
    const duePosition = orderTableTop + (i + 1) * 30;
    doc.font("DejaVuSans-Bold")
        .text("Загальна сума", 310, duePosition + 20);
    doc.font("DejaVuSans")
        .text(`${totalPrice} UAH`, 480, duePosition + 20);

    // сплачено
    doc.font("DejaVuSans-Bold")
        .text("сплачено", 310, duePosition + 40);
    doc.font("DejaVuSans")
        .text("0 UAH", 480, duePosition + 40);
    doc.font("DejaVuSans-Bold")
        .text("до сплати", 310, duePosition + 60);
}

```

```

doc.font("DejaVuSans")
    .text(`${totalPrice} UAH`, 480, duePosition + 60);
}
function buildPDF(specification, dataCallback, endCallback) {
    const doc = new PDFDocument({ deflateLevel: 9 });
    doc.on('data', dataCallback);
    doc.on('end', endCallback);
    doc.registerFont('DejaVuSans', './public/fonts/djsans/DejaVuSans.ttf');
    doc.registerFont('DejaVuSans-Bold', './public/fonts/djsans/DejaVuSans-
Bold.ttf');
    doc.registerFont('helvetica_cyr_oblique',
'./public/fonts/helvetica/helvetica_cyr_oblique.ttf');
    generateHeader(doc);
    generateOrderInformation(doc, specification);
    generateOrderTable(doc, specification);
    const currentHeight = doc.y;
    const footerHeight = 30;
    const maxHeight = doc.page.height - 50;
    if (currentHeight + footerHeight > maxHeight) {
        doc.addPage();
    }
    generateFooter(doc);
    doc.end();
}
function generateHr(doc, y) {
    doc
        .strokeColor("#aaaaaa")
        .linewidth(1)
        .moveTo(50, y)
        .lineTo(550, y)
        .stroke();
}
function generateTableRow(
    doc,
    y,
    title,
    quantity,
    price
) {
    doc
        .fontSize(10)
        .text(title, 50, y, { width: 90, align: "center" })
        .text(quantity, 250, y, { width: 90, align: "center" })
        .text(price, 450, y, { width: 90, align: "center" })
}

function formatDate(date) {
    const day = date.getDate();

```

```

    const month = date.getMonth() + 1;
    const year = date.getFullYear();
    const monthFormatted = month < 10 ? "0" + month : month;
    return (day < 10 ? "0" : "") + day + "." + monthFormatted + "." + year;
  }
  module.exports = { buildPDF };

```

Reports.hbs:

```

<h2>Статистика закупівель за <span id="currentYear"></span> рік</h2>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<div class="chart-container">
  <canvas id="myChart"></canvas>
  <div class="chartControls">
    <button id="prevYearBtn" class="btn waves-effect waves-light">
      <i class="material-icons prefix">chevron_left</i>
    </button>
    <button id="nextYearBtn" class="btn waves-effect waves-light">
      <i class="material-icons prefix">chevron_right</i>
    </button>
  </div>
</div>

<script>
const labels = {{{ labels }}};
const data = {{{ data }}};

const ctx = document.getElementById("myChart").getContext("2d");
let myChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: labels,
    datasets: [{
      label: 'Місячні витрати',
      data: data,
      borderColor: 'red',
      fill: false
    }]
  },
  options: {
    responsive: true,
    scales: {
      x: {
        display: true,
        title: {
          display: true,
          text: 'Місяць/Рік'
        }
      }
    }
  }
});

```

```

display: true,
  title: {
    display: true,
    text: 'Cyma'
  },
  suggestedMin: 0,
  suggestedMax: Math.max(...data) * 1.2 }
}
}
});

const prevYearBtn = document.getElementById('prevYearBtn');
const nextYearBtn = document.getElementById('nextYearBtn');
const currentYearDisplay = document.getElementById('currentYear');
let currentYear = new Date().getFullYear();

currentYearDisplay.textContent = currentYear;

prevYearBtn.addEventListener('click', () => {
  currentYear--;
  updateChart(currentYear);
});

nextYearBtn.addEventListener('click', () => {
  currentYear++;
  updateChart(currentYear);
});

function updateChart(year) {
  const filteredLabels = labels.filter(label => label.endsWith(`/${year}`));
  const filteredData = filteredLabels.map(label => data[labels.indexOf(label)]);

  myChart.data.labels = filteredLabels;
  myChart.data.datasets[0].data = filteredData;
  myChart.update();

  currentYearDisplay.textContent = year;
}

updateChart(currentYear);
</script>

```

Prod-plnr-navbar.hbs:

```

<nav class="blue darken-4 z-depth-3">
  <div class="nav-wrapper">
    <a href="/prod-orders-list" class="brand-logo left">
      
    </a>

```

```

| <ul id="nav-mobile" class="right hide-on-med-and-down">
  <li class="user-info">
    <i class="material-icons left">account_circle</i>{{name}}, планувальник
виробництва
  </li>

  {{#if isOrders}}
  <li class="active"><a href="/prod-orders-list">Замовлення</a></li>
  {{else}}
  <li><a href="/prod-orders-list">Замовлення</a></li>
  {{/if}}

  {{#if isProducts}}
  <li class="active"><a href="/">Матеріали</a></li>
  {{else}}
  <li><a href="/">Матеріали</a></li>
  {{/if}}

  {{#if isAuthenticated}}
  <li><a href="/logout">Вийти</a></li>
  {{/if}}

</ul>
</div>
</nav>

<script>
document.addEventListener('DOMContentLoaded', function () {
  var elems = document.querySelector('nav');
  var instances = M.Pushpin.init(elems, {
    top: elems.offsetTop
  });
});
</script>

```

