Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматизації, комп'ютерно-інтегрованих технологій та систем
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та б безпекової інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший(бакалаврський)
Розроблення автоматизованої системи планування та обліку виробничих операцій Lean Production
(тема)

Виконав:
здобувач 4 року навчання,
групи АКТАКІТ-22-1
Микита ВОРОБЙОВ
(власне ім'я, прізвище)

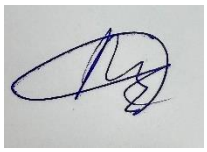
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник проф. Ірина СЕЗОНОВА
(посада, власне ім'я, прізвище)

Допускається до захисту
Завідувач кафедри _____ Ігор НЕВЛЮДОВ
(підпис) (власне ім'я, прізвище)

2026 р.

Я, Воробйов Микита Костянтинович, як здобувач вищої освіти ХНУРЕ, розумію та підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволеної допомоги під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів має містити посилання на відповідне джерело.

5 липня 2026р.  Микита ВОРОБЙОВ

Харківський національний університет радіоелектроніки

Факультет Автоматизації, комп'ютерно-інтегрованих технологій та систем

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії

Рівень вищої освіти перший (бакалаврський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«_____»_____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Воробйову Микита Костянтиновичу
(прізвище, ім'я, по батькові)

1. Розроблення автоматизованої системи планування та обліку виробничих операцій Lean Production
затверджена наказом університету від 04 06 2026 р. № 472 Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 10 07 2026 р.
3. Вихідні дані до роботи. Модель автоматизованої системи планування та обліку виробничих операцій. База даних SQL. Програмний продукт на мові Python для розробки серверної частини (фреймворк FastAPI) та JavaScript для розробки клієнтської частини (бібліотека React 18)
4. Перелік питань, що потрібно опрацювати в роботі
4.1 Вступ
4.2 Аналіз предметної області та існуючих рішень
4.3 Проєктування автоматизованої системи та бази даних
4.4 Розроблення алгоритму автоматичного планування виробництва
4.5 Програмна реалізація серверної та клієнтської частин системи
4.6 Розгортання, тестування системи та аналіз результатів
4.7 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Графічний демонстраційний матеріал у форматі powerpoint(*.ppt), 12 арк. ф. А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк/терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	30.04.2026 - 03.05.2026	Виконано
2	Опрацювання технічної документації за темою	04.05.2026 - 06.05.2026	Виконано
3	Аналіз аналогів та обґрунтування стеку	07.05.2026 - 09.05.2026	Виконано
4	Проєктування архітектури системи та структури бази даних	10.05.2026 - 16.05.2026	Виконано
5	Розробка алгоритму автоматичного планування виробництва	17.05.2026 - 21.05.2026	Виконано
6	Розробка серверної частини	22.05.2026 - 28.05.2026	Виконано
7	Розробка клієнтської частини	29.05.2026 - 03.06.2026	Виконано
8	Розгортання системи та тестування запитів	04.06.2026 - 06.06.2026	Виконано
9	Оформлення пояснювальної записки	07.06.2026 - 15.06.2026	Виконано

Дата видачі завдання 27 04 2026 р.

Здобувач _____ Микита ВОРОБЙОВ
(підпис) (власне ім'я, прізвище)

Керівник роботи _____ проф. Ірина СЕЗОНОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 60 с., 8 табл., 19 рис., 3 дод., 14 джерел.

АВТОМАТИЗОВАНА СИСТЕМА, ПЛАНУВАННЯ ВИРОБНИЦТВА, ОБЛІК ВИРОБНИЧИХ ОПЕРАЦІЙ, LEAN PRODUCTION, АВТОПЛАНУВАЛЬНИК, ВЕБ-ЗАСТОСУНОК, KPI, БАЗА ДАНИХ.

Об’єкт розробки – процеси планування та обліку виробничих операцій радіоелектронного підприємства.

Мета роботи – розроблення автоматизованої системи планування та обліку виробничих операцій Lean production з автоматичним формуванням виробничих завдань за технологічним маршрутом виробу.

Методи розроблення – аналіз рішень класів ERP, MES, APS; проєктування реляційної бази даних методом «сутність-зв’язок»; трирівнева веб-архітектура; алгоритмічне планування виробничих завдань з урахуванням норм часу та завантаженості обладнання.

У результаті розроблено веб-систему на базі FastAPI, PostgreSQL та React. Ядром системи є автоматичний планувальник, який генерує виробничі завдання, балансує завантаження обладнання та розраховує строки виконання і показники ефективності (KPI).

Результати роботи можуть бути використані на малих та середніх підприємствах радіоелектронної галузі.

Отримані результати відповідають Цілям сталого розвитку 9 «Промисловість, інновації та інфраструктура» та 12 «Відповідальне споживання та виробництво», оскільки система сприяє модернізації виробництва та скороченню втрат за принципами Lean production.

ABSTRACT

Bachelor’s thesis: 60 pages, 8 tables, 19 figures, 3 appendices, 14 sources.

AUTOMATED SYSTEM, PRODUCTION PLANNING, MANUFACTURING OPERATIONS ACCOUNTING, LEAN PRODUCTION, AUTOMATIC SCHEDULER, WEB APPLICATION, KPI, DATABASE.

The object of development is the processes of planning and accounting of manufacturing operations at a radio-electronic enterprise.

The goal of the work is the development of an automated Lean production planning and accounting system with automatic generation of production tasks along the technological route of a product.

Development methods include analysis of ERP, MES and APS class solutions; relational database design using the entity-relationship method; three-tier web architecture; algorithmic scheduling of production tasks considering time standards and equipment workload.

As a result, a web system based on FastAPI, PostgreSQL and React was developed. Its core is an automatic scheduler that generates production tasks, balances equipment workload and calculates deadlines and key performance indicators (KPI).

The results of the work can be applied at small and medium-sized radio-electronic enterprises.

The obtained results correspond to Sustainable Development Goals 9 "Industry, Innovation and Infrastructure" and 12 "Responsible Consumption and Production", as the system promotes production modernization and waste reduction in line with Lean production principles.

ЗМІСТ

Скорочення та умовні познаки 9
Вступ 10
1 Аналіз предметної області та існуючих рішень 12
1.1 Актуальність та постановка задачі 12
1.2 Огляд існуючих систем планування та обліку виробництва 14
1.3 Аналіз структур баз даних для виробничих систем 17
1.4 Аналіз методів розроблення програмних застосунків 18
1.5 Застосування принципів теорії автоматичного керування в системах планування виробництва 20
2 Проєктування автоматизованої системи 22
2.1 Структурна схема та архітектура системи 22
2.2 Проєктування бази даних 24
2.3 Алгоритм автоматичного планування виробництва 28
2.4 Проєктування інтерфейсу користувача 31
3 Програмна реалізація системи 33
3.1 Реалізація серверної частини 33
3.2 Реалізація автопланувальника та аналітики 35
3.3 Реалізація клієнтської частини 36
3.4 Розгортання та тестування системи 39
3.5 Приклад роботи автоматичного планувальника 41
3.6 Охорона праці при експлуатації автоматизованої системи 42
Висновки 45
Перелік джерел посилання 47
Додаток А Лістинги ключових фрагментів програмного коду 49
Додаток Б Супровідні та цифрові матеріали проєкту 53
Додаток В Демонстраційний матеріал 54

ПЕРЕЛІК СКОРОЧЕНЬ

АСУ – автоматизована система управління;

БД – база даних;

КІТАР – комп’ютерно-інтегровані технології, автоматизація та робототехніка;

СУБД – система управління базами даних;

ХНУРЕ – Харківський національний університет радіоелектроніки;

3НФ – третя нормальна форма;

API – Application Programming Interface (інтерфейс прикладного програмування);

APS – Advanced Planning and Scheduling (вдосконалене планування та диспетчеризація);

BOM – Bill of Materials (специфікація матеріалів);

CRUD – Create, Read, Update, Delete (створення, читання, оновлення, видалення);

ER – Entity-Relationship (сутність-зв’язок);

ERP – Enterprise Resource Planning (планування ресурсів підприємства);

JSON – JavaScript Object Notation (об’єктна нотація JavaScript);

JWT – JSON Web Token (вебтокен JSON);

KPI – Key Performance Indicator (ключовий показник ефективності);

MES – Manufacturing Execution System (система управління виробництвом);

OEE – Overall Equipment Effectiveness (загальна ефективність обладнання);

ORM – Object-Relational Mapping (об’єктно-реляційне відображення);

PaaS – Platform as a Service (платформа як послуга);

REST – Representational State Transfer (передача репрезентативного стану);

SPA – Single Page Application (односторінковий застосунок);

SQL – Structured Query Language (мова структурованих запитів);

UI – User Interface (інтерфейс користувача).

ВСТУП

Сучасні підприємства функціонують в умовах жорсткої конкуренції та постійно зростаючих вимог до якості продукції й ефективності виробничих процесів. Одним із ключових чинників підвищення конкурентоспроможності підприємств радіоелектронної галузі є впровадження автоматизованих систем управління виробництвом, що дозволяють оптимізувати планування виробничих операцій, забезпечити точний облік ресурсів і мінімізувати виробничі втрати.

Більшість малих і середніх підприємств радіоелектронної галузі не мають доступних програмних рішень, які враховували б специфіку їхніх виробничих процесів – маршрутну технологію виготовлення, нормування праці та облік комплектуючих. Наявні на ринку системи класів ERP, MES та APS орієнтовані на великі підприємства, мають високу вартість впровадження та складні у налаштуванні. Через це розробка спеціалізованої автоматизованої системи планування та обліку виробничих операцій для радіоелектронного підприємства є актуальним завданням.

Концепція Lean production, покладена в основу системи, передбачає мінімізацію втрат у виробничому процесі. Ключовим інструментом її реалізації є автоматизація рутинних операцій планування – формування виробничих завдань, розподіл їх по обладнанню та розрахунок строків виконання, що традиційно виконуються вручну й займають значний час.

Метою кваліфікаційної роботи є розроблення автоматизованої системи планування та обліку виробничих операцій підприємства Lean production на основі реляційної бази даних, що забезпечує автоматичне формування виробничого розкладу.

Об’єкт розробки – процеси планування та обліку виробничих операцій підприємства.

Предмет розробки – структура бази даних, алгоритми планування виробничих операцій та програмний застосунок для взаємодії користувача із системою.

Для досягнення поставленої мети у роботі вирішено такі задачі:

– проаналізовано предметну область та існуючі системи планування й обліку виробничих операцій;

– обґрунтовано вибір технологічного стеку та спроєктовано архітектуру системи;

– спроєктовано реляційну базу даних методом «сутність-зв’язок» з нормалізацією до третьої нормальної форми;

– розроблено алгоритм автоматичного планування виробничих завдань за технологічним маршрутом виробу;

– реалізовано серверну частину з REST API та клієнтський веб-інтерфейс;

– реалізовано автоматичний розрахунок ключових показників ефективності виробництва;

– здійснено розгортання системи у хмарному середовищі та її тестування.

Практична цінність роботи полягає у створенні працюючої автоматизованої системи, придатної для впровадження на малих та середніх підприємствах радіоелектронної галузі. Ключовою особливістю системи є автоматичний планувальник, який звільняє персонал від ручного складання виробничого розкладу.

Кваліфікаційну роботу виконано згідно з ДСТУ 3008:2015 [1] та з урахуванням рекомендацій навчального посібника з підготовки кваліфікаційної роботи бакалавра [2] і відповідних методичних вказівок [3].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

Сучасні промислові підприємства функціонують в умовах жорсткої конкуренції та постійно зростаючих вимог до якості продукції й ефективності виробничих процесів. Одним із ключових чинників підвищення конкурентоспроможності таких підприємств є впровадження автоматизованих систем управління виробництвом, що дозволяють оптимізувати планування виробничих операцій, забезпечити точний облік ресурсів і мінімізувати виробничі витрати. Цей розділ присвячено аналізу предметної області, дослідженню існуючих програмних рішень для планування й обліку виробництва, а також обґрунтуванню доцільності розроблення спеціалізованої системи.

1.1 Актуальність та постановка задачі

Ринок програмного забезпечення для управління виробництвом демонструє стабільне зростання. За оцінками аналітичних агентств, обсяг ринку систем управління виробництвом (MES-систем) до 2027 року може досягти 23,8 млрд доларів США [6]. Водночас більшість малих і середніх підприємств радіоелектронної галузі не мають доступних рішень, які враховували б специфіку їхніх виробничих процесів, зокрема маршрутну технологію виготовлення, нормування праці та облік комплектуючих.

Цифровізація виробничих процесів є однією з головних тенденцій розвитку сучасної промисловості. Впровадження автоматизованих інформаційних систем у виробничий процес дозволяє скоротити час виготовлення продукції, зменшити кількість помилок при плануванні, підвищити прозорість виробничого процесу та забезпечити швидкий доступ до актуальної інформації про стан замовлень, завантаженість обладнання та наявність матеріалів. Для радіоелектронних підприємств, що виготовляють складну технічну продукцію з великою кількістю

найменувань комплектуючих та багатоступеневою маршрутною технологією, ці переваги є особливо цінними.

В умовах воєнного часу та повоєнного відновлення українські підприємства радіоелектронної галузі стикаються з додатковими викликами: нестабільністю ланцюгів постачання, дефіцитом кваліфікованих кадрів, необхідністю швидкого переорієнтування виробництва. Саме тому автоматизація планування та обліку виробничих операцій набуває особливого значення – вона дозволяє мінімізувати людський фактор при плануванні, оперативно реагувати на зміни у виробничих планах та забезпечити безперервність обліку навіть за умов часткової зміни персоналу.

Концепцію Lean production (ощадливе виробництво) [4], покладену в основу системи, спрямовано на мінімізацію усіх видів втрат у виробничому процесі – надлишкових запасів, простоїв обладнання, зайвих переміщень, очікування. Одним із ключових інструментів реалізації цієї концепції є візуалізація виробничих процесів та автоматизація рутинних операцій планування, що дозволяє звільнити персонал від ручної праці й зосередити його на прийнятті управлінських рішень.

Таким чином, актуальність роботи зумовлена потребою малих та середніх радіоелектронних підприємств у доступній спеціалізованій системі планування й обліку виробництва, яка б автоматизувала формування виробничого розкладу.

Метою роботи є розроблення автоматизованої системи планування та обліку виробничих операцій підприємства Lean production на основі реляційної бази даних. Для досягнення поставленої мети необхідно вирішити такі задачі:

– проаналізувати предметну область та існуючі системи планування й обліку виробничих операцій класів ERP, MES, APS;

– обґрунтувати вибір технологічного стеку та архітектури системи;

– спроєктувати структуру реляційної бази даних для зберігання технологічної, ресурсної та облікової інформації;

– розробити алгоритм автоматичного планування виробничих завдань за технологічним маршрутом виробу;

– реалізувати серверну частину системи з REST API та клієнтський веб-інтерфейс;

– реалізувати автоматичний розрахунок показників ефективності виробництва;

– розгорнути систему у хмарному середовищі та провести тестування.

Об’єктом розробки є процеси планування та обліку виробничих операцій підприємства. Предметом розробки є структура бази даних, алгоритми планування виробничих операцій та програмний застосунок для взаємодії користувача із системою.

1.2 Огляд існуючих систем планування та обліку виробництва

Автоматизовані системи планування та обліку виробничих операцій відіграють ключову роль у підвищенні ефективності виробничих підприємств. Залежно від масштабу підприємства та функціональних потреб, такі системи можна класифікувати на декілька основних категорій.

1.2.1 Системи класу ERP

Системи класу ERP (Enterprise Resource Planning) – це комплексні рішення, що охоплюють усі аспекти діяльності підприємства: виробництво, фінанси, логістику, кадровий облік тощо. Найбільш поширеними представниками цього класу є SAP ERP, Oracle E-Business Suite, Microsoft Dynamics 365. Незважаючи на широкий функціонал, ERP-системи мають суттєві недоліки: висока вартість ліцензій та впровадження (що може сягати сотень тисяч доларів), складність налаштування під специфічні потреби підприємства, тривалий термін впровадження, який нерідко вимірюється роками. Для малого підприємства радіоелектронної галузі впровадження повноцінної ERP-системи є економічно недоцільним.

1.2.2 Системи класу MES

Системи класу MES (Manufacturing Execution System) [5] призначені для управління безпосередньо виробничими процесами в реальному часі. Вони забезпечують моніторинг виконання виробничих операцій, управління технологічними маршрутами, контроль якості та збір виробничої статистики. Прикладами таких систем є Siemens Opcenter, Rockwell Automation FactoryTalk та AVEVA MES. Ці рішення орієнтовані переважно на великі промислові підприємства та потребують значної IT-інфраструктури, інтеграції з обладнанням через промислові протоколи, а також штату кваліфікованих спеціалістів для обслуговування.

1.2.3 Системи класу APS

Спеціалізовані системи планування виробництва (Advanced Planning and Scheduling, APS) призначені для оптимізації виробничих планів з урахуванням обмежень за ресурсами, технологічними маршрутами та строками виконання замовлень. Такі системи, як Preactor, Opcenter APS та PlanetTogether, дозволяють будувати деталізовані виробничі розклади з використанням складних алгоритмів оптимізації. Проте вони також є дорогими, складними у налаштуванні та зазвичай потребують інтеграції з наявною ERP-системою, що робить їх непридатними для самостійного впровадження на малому підприємстві.

1.2.4 Вітчизняні та відкриті рішення

На українському ринку присутні такі продукти, як «Парус», «Галактика ERP», а також численні власні розробки підприємств. Ці системи частково охоплюють функції планування та обліку виробничих операцій, проте переважно орієнтовані на бухгалтерський та управлінський облік, а не на детальне оперативне планування виробництва. Крім того, у зв’язку зі змінами у законодавстві щодо використання програмного забезпечення російського походження, підприємства активно шукають альтернативні рішення.

Серед відкритих рішень з відкритим кодом варто відзначити Odoo (модуль Manufacturing), OpenBravo та Dolibarr. Ці продукти можуть бути корисними як відправна точка, однак їх адаптація під специфіку конкретного радіоелектронного підприємства, як правило, потребує значних витрат часу та кваліфікованих розробників, що нівелює початкову перевагу у вартості ліцензій. Більш того, підтримка та розвиток такого рішення повністю лягає на плечі підприємства, що є суттєвим ризиком для безперервності виробничого процесу.

Систематизовані переваги та недоліки розглянутих класів систем наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз існуючих систем управління виробництвом

Клас системи	**Переваги**	**Недоліки**
ERP	Комплексне охоплення всіх процесів, інтеграція між модулями	Висока вартість, складність налаштування, тривале впровадження
MES	Управління виробництвом у реальному часі, моніторинг операцій	Орієнтація на велике виробництво, значна вартість інфраструктури
APS	Оптимізація розкладів з урахуванням обмежень	Складне налаштування, потребує інтеграції з ERP
Спеціалізована розробка	Врахування специфіки підприємства, нижча вартість володіння	Потребує часу на розробку, обмежена підтримка

Аналіз показав, що для малих і середніх радіоелектронних підприємств України, які не мають значних бюджетів на IT-інфраструктуру, застосування «коробкових» рішень класів ERP, MES, APS є економічно недоцільним. Натомість доцільним є розроблення спеціалізованої веб-системи, що поєднуватиме базовий функціонал планування та обліку виробничих операцій (характерний для MES/APS) з можливістю гнучкого налаштування під специфіку конкретного виробництва та доступною вартістю володіння.

1.3 Аналіз структур баз даних для виробничих систем

База даних є ключовим компонентом будь-якої автоматизованої виробничої системи. Вибір моделі даних і архітектури бази даних суттєво впливає на продуктивність системи, зручність її підтримки та розширюваність.

Реляційні бази даних є найбільш поширеним рішенням для виробничих систем. Вони базуються на моделі таблиць зі зв’язками між ними та забезпечують цілісність даних завдяки транзакційному механізму. СУБД PostgreSQL, MySQL та Microsoft SQL Server широко використовуються у виробничих системах завдяки надійності, підтримці складних SQL-запитів та розвиненій екосистемі інструментів.

Для автоматизованої системи планування та обліку виробничих операцій радіоелектронного підприємства доцільно використати реляційну модель даних з такими основними сутностями: вироби, матеріали та комплектуючі, технологічні маршрути, виробничі операції, замовлення, виробничі завдання, обладнання та персонал. Зв’язки між цими сутностями визначають структуру виробничого процесу.

Проаналізувавши відомі підходи до проєктування баз даних для виробничих систем, можна виділити такі типові шаблони структур даних:

– «Специфікація матеріалів» (Bill of Materials, BOM) – деревоподібна структура для опису складу виробу;

– «Технологічний маршрут» – послідовність операцій виготовлення виробу з нормами часу;

– «Виробниче замовлення» – зв’язок між замовленням клієнта та виробничими завданнями;

– «Облік ресурсів» – структура для відстеження використання матеріалів та трудових ресурсів.

Окрім реляційних, у сучасних виробничих системах можна зустріти бази даних типу NoSQL – документоорієнтовані (MongoDB), колонкові (Apache Cassandra) та графові (Neo4j). Документоорієнтовані бази даних добре підходять для збереження неструктурованих даних, наприклад технологічних карт у форматі JSON. Графові бази даних ефективні для моделювання складних мереж залежностей між виробничими операціями. Проте для задачі планування та обліку виробничих операцій, де важлива цілісність транзакцій та чіткі реляційні зв’язки між сутностями, реляційна СУБД залишається оптимальним вибором.

З огляду на наведене, реляційну СУБД PostgreSQL [7] обрано як основну завдяки відкритому коду, підтримці розширених типів даних, надійності, дотриманню принципів ACID та широкому поширенню в промислових рішеннях.

1.4 Аналіз методів розроблення програмних застосунків

Для реалізації клієнтського інтерфейсу автоматизованої системи розглянуто два основні підходи: desktop-застосунок та web-застосунок.

Desktop-застосунки виконуються безпосередньо на операційній системі користувача та забезпечують повний доступ до ресурсів комп’ютера. Їх перевагами є висока продуктивність, можливість роботи без мережевого з’єднання та глибока інтеграція з ОС. Водночас, desktop-застосунки потребують окремої інсталяції та обслуговування на кожному робочому місці, що ускладнює їх підтримку в умовах підприємства з багатьма користувачами.

Web-застосунки доступні через браузер і не потребують встановлення на робочих місцях, що значно спрощує розгортання та оновлення системи. Такі застосунки добре підходять для корпоративних інформаційних систем, де кілька користувачів працюють з єдиною базою даних одночасно. Архітектура web-застосунку зазвичай включає клієнтську частину (frontend) та серверну частину (backend), що взаємодіють через мережевий протокол. З огляду на специфіку задачі

– багатокористувацький доступ до спільної бази даних – для розроблюваної системи обрано саме веб-архітектуру.

Для побудови клієнтської частини розглядалися JavaScript-фреймворки React.js, Vue.js та Angular. React.js обрано як найбільш поширений інструмент [10] з розвиненою екосистемою компонентів, великою спільнотою та стабільною підтримкою. Для побудови візуальних компонентів інтерфейсу використано бібліотеку Ant Design [11], що пропонує понад 60 готових компонентів промислового рівня.

Для серверної частини розглядалися FastAPI (Python), Node.js з Express та Django (Python). Порівняльний аналіз технологій наведено в таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз технологій розроблення серверної частини

Технологія	**Мова**	**Переваги**	**Недоліки**
FastAPI	Python	Автодокументація, асинхронність, типізація через Pydantic	Відносно молода технологія
Django	Python	Вбудована адмін-панель, ORM, зрілість	Надлишковість для невеликих проєктів
Express.js	Node.js	Висока продуктивність, гнучкість	Відсутність жорсткої структури

FastAPI обрано як оптимальне рішення [8] завдяки автоматичній генерації документації API (Swagger UI), підтримці асинхронного програмування, типізації даних через Pydantic та простоті інтеграції з ORM-бібліотекою SQLAlchemy [9] для роботи з PostgreSQL.

При виборі між монолітною та мікросервісною архітектурою для розроблюваної системи обрано монолітний підхід. Враховуючи, що система розробляється для конкретного підприємства і не передбачає масштабування до рівня хмарного сервісу з мільйонами користувачів, монолітна архітектура є простішою у розробці, тестуванні та розгортанні. Мікросервісна архітектура, попри

переваги у масштабованості, вимагає значно більших витрат на інфраструктуру та складнішого управління.

Таким чином, за результатами аналізу для реалізації автоматизованої системи обрано веб-архітектуру з технологічним стеком: PostgreSQL (база даних), FastAPI із SQLAlchemy (серверна частина) та React.js (клієнтська частина). Обґрунтування архітектури та детальне проєктування системи наведено у розділі 2.

1.5 Застосування принципів теорії автоматичного керування в системах планування виробництва

З позицій теорії автоматичного керування виробничий процес підприємства доцільно розглядати як об’єкт керування, а автоматизовану систему планування та обліку виробничих операцій – як замкнену систему керування зі зворотним зв’язком. Такий підхід дозволяє формально обґрунтувати архітектуру системи та алгоритм автоматичного планування, що розробляються у цій роботі.

Роль задавального впливу у такій системі виконує плановий графік виробництва: перелік виробничих завдань з призначеним обладнанням, плановими строками початку і завершення та плановою кількістю продукції. Задавальний вплив формується автоматичним планувальником на підставі технологічного маршруту виробу та норм часу на операції.

Об’єктом керування є виробничий процес, вихідними координатами якого виступають фактичні строки виконання операцій, фактично виготовлена кількість продукції та поточне завантаження одиниць обладнання. Канал зворотного зв’язку реалізується підсистемою обліку: оператори фіксують у системі початок і завершення операцій та фактичний обсяг виробітку, завдяки чому система у кожний момент часу має актуальну інформацію про стан об’єкта.

Пристроєм порівняння слугує аналітичний модуль системи, який обчислює розсогласування – відхилення фактичних показників від планових: відставання від графіка за строками та відхилення виготовленої кількості від планової. За

величиною розсогласування система автоматично формує сигнали для користувачів (індикація відставань) та надає керуючий вплив – перепланування замовлення, під час якого планувальник перерозподіляє незавершені завдання по обладнанню та перераховує строки їх виконання.

Оскільки облік виробничих операцій ведеться подієво, розглянута система керування має дискретний характер: інформація про стан об’єкта оновлюється у моменти реєстрації подій, а керуючий вплив формується за запитом користувача або за фактом виявлення відставання. Критеріями якості керування у такій постановці є мінімізація відхилення фактичних строків виконання замовлень від планових та рівномірність завантаження виробничого обладнання.

Таким чином, розроблювана система реалізує класичний замкнений контур керування «планування – виконання – облік – аналіз відхилень – перепланування», що відповідає принципам теорії автоматичного керування та є формальним підґрунтям для алгоритму автоматичного планування, наведеного у пункті 2.3.

2 ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

У цьому розділі наведено результати проєктування автоматизованої системи планування та обліку виробничих операцій. Описано структурну схему та трирівневу архітектуру системи, спроєктовано структуру реляційної бази даних з усіма сутностями предметної області, а також розроблено алгоритм автоматичного планування виробничих завдань, що становить ядро автоматизації системи.

2.1 Структурна схема та архітектура системи

Відповідно до результатів аналізу, проведеного в розділі 1, розроблювана система являє собою веб-застосунок з трирівневою архітектурою. Така архітектура забезпечує розподіл відповідальності між компонентами системи, полегшує її підтримку та подальше розширення. Виокремлюють три рівні: рівень представлення (клієнтська частина), рівень бізнес-логіки (серверна частина) та рівень даних (база даних).

Рівень представлення реалізовано у вигляді односторінкового застосунку (SPA) на основі бібліотеки React.js. Він відповідає за відображення інтерфейсу користувача, обробку дій користувача та обмін даними із серверною частиною через HTTP-запити. Рівень бізнес-логіки реалізовано засобами фреймворку FastAPI; він надає програмний інтерфейс REST API, виконує валідацію даних, реалізує алгоритми планування й розрахунку показників, забезпечує автентифікацію та авторизацію користувачів. Рівень даних реалізовано на основі реляційної СУБД PostgreSQL, що зберігає всю технологічну, ресурсну та облікову інформацію.

Загальна структурна схема системи з відображенням взаємодії між компонентами наведена на рисунку 2.1.

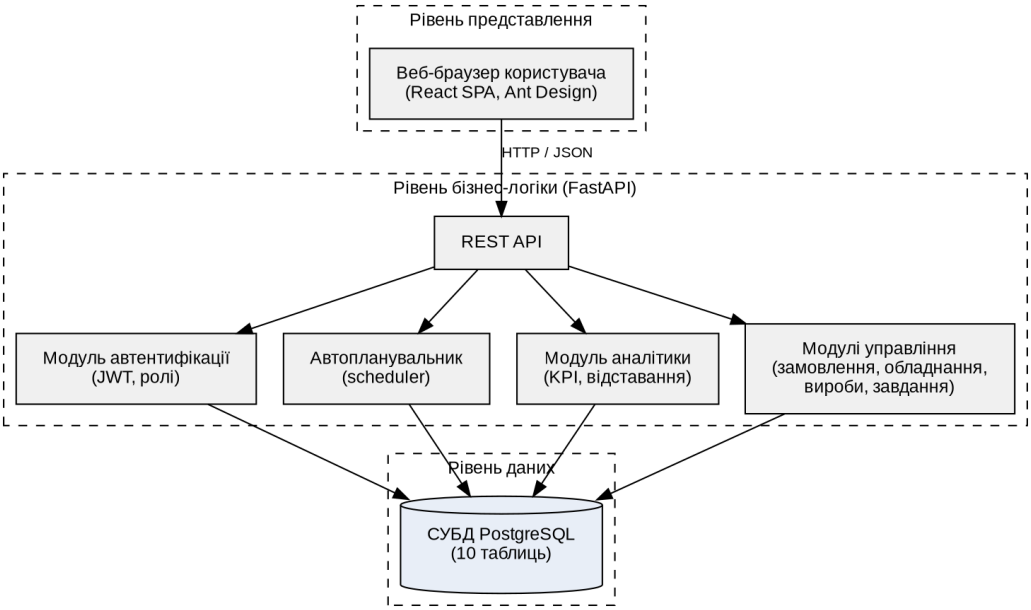


Рисунок 2.1 – Структурна схема автоматизованої системи

Взаємодія між клієнтською та серверною частинами здійснюється через REST API [13] з використанням формату обміну даними JSON. Серверна частина надає набір кінцевих точок (endpoints), згрупованих за функціональними модулями. Кожен запит від клієнта, що потребує авторизації, супроводжується токеном JWT [12], який сервер перевіряє перед виконанням операції.

Серверна частина системи реалізує такі функціональні модулі:

– модуль управління нормативно-довідковою інформацією (вироби, операції, обладнання, типи обладнання);

– модуль технологічних маршрутів (послідовності операцій виготовлення виробів);

– модуль управління замовленнями (реєстрація, статуси, терміни виконання);

– модуль автоматичного планування виробничих завдань (розподіл за часом та ресурсами);

– модуль обліку виконання операцій (фактичні дані, відхилення);

– модуль аналітики (розрахунок показників ефективності, виявлення відставань);

– модуль автентифікації та авторизації (управління користувачами та ролями).

Автентифікація користувачів реалізована на основі токенів JWT (JSON Web Token). Система підтримує чотири ролі користувачів: адміністратор, плановик, технолог та оператор виробничої дільниці. Розмежування доступу до функцій системи виконується на рівні серверного API з перевіркою ролі, закодованої у токені. Розподіл прав доступу за ролями наведено в таблиці 2.1.

Таблиця 2.1 – Розподіл прав доступу за ролями користувачів

Роль	**Основні права та функції**
Адміністратор	Повний доступ до всіх функцій, управління користувачами, видалення замовлень
Плановик	Створення та редагування замовлень, запуск автопланування, перепланування
Технолог	Управління довідником обладнання та виробів, технологічними маршрутами
Оператор	Перегляд та облік виконання виробничих завдань, зміна їх статусів

Для забезпечення безпеки паролі користувачів зберігаються не у відкритому вигляді, а у формі хешів, отриманих за алгоритмом bcrypt. Це унеможливлює відновлення паролів навіть у разі компрометації бази даних. Система також підтримує журналювання дій користувачів (audit log): кожна операція зміни даних фіксується у спеціальній таблиці з зазначенням ідентифікатора користувача, типу дії, часу та деталей операції, що забезпечує прозорість роботи системи.

2.2 Проєктування бази даних

База даних є центральним компонентом розроблюваної системи. Вона зберігає всю технологічну, ресурсну та облікову інформацію, необхідну для планування та обліку виробничих операцій. Проєктування бази даних виконано з

використанням методу «сутність-зв’язок» (ER-моделювання). На першому етапі визначено основні сутності предметної галузі, на другому – атрибути сутностей та зв’язки між ними, на третьому – виконано нормалізацію структури до третьої нормальної форми (3НФ), що усуває надмірність даних та аномалії при їх зміні.

Структура бази даних включає десять основних таблиць. Центральною сутністю предметної області є виріб (products), для якого визначається технологічний маршрут (routes) – впорядкована послідовність операцій (route_operations), кожна з яких посилається на довідник операцій (operations). Замовлення (orders) пов’язане з конкретним виробом, а в результаті автоматичного планування для нього генеруються виробничі завдання (production_tasks), кожне з яких призначається на одиницю обладнання (equipment). Перелік основних таблиць бази даних та їх призначення наведено в таблиці 2.2.

Таблиця 2.2 – Перелік основних таблиць бази даних

Назва таблиці	**Призначення**
users	Облікові записи користувачів системи та їх ролі
equipment_types	Довідник типів обладнання (SMT-лінія, пайка тощо)
equipment	Перелік виробничого обладнання підприємства
products	Перелік виробів, що виготовляються
operations	Довідник технологічних операцій з нормами часу
routes	Технологічні маршрути виготовлення виробів
route_operations	Послідовність операцій у складі маршруту
orders	Реєстр виробничих замовлень
production_tasks	Виробничі завдання, згенеровані планувальником
audit_log	Журнал дій користувачів системи

Розглянемо призначення та структуру ключових таблиць детальніше.

Таблиця products (Вироби) зберігає перелік виробів, що виготовляються підприємством. Основні поля: id (первинний ключ), code (код виробу), name

(найменування), description (опис), unit_of_measure (одиниця виміру), is_active (ознака активності запису).

Таблиця operations (Виробничі операції) містить довідник технологічних операцій. Ключовими для автоматичного планування є поля norm_time_min (норма часу на одиницю виробу у хвилинах) та setup_time_min (підготовчо-заключний час на партію). Поле equipment_type_id визначає тип обладнання, потрібний для виконання операції.

Таблиця routes (Технологічні маршрути) зберігає маршрути виготовлення виробів, а її деталізація – послідовність операцій – описується в таблиці route_operations, що містить поле sequence_number (порядковий номер операції в маршруті). Така структура дозволяє для кожного виробу задати власну технологічну послідовність.

Таблиця orders (Замовлення) реалізує реєстр виробничих замовлень. Поля: id, order_number, product_id, quantity, planned_start_date, planned_end_date, status, priority, created_by, created_at. Поле planned_end_date розраховується автоматично планувальником, а не вводиться вручну.

Таблиця production_tasks (Виробничі завдання) деталізує замовлення до рівня конкретних операцій, призначених на обладнання. Ця таблиця заповнюється автоматично планувальником. Структуру її полів наведено в таблиці 2.3.

Для забезпечення посилальної цілісності між таблицями використовуються зовнішні ключі (FOREIGN KEY) з відповідними обмеженнями. Наприклад, при видаленні замовлення каскадно видаляються пов’язані з ним виробничі завдання, що запобігає утворенню «осиротілих» записів. Для часто виконуваних запитів – фільтрації замовлень за статусом, отримання завдань за обладнанням – передбачено індекси на відповідних полях, що забезпечує прийнятну продуктивність навіть при великій кількості записів.

Таблиця 2.3 – Структура полів таблиці production_tasks

Поле	Тип даних	Призначення
id	INTEGER	Первинний ключ
order_id	INTEGER	Посилання на замовлення (FK)
operation_name	VARCHAR	Найменування операції
sequence_number	INTEGER	Порядковий номер у маршруті
equipment_id	INTEGER	Призначене обладнання (FK)
planned_start	TIMESTAMP	Плановий початок
planned_end	TIMESTAMP	Планове завершення
quantity_planned	NUMERIC	Планова кількість
quantity_done	NUMERIC	Фактично виготовлено
status	VARCHAR	Статус виконання
planned_duration_min	FLOAT	Розрахункова тривалість, хв

Концептуальну модель бази даних у вигляді ER-діаграми, що відображає сутності та зв’язки між ними, наведено на рисунку 2.2.

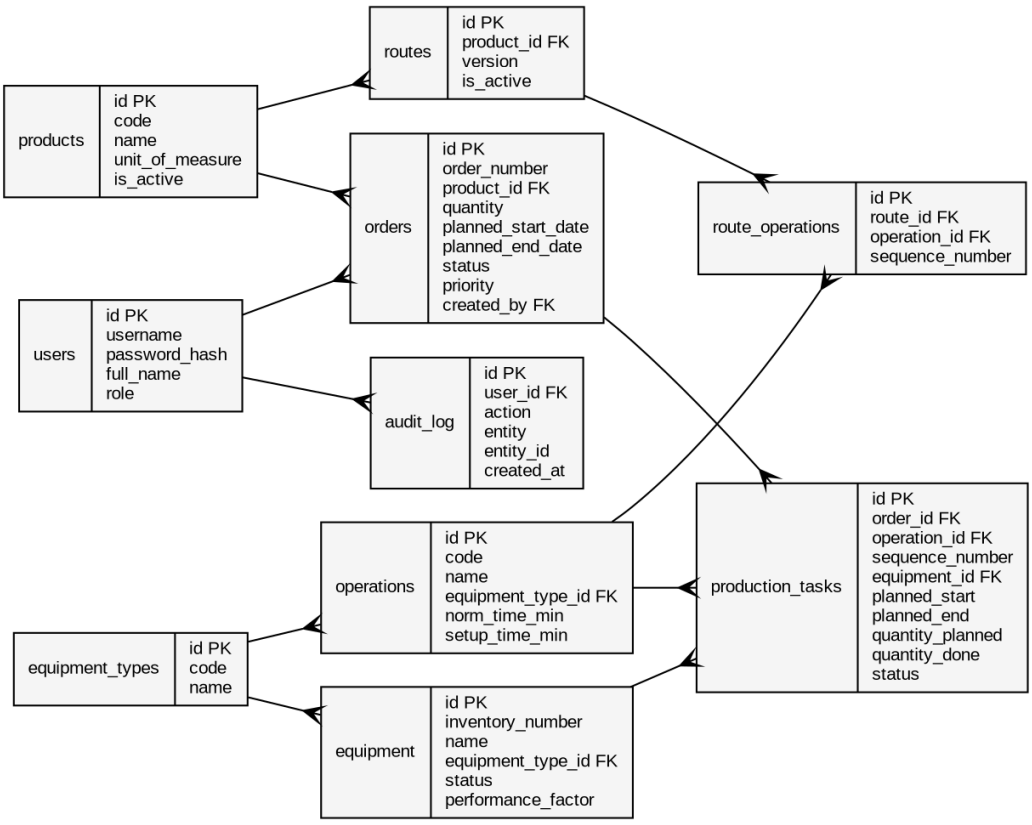


Рисунок 2.2 – ER-діаграма бази даних системи

2.3 Алгоритм автоматичного планування виробництва

Алгоритм автоматичного планування виробничих завдань є ключовим елементом автоматизації системи. Саме він відрізняє розроблювану систему від простого засобу обліку (електронного журналу) та реалізує принцип Lean production щодо звільнення персоналу від рутинної ручної праці. Завдання алгоритму – на основі замовлення автоматично сформувати повний виробничий розклад без участі людини.

Вхідними даними для алгоритму є замовлення, що містить посилання на виріб, потрібну кількість, планову дату початку та пріоритет. Результатом роботи алгоритму є набір виробничих завдань, кожне з яких призначене на конкретну одиницю обладнання та має розраховані планові строки початку й завершення. Алгоритм виконує таку послідовність кроків:

– крок 1 – пошук активного технологічного маршруту виробу, вказаного в замовленні;

– крок 2 – послідовний перебір операцій маршруту в порядку їх виконання;

– крок 3 – для кожної операції - підбір одиниці обладнання потрібного типу за принципом найменшої завантаженості;

– крок 4 – розрахунок тривалості операції за нормою часу з урахуванням продуктивності обладнання;

– крок 5 – визначення планових строків операції з урахуванням робочого календаря;

– крок 6 – розрахунок планової дати завершення всього замовлення.

Розрахунок тривалості операції виконується за формулою, що враховує норму часу, кількість виробів, коефіцієнт продуктивності обладнання та підготовчо-заключний час. Тривалість операції T (у хвилинах) обчислюється як:

$$T = (t_{норм} \cdot Q) / K_{прод} + t_{нал}, \qquad (2.1)$$

де $t_{норм}$ – норма часу на одиницю виробу, хв;

Q – кількість виробів у замовленні;

$K_{прод}$ – коефіцієнт продуктивності обладнання (1,0 відповідає нормі, більші значення – швидшому обладнанню);

$t_{нал}$ – підготовчо-заключний час (час налагодження), хв.

Підбір обладнання (крок 3) реалізує балансування завантаження. Серед усіх активних одиниць обладнання потрібного типу алгоритм обирає ту, що має найменшу поточну сумарну завантаженість за вже призначеними завданнями. Завдяки цьому замовлення рівномірно розподіляються між однотипним обладнанням, а не накопичуються на одній одиниці, що відповідає принципам ощадливого виробництва щодо запобігання простоям та «вузьким місцям».

Визначення строків (крок 5) виконується з урахуванням робочого календаря підприємства. У системі прийнято двозмінний робочий день тривалістю з 8:00 до 20:00 (12 робочих годин). Якщо тривалість операції перевищує час, що залишився до кінця робочого дня, виконання операції автоматично переноситься на початок наступного робочого дня – нічний час до виробничого розкладу не включається. Операції в межах одного замовлення планують послідовно: кожна наступна операція маршруту починається після завершення попередньої, що відповідає технологічній логіці виготовлення.

Блок-схему алгоритму автоматичного планування наведено на рисунку 2.3.

Окрім первинного планування, алгоритм підтримує перепланування замовлення. У разі зміни кількості виробів або планової дати початку система автоматично видаляє раніше згенеровані, але ще не запущені у виробництво завдання, та формує новий розклад. Завдання, що вже виконуються або завершені, при цьому зберігаються, що забезпечує коректність обліку фактично виконаної роботи.

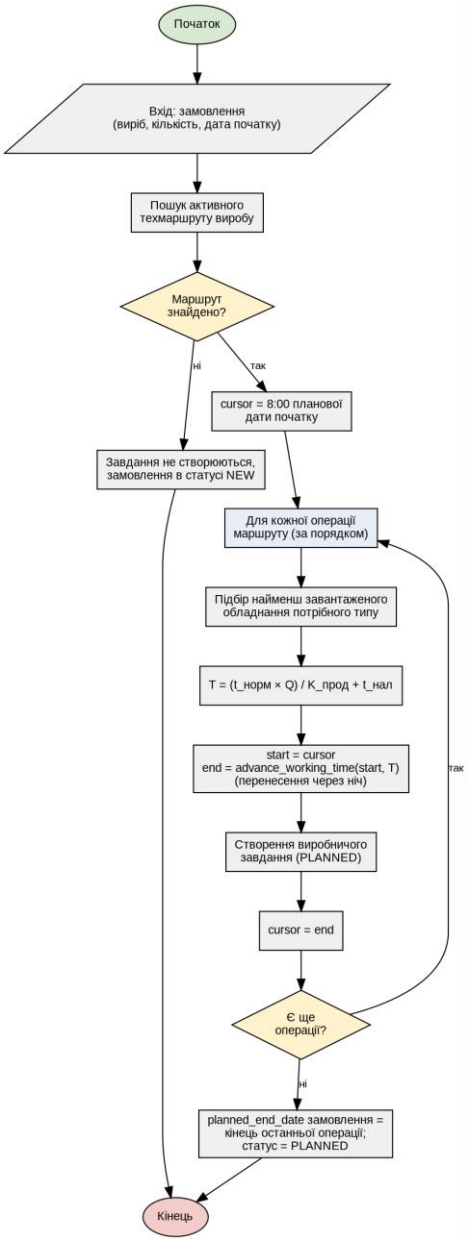


Рисунок 2.3 – Блок-схема алгоритму автоматичного планування

Таким чином, розроблений алгоритм повністю автоматизує процес складання виробничого розкладу. Користувачу достатньо вказати виріб та кількість – решту роботи (розбиття на операції, призначення обладнання, розрахунок строків) система виконує самостійно. Це усуває помилки ручного планування, прискорює реакцію на нові замовлення та підвищує прозорість виробничого процесу.

2.4 Проєктування інтерфейсу користувача

Інтерфейс користувача спроєктовано з урахуванням функціональних обов’язків кожної з чотирьох ролей системи. Основним принципом проєктування обрано мінімізацію кількості дій, потрібних для виконання типових операцій, що відповідає концепції Lean production стосовно усунення втрат часу.

Для плановика ключовим сценарієм є створення замовлення. Спроєктований сценарій складається лише з чотирьох кроків: відкриття форми, вибір виробу зі списку, введення кількості та дати початку, підтвердження. Уся подальша робота – формування виробничих завдань, призначення обладнання, розрахунок строків – виконується системою автоматично, і результат одразу відображається користувачу. Для порівняння, при ручному плануванні аналогічна операція потребує розрахунку кожної операції маршруту окремо.

Для оператора виробничої дільниці основним сценарієм є облік виконання завдань. Інтерфейс надає перелік завдань з фільтрацією за статусом та кнопки керування станом завдання (розпочати, призупинити, завершити), що дозволяє фіксувати хід виробництва одним натисканням. Фактичні часові мітки початку та завершення операцій проставляються системою автоматично.

Для керівництва та плановика спроєктовано дешборд – головний екран системи зі зведеними показниками стану виробництва. Показники згруповано за чотирма картками (замовлення, завдання, обладнання, середня завантаженість), доповнено діаграмою завантаження обладнання та блоком попереджень про завдання, що відстають від графіка. Така композиція дозволяє оцінити стан виробництва за кілька секунд без переходів між екранами.

Схему навігації між екранами системи побудовано за принципом плоскої структури: всі основні розділи (дешборд, замовлення, виробничі завдання, обладнання) доступні з бічної навігаційної панелі за одне натискання. Деталізація замовлення відкривається з реєстру замовлень і містить автоматично згенерований перелік виробничих завдань.

Кольорове кодування статусів уніфіковано в усій системі: нейтральний колір для нових об’єктів, блакитний для запланованих, синій для тих, що виконуються, жовтий для призупинених, зелений для завершених. Аналогічне кодування застосовано до індикаторів завантаження обладнання: зелений відповідає нормальному завантаженню, синій – помірному, червоний – перевантаженню понад 80 відсотків.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

У цьому розділі описано програмну реалізацію автоматизованої системи планування та обліку виробничих операцій відповідно до проєктних рішень, наведених у розділі 2. Розглянуто реалізацію серверної частини на FastAPI, ключових алгоритмів автоматичного планування та аналітики, клієнтської частини на React, а також процес розгортання й тестування системи.

3.1 Реалізація серверної частини

Серверну частину системи реалізовано мовою Python з використанням фреймворку FastAPI [8]. Вибір FastAPI зумовлений його продуктивністю, вбудованою підтримкою асинхронності, автоматичною генерацією документації REST API та інтеграцією з бібліотекою валідації даних Pydantic. Для роботи з базою даних використано ORM-бібліотеку SQLAlchemy [9], що дозволяє описувати структуру таблиць у вигляді класів Python та виконувати запити до БД без прямого написання SQL-коду.

Структуру серверної частини організовано за модульним принципом. Основні складові проєкту:

– models.py – опис моделей даних (таблиць) засобами SQLAlchemy;

– schemas.py – схеми валідації та серіалізації даних на основі Pydantic;

– auth.py – реалізація автентифікації та авторизації через JWT;

– services/ – сервіси з бізнес-логікою (планувальник, аналітика);

– routers/ – обробники HTTP-запитів, згруповані за модулями;

– main.py – точка входу застосунку, реєстрація маршрутів.

Кожна таблиця бази даних описана у вигляді класу-моделі SQLAlchemy. Наприклад, модель виробничого завдання визначає всі поля таблиці

production_tasks та зв’язки з іншими таблицями. Фрагмент опису моделі наведено на рисунку 3.1.

```
class ProductionTask(Base):
__tablename__ = "production_tasks"

id = Column(Integer, primary_key=True, index=True)
order_id = Column(Integer, ForeignKey("orders.id"))
operation_name = Column(String(200), nullable=False)
sequence_number = Column(Integer, nullable=False)
equipment_id = Column(Integer, ForeignKey("equipment.id"))
planned_start = Column(DateTime, nullable=False)
planned_end = Column(DateTime, nullable=False)
quantity_planned = Column(Numeric(10, 2), nullable=False)
quantity_done = Column(Numeric(10, 2), default=0)
status = Column(SqlEnum(TaskStatus), default=TaskStatus.PLANNED)
planned_duration_min = Column(Float, default=0.0)
```

Рисунок 3.1 – Фрагмент опису моделі виробничого завдання

Взаємодію клієнта із сервером організовано через REST API. Кожен функціональний модуль має власний набір кінцевих точок. Основні кінцеві точки API наведено в таблиці 3.1.

Таблиця 3.1 – Основні кінцеві точки REST API

Метод та шлях	**Призначення**
POST /api/auth/login	Автентифікація користувача, видача JWT
GET /api/orders	Отримання переліку замовлень
POST /api/orders	Створення замовлення із автопланування̆м
POST /api/orders/{id}/replan	Перепланування замовлення
GET /api/tasks	Отримання виробничих завдань
PATCH /api/tasks/{id}	Оновлення статусу завдання
GET /api/equipment	Перелік обладнання
GET /api/analytics/dashboard	Зведені показники для дешборду

Автентифікацію реалізовано на основі стандарту JWT [12]. При успішному вході користувач отримує підписаний токен, що містить його ідентифікатор та роль. Цей токен додається до кожного наступного запиту в заголовку Authorization.

Перевірка прав доступу до захищених операцій виконується через механізм залежностей FastAPI: спеціальна функція перевіряє наявність у користувача потрібної ролі перед виконанням операції. Паролі зберігаються у вигляді bcrypt-хешів.

3.2 Реалізація автопланувальника та аналітики

Автоматичний планувальник реалізовано у вигляді окремого сервісного модуля services/scheduler.py. Головна функція plan_order приймає об'єкт замовлення та формує для нього перелік виробничих завдань згідно з алгоритмом, описаним у 2.3. Ключовий фрагмент реалізації планувальника наведено на рисунку 3.2.

```
def plan_order(db, order):
# 1. Знайти активний маршрут виробу
route = db.query(Route).filter(
    Route.product_id == order.product_id,
    Route.is_active == True).first()
if route is None or not route.operations:
    return []

qty = float(order.quantity)
cursor = datetime.combine(order.planned_start_date,
    datetime.min.time()).replace(hour=WORK_DAY_START_HOUR)

for route_op in route.operations:
    operation = route_op.operation
    # 2. Підбір обладнання за найменшою завантаженістю
    equipment = _pick_equipment(db, operation.equipment_type_id)
    perf = equipment.performance_factor if equipment else 1.0
    # 3. Розрахунок тривалості операції
    duration = (operation.norm_time_min * qty) / max(perf, 0.1) \
               + operation.setup_time_min
    # 4. Послідовні строки з урахуванням календаря
    start = cursor
    end = _advance_working_time(start, duration)
    # ... створення завдання ...
    cursor = end
```

Рисунок 3.2 – Фрагмент реалізації автопланувальника

Балансування завантаження обладнання реалізовано у функції _pick_equipment. Серед активних одиниць обладнання потрібного типу вона обирає ту, що має найменшу поточну сумарну завантаженість. Реалізацію наведено на рисунку 3.3.

```
def _pick_equipment(db, equipment_type_id):
q = db.query(Equipment).filter(
    Equipment.status == EquipmentStatus.ACTIVE)
if equipment_type_id is not None:
    q = q.filter(
        Equipment.equipment_type_id == equipment_type_id)
candidates = q.all()
if not candidates:
    return None
# обираємо за мінімальною поточною завантаженістю
return min(candidates,
    key=lambda e: _equipment_current_load(db, e.id))
```

Рисунок 3.3 – Підбір обладнання за принципом балансування завантаження

Модуль аналітики services/analytics.py реалізує автоматичний розрахунок показників ефективності виробництва [15]. Функція dashboard_summary обчислює зведені показники: кількість замовлень і завдань за статусами, стан обладнання, середню завантаженість, відсоток виконання плану та кількість завдань, що відстають від графіка. Усі показники розраховуються безпосередньо з даних бази без ручного вводу, що забезпечує їх актуальність у будь-який момент часу.

Окремо реалізовано функцію detect_overdue_tasks, яка автоматично виявляє виробничі завдання, що відстають від графіка. Завдання вважається простроченим, якщо його планова дата завершення вже минула, а воно не виконане, або якщо перевищено плановий час виконання. Ця інформація відображається на дешборді у вигляді попереджень, що дозволяє плановику оперативно реагувати на відхилення. Фрагмент реалізації наведено на рисунку 3.4.

```
def detect_overdue_tasks(db):
now = datetime.utcnow()
overdue = []
tasks = db.query(ProductionTask).filter(
    ProductionTask.status != TaskStatus.COMPLETED).all()
for t in tasks:
    if t.planned_end and t.planned_end < now:
        overdue.append({
            "task_id": t.id,
            "order_id": t.order_id,
            "operation_name": t.operation_name,
            "reason": "Планова дата завершення минула",
        })
return overdue
```

Рисунок 3.4 – Автоматичне виявлення завдань, що відстають

3.3 Реалізація клієнтської частини

Клієнтську частину системи реалізовано мовою JavaScript з використанням бібліотеки React 18 [10] та інструмента збірки Vite. Для побудови інтерфейсу використано компонентну бібліотеку Ant Design [11], що надає готові елементи промислового рівня: таблиці, форми, модальні вікна, індикатори. Для побудови діаграм використано бібліотеку Recharts, для маршрутизації між екранами – React Router, для взаємодії із сервером – HTTP-клієнт Axios.

Клієнтський застосунок побудовано за принципом односторінкового застосунку (SPA). Структуру інтерфейсу організовано навколо основного макета (Layout), що містить бічну навігаційну панель та область відображення активної сторінки. Реалізовано такі основні екрани: сторінка входу, дешборд, реєстр замовлень, деталі замовлення, перелік виробничих завдань, довідник обладнання.

Взаємодію із сервером інкапсульовано в окремому модулі-клієнті на основі Axios. До кожного запиту автоматично додається JWT-токен авторизації через механізм перехоплювачів (interceptors). Фрагмент налаштування HTTP-клієнта наведено у рисунку 3.5.

```
const api = axios.create({
baseURL: import.meta.env.VITE_API_BASE_URL || '',
 headers: { 'Content-Type': 'application/json' },
});

api.interceptors.request.use((config) => {
const token = localStorage.getItem('token');
if (token)
config.headers.Authorization = `Bearer ${token}`;
return config;
});
```

Рисунок 3.5 – Налаштування HTTP-клієнта з авторизацією

Головним екраном системи є дешборд, на якому відображаються ключові показники виробництва: кількість замовлень і завдань, стан обладнання, середня завантаженість. Показники супроводжуються діаграмою завантаження обладнання

та попередженнями про завдання, що відстають від графіка. Загальний вигляд дешборду наведено на рисунку 3.6.

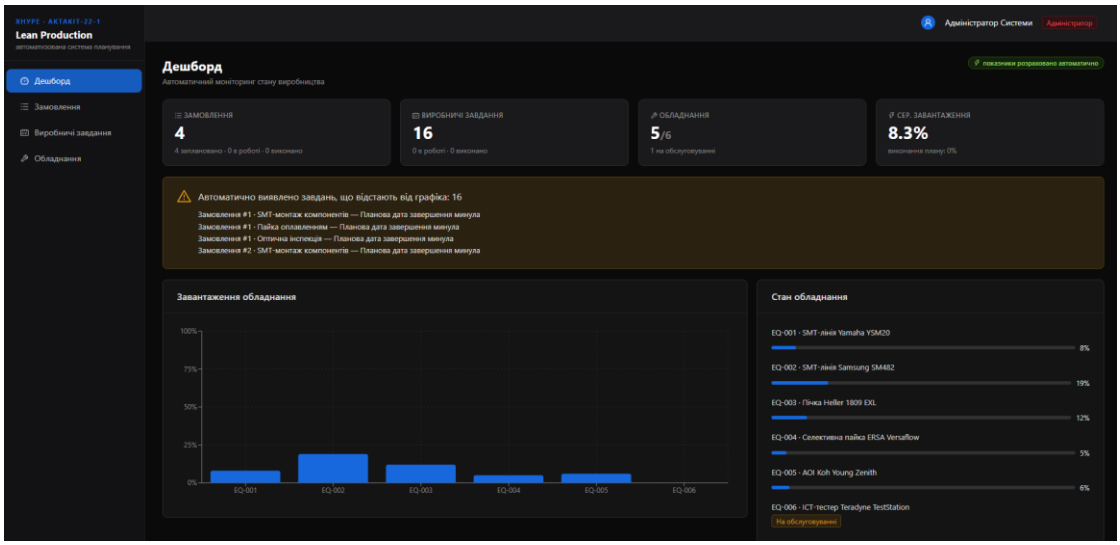


Рисунок 3.6 – Головний екран системи (дешборд)

Екран реєстру замовлень відображає перелік усіх замовлень з можливістю фільтрації за статусом. Створення нового замовлення відбувається через модальне вікно, у якому користувач вказує виріб, кількість, дату початку та пріоритет. Після підтвердження система автоматично запускає планувальник, і користувач одразу бачить згенерований перелік виробничих завдань. Вигляд екрана реєстру замовлень наведено на рисунку 3.7.

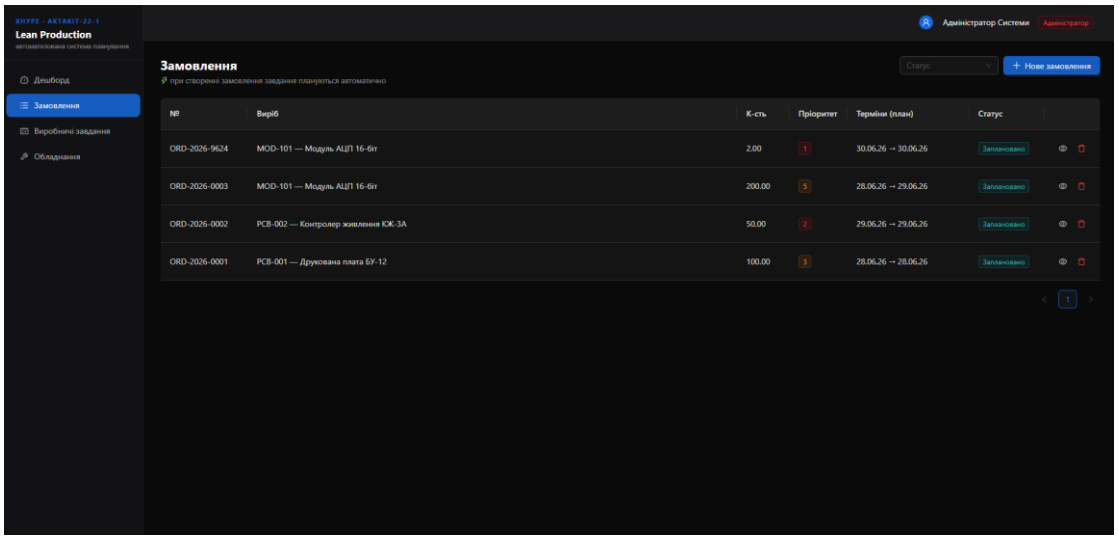


Рисунок 3.7 – Екран реєстру замовлень

Екран деталей замовлення відображає повну інформацію про замовлення та автоматично згенерований перелік виробничих завдань із зазначенням призначеного обладнання та планових строків виконання кожної операції. Вигляд екрана деталей замовлення наведено на рисунку 3.8.

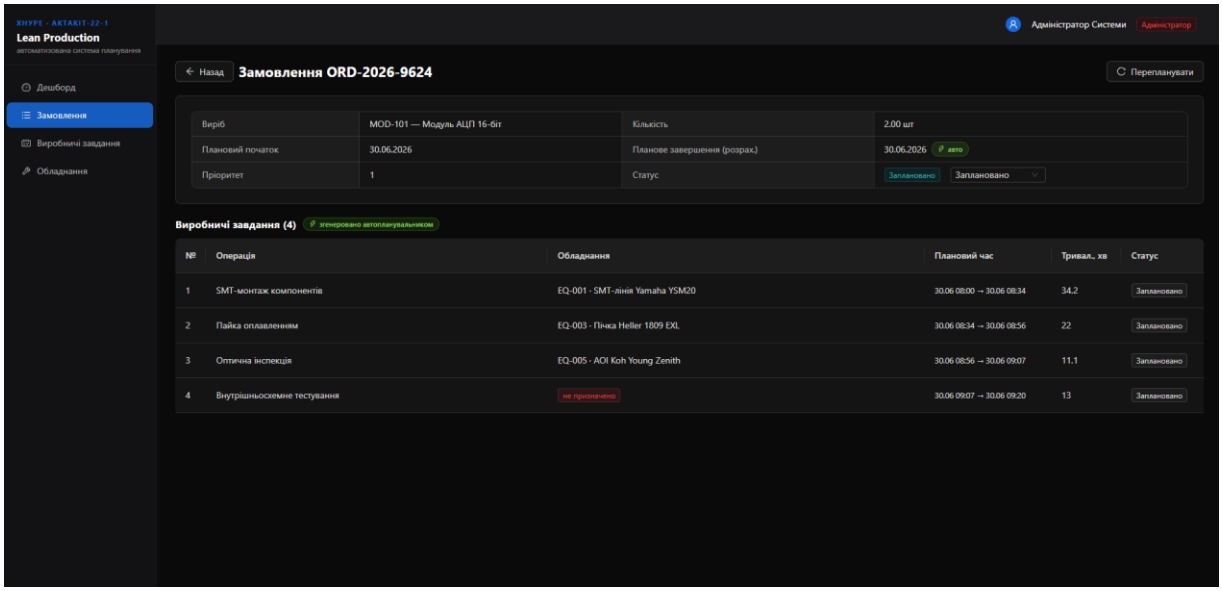


Рисунок 3.8 – Екран деталей замовлення

Екран виробничих завдань призначений для оператора виробничої дільниці. Він дозволяє переглядати завдання, фільтрувати їх за статусом та змінювати статус виконання (розпочати, призупинити, завершити). При зміні статусу автоматично фіксуються фактичні часові мітки початку та завершення операції.

3.4 Розгортання та тестування системи

Для забезпечення доступності системи її розгорнуто у хмарному середовищі. Особливістю реалізації є те, що зібрана клієнтська частина роздається тим самим сервером, що й API. Це дозволяє розгорнути всю систему як єдиний сервіс, що спрощує процес розгортання та усуває проблеми з налаштуванням міжсайтових запитів (CORS).

Розгортання виконано на платформі Render [14] – хмарному сервісі класу PaaS (Platform as a Service). Процес розгортання автоматизовано за допомогою файлу конфігурації render.yaml, який описує веб-сервіс та базу даних PostgreSQL. При розгортанні платформа автоматично збирає клієнтську частину, встановлює залежності серверної частини, створює базу даних та наповнює її початковими даними. Структуру процесу розгортання наведено на рисунку 3.9.

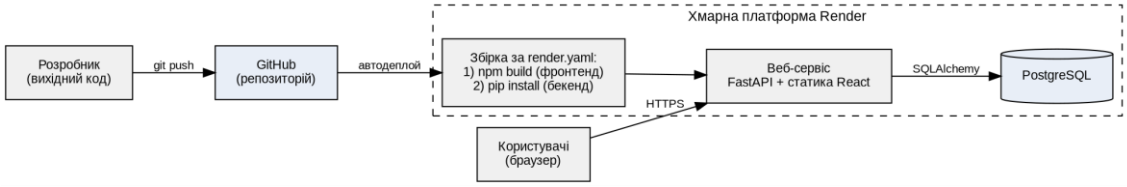


Рисунок 3.9 – Схема розгортання системи

Тестування системи проводилося на кожному етапі розробки. Для перевірки серверної частини використано вбудований механізм тестування FastAPI (TestClient), що дозволяє виконувати запити до застосунку без запуску реального веб-сервера. Перевірено такі сценарії:

– автентифікація користувачів різних ролей та перевірка розмежування прав доступу;

– створення замовлення з автоматичною генерацією виробничих завдань;

– коректність розрахунку тривалості операцій та планових строків;

– балансування завантаження обладнання при плануванні;

– зміна статусів завдань та автоматичне оновлення статусу замовлення;

– перепланування замовлення при зміні його параметрів;

– каскадне видалення завдань при видаленні замовлення.

Результати тестування підтвердили коректність роботи всіх ключових функцій системи. Зокрема, перевірено, що при створенні замовлення планувальник правильно розбиває його на операції за технологічним маршрутом, розподіляє завдання по обладнанню з балансуванням завантаження та коректно розраховує

строки з урахуванням робочого календаря. Перевірка розмежування прав доступу підтвердила, що користувач з роллю «оператор» не має можливості створювати замовлення, а неавторизований доступ до захищених ресурсів блокується.

Результати тестування основних функцій системи наведено в таблиці 3.2.

Таблиця 3.2 – Результати тестування основних функцій системи

Функція, що тестувалася	**Результат**
Автентифікація та авторизація за ролями	Пройдено
Автоматична генерація виробничих завдань	Пройдено
Розрахунок тривалості та строків операцій	Пройдено
Балансування завантаження обладнання	Пройдено
Зміна статусів завдань та замовлень	Пройдено
Перепланування замовлення	Пройдено
Каскадне видалення пов’язаних записів	Пройдено

3.5 Приклад роботи автоматичного планувальника

Для демонстрації роботи системи розглянемо наскрізний приклад автоматичного планування реального замовлення. Нехай плановик створює замовлення на виготовлення 50 одиниць виробу «Контролер живлення КЖ-3А», технологічний маршрут якого складається з п’яти операцій: SMT-монтаж компонентів, пайка оплавленням, селективна пайка рознімачів, оптична інспекція та внутрішньосхемне тестування.

Після підтвердження форми система виконала планування автоматично. Для першої операції (SMT-монтаж, норма часу 2,5 хв/од., налагодження 30 хв) планувальник обрав з двох наявних SMT-ліній найменш завантажену – лінію з коефіцієнтом продуктивності 1,2. Тривалість операції розраховано як $(2{,}5 \cdot 50) / 1{,}2 + 30 = 134{,}2$ хв. Аналогічно розраховано решту операцій маршруту з підбором обладнання для кожної.

Результати автоматичного планування замовлення наведено в таблиці 3.3.

Таблиця 3.3 – Результат автоматичного планування замовлення (50 од.)

№	Операція	Обладнання	Тривалість, хв
1	SMT-монтаж компонентів	EQ-001 (K=1,2)	134,2
2	Пайка оплавленням	EQ-003 (K=1,0)	70,0
3	Селективна пайка рознімачів	EQ-004 (K=0,9)	181,7
4	Оптична інспекція	EQ-005 (K=1,5)	36,7
5	Внутрішньосхемне тестування	EQ-006 (K=1,0)	85,0

Планові строки операцій вибудувано послідовно з урахуванням робочого календаря: операції, що не вміщуються до кінця робочого дня (20:00), автоматично перенесено на ранок наступного дня. Планову дату завершення замовлення визначено як момент завершення останньої операції маршруту. Уся процедура планування зайняла менше секунди, тоді як ручне складання аналогічного розкладу з розрахунком кожної операції потребувало б від планувальника 15-20 хвилин та могло містити арифметичні помилки.

Цей приклад демонструє ключову цінність системи: рутинна розрахункова робота повністю передана алгоритму, а людина зосереджується на змістовних рішеннях – визначенні пріоритетів замовлень та реагуванні на відхилення, які система також виявляє автоматично.

3.6 Охорона праці при експлуатації автоматизованої системи

Розроблена автоматизована система експлуатується користувачами (плановиком, технологом, оператором, адміністратором) на робочих місцях, обладнаних персональними комп’ютерами з екранними пристроями. Тому під час впровадження та експлуатації системи мають виконуватися вимоги нормативних документів з охорони праці під час роботи з екранними пристроями та вимоги безпеки праці в індустрії інформаційних технологій [16].

Робоче місце користувача системи має відповідати санітарним нормам: площа одного робочого місця з відеотерміналом повинна становити не менше ніж 6,0 м², об’єм – не менше ніж 20 м³. Екран відеотермінала розташовують на оптимальній відстані 600–700 мм від очей користувача, верхній край екрана – на рівні очей або дещо нижче. Освітленість робочої поверхні столу має становити 300–500 лк, при цьому слід уникати відблисків на екрані від джерел світла та вікон.

Вимоги електробезпеки передбачають живлення комп’ютерного обладнання від мережі із захисним заземленням (зануленням), використання справних кабелів та розеток, заборону роботи з відкритими корпусами системних блоків і самостійного ремонту обладнання під напругою. Прокладання кабелів живлення виконується так, щоб унеможливити їх механічне пошкодження на робочому місці.

Для запобігання перевтомі та зниження зорового і психоемоційного навантаження встановлюється раціональний режим праці та відпочинку: при восьмигодинній робочій зміні регламентовані перерви тривалістю 15 хвилин призначаються через кожні дві години роботи з екранним пристроєм. Під час перерв рекомендується виконувати вправи для очей та зміну робочої пози. У приміщенні підтримуються нормовані параметри мікроклімату: температура повітря 22–24 °С у холодний період року, відносна вологість 40–60 %.

Слід зазначити, що впровадження розробленої системи саме по собі покращує умови праці персоналу: автоматичне формування виробничих завдань і розрахунок показників ефективності усувають рутинні ручні обчислення, скорочують тривалість безперервної роботи плановика з екранним пристроєм та знижують психоемоційне навантаження, пов’язане з ризиком помилок ручного планування.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано актуальну задачу автоматизації планування та обліку виробничих операцій малого радіоелектронного підприємства шляхом розроблення спеціалізованої веб-системи на основі концепції Lean production. У ході виконання роботи отримано такі результати.

Проведено аналіз предметної області та існуючих систем планування й обліку виробництва класів ERP, MES та APS. Встановлено, що наявні на ринку рішення орієнтовані на великі підприємства, мають високу вартість впровадження та володіння, що робить їх недоступними для малих і середніх підприємств радіоелектронної галузі. Обґрунтовано доцільність розроблення спеціалізованої веб-системи з базовим функціоналом оперативного планування та обліку.

Обґрунтовано вибір технологічного стеку: реляційна СУБД PostgreSQL для зберігання даних, фреймворк FastAPI мовою Python для серверної частини та бібліотека React.js для клієнтської частини. Спроєктовано трирівневу архітектуру системи з розподілом на рівні представлення, бізнес-логіки та даних, а також механізм рольового розмежування доступу на основі токенів JWT для чотирьох ролей користувачів: адміністратора, плановика, технолога та оператора.

Спроєктовано структуру реляційної бази даних із десяти таблиць, нормалізовану до третьої нормальної форми. Структура охоплює нормативно-довідкову інформацію (вироби, операції, обладнання, типи обладнання), технологічні маршрути виготовлення виробів, виробничі замовлення та завдання, а також журнал дій користувачів.

Розроблено алгоритм автоматичного планування виробничих завдань, що є ключовим елементом автоматизації системи. Алгоритм за технологічним маршрутом виробу самостійно генерує виробничі завдання, розподіляє їх по обладнанню за принципом балансування завантаження, розраховує тривалість операцій за нормами часу з урахуванням продуктивності обладнання та визначає

планові строки з урахуванням робочого календаря підприємства. Завдяки цьому користувачу достатньо вказати виріб, кількість і дату початку – повний виробничий розклад формується системою автоматично.

Здійснено програмну реалізацію всіх компонентів системи: серверної частини з REST API, моделями даних та JWT-авторизацією; автоматичного планувальника та модуля аналітики з автоматичним розрахунком показників ефективності й виявленням відставань від графіка; клієнтської частини з шістьма основними екранами. Систему розгорнуто у хмарному середовищі Render як єдиний сервіс, що роздає і програмний інтерфейс, і клієнтський застосунок.

Проведено тестування системи, яке підтвердило коректність роботи всіх ключових функцій: автентифікації та розмежування прав за ролями, автоматичної генерації виробничих завдань, розрахунку тривалості та строків операцій, балансування завантаження обладнання, зміни статусів завдань з автоматичним оновленням статусу замовлення, перепланування та каскадного видалення пов’язаних записів.

Практична цінність роботи полягає у створенні готової до впровадження автоматизованої системи, що звільняє персонал малих і середніх радіоелектронних підприємств від рутинної ручної праці зі складання виробничого розкладу, зменшує кількість помилок планування та підвищує прозорість виробничого процесу відповідно до принципів Lean production.

Перспективними напрямами розвитку системи є: впровадження оптимізаційних алгоритмів планування з урахуванням пріоритетів замовлень та кількох критеріїв одночасно; інтеграція з обладнанням для автоматичного збору фактичних даних про виконання операцій; розроблення модуля управління матеріальними запасами з підтримкою специфікацій виробів; додавання прогнозної аналітики на основі накопичених історичних даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ : ДП «УкрНДНЦ», 2016. 26 с.

2. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник/ І. Ш. Невлюдов, В.А. Андрусевич, О. В. Токарєва, С. П. Новоселов, О. В. Сичова. – Харків : Видавництво Іванченка І. С., 2022. 151 с.

3.Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»освітньої програми Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипенко, О.В. Токарєва, С.П. Новоселов, О.В Сичова. Харків: ХНУРЕ, 2023. 64 с.

4. Womack J. P., Jones D. T. Lean Thinking: Banish Waste and Create Wealth in Your Corporation. New York : Free Press, 2003. 396 p.

5. Kletti J. Manufacturing Execution System - MES. Berlin : Springer, 2007. 272 p.

6. Manufacturing Execution System Market Size Report, 2024-2030. Grand View Research. URL: https://www.grandviewresearch.com/industry-analysis/manufacturing-execution-systems-mes-market (дата звернення: 15.05.2026).

7. PostgreSQL 16 Documentation. PostgreSQL Global Development Group. URL: https://www.postgresql.org/docs/16/ (дата звернення: 15.05.2026).

8. FastAPI Documentation. URL: https://fastapi.tiangolo.com/ (дата звернення: 15.05.2026).

9. SQLAlchemy 2.0 Documentation. URL: https://docs.sqlalchemy.org/en/20/ (дата звернення: 15.05.2026).

10. React Documentation. Meta Platforms, Inc. URL: https://react.dev/ (дата звернення: 15.05.2026).

11. Ant Design - The world’s second most popular React UI framework. URL: https://ant.design/ (дата звернення: 15.05.2026).

12. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT) : RFC 7519. Internet Engineering Task Force, 2015. URL: https://www.rfc-editor.org/rfc/rfc7519 (дата звернення: 15.05.2026).

13. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures : PhD dissertation. University of California, Irvine, 2000. 162 p.

14. Render Documentation. Render Services, Inc. URL: https://render.com/docs (дата звернення: 15.05.2026).

15. Провост Н., Фосетт Т. Data Science для бізнесу : пер. з англ. Київ : Наш формат, 2019. 400 с.

16. Комплекс навчально-методичного забезпечення навчальної дисципліни «Безпека праці в індустрії ІТ-технологій» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [http://catalogue.nure.ua/knmz] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. - Харків, 2017. - 122 с.