Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект системи автоматизації виробництва компанії Lomany з модулем складання плану роботи»

Виконав студент 4-го курсу, групи КІ-3
спеціальності 123 «Комп'ютерна інженерія»
_____ А.Р. Опанасюк
Керівник к.т.н., ст. викладач кафедри ІПЗ
_____ А.Ю. Левченко
Рецензент к.т.н., доцент кафедри КІ та КБ
_____ О.В. Пірог

Житомир – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
(назва прописними)

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
(назва прописними)

СПЕЦІАЛЬНІСТЬ 123 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»
(назва прописними)

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерної інженерії та
кібербезпеки
к.т.н. А.А. Єфіменко
«12» 04 2021 р.

ЗАВДАННЯ
на випускну кваліфікаційну роботу

Студента Опанасюка Артура Руслановича

Тема роботи: «Проект системи автоматизації виробництва компанії Lomany з модулем складання плану роботи»
Затверджена Наказом університету від «12» квітня 2021 р. № 133/с
Термін здачі студентом закінченої роботи 01 червня 2021 р
Вихідні дані роботи (зазначається предмет і об'єкт дослідження)
Предметом дослідження є використання веб орієнтованих технологій для реалізації проекту веб-додатку, системи автоматизації виробничих процесів компанії Lomany. Об'єктом дослідження є процес управління та функціонування закладу.

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Левківський В. Л.	1.03.2021	1.03.2021
2	Левківський В. Л.	5.04.2021	5.04.2021
3	Левківський В. Л.	3.05.2021	3.05.2021

Керівник
(підпис)

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання. Пошук, огляд та аналіз аналогічних розробок. Опрацювання літературних джерел	1 березня 2021 – 11 квітня 2021	Виконано
2	Проектування структури системи	12 квітня 2021 – 16 квітня 2021	Виконано
3	Написання програмного коду	17 квітня 2021 – 12 травня 2021	Виконано
4	Тестування	13 травня 2021 – 25 травня 2021	Виконано
5	Написання пояснювальної записки	26 травня 2021 – 5 червня 2021	Виконано
6	Підготовка презентації та виступу	6 червня – 12 червня 2021	Виконано
7	Отримання відгуку керівника та рецензії на квіліфікаційну роботу	13 червня 2021	Виконано
8	Захист	25 червня 2021	Виконано

Студент _____
(підпис)

Керівник _____
(підпис)

РЕФЕРАТ

Випускна робота бакалавра складається з програмного комплексу «Система автоматизації компанії Lomany з модулем складання плану роботи» та пояснювальної записки. Пояснювальна записка до випускної роботи містить 58 сторінок, 37 ілюстрацій та 11 таблиць.

Метою роботи є розробка системи, що автоматизує процеси роботи компанії, що займається виробництвом панчішних виробів. Програма автоматизує процес складання плану робіт на період та складання графіку робіт працівників.

В роботі визначено основні завдання на розробку системи, проаналізовано аналоги розробленої системи. Наведено сценарії роботи програмного комплексу, загальна структура програмного комплексу, опис алгоритмів взаємодії окремих модулів системи, опис об'єктної структури системи (діаграма класів). Також наводиться результати тестування системи.

КЛЮЧОВІ СЛОВА: ПЛАН ВИРОБНИЦТВА, ГРАФІК РОБОТИ, HMVC, БАЗА ДАНИХ.

ABSTRACT

The bachelor's thesis consists of the software package "Automation system of Lomany with work plan module" and an explanatory note. The explanatory note to the final work contains 58 pages, 37 illustrations and 11 tables.

The purpose of the work is to develop a system that automates the work processes of a company engaged in the production of hosiery. The program automates the process of drawing up a work plan for the period and scheduling the work of employees.

The main tasks for system development are defined in the work, analogues of the developed system are analyzed. The scenarios of the software package, the general structure of the software package, the description of the algorithms of interaction of individual modules of the system, the description of the object structure of the system (class diagram) are given. The results of system testing are also given.

KEY WORDS: PRODUCTION PLAN, WORK SCHEDULE, HMVC, DATABASE.

ЗМІСТ

РЕФЕРАТ ..... **Ошибка! Закладка не определена.**
ABSTRACT 5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ..... 8
ВСТУП 9
РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ ДОСЛІДЖЕНЬ ПО АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ВИРОБНИЦТВА 11
1.1 Обґрунтування актуальності теми досліджень по розробці системи автоматизації процесів виробництва 11
1.2 Аналіз аналогів програмного продукту..... 12
1.3 Аналіз основних вимог до додатку системи автоматизації процесів виробництва..... 15
1.4 Постановка завдання 17
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ КОМПАНІЇ LOMANY 20
2.1 Розробка загальної структури системи..... 20
2.2 Проектування та реалізація бази даних системи..... 24
2.3. Вибір архітектури системи 29
2. 4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення..... 31
2.5 Реалізація додатку системи автоматизації компанії 35
РОЗДІЛ 3. ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ КОМПАНІЇ LOMANY 40
3.1 Порядок встановлення та налаштування системи..... 40
3.2 Розробка рекомендацій щодо застосування програмного забезпечення системи автоматизації виробничих процесів компанії Lomany 42
3.3 Дослідження працездатності програмного забезпечення системи автоматизації виробничих процесів компанії Lomany 53

Висновки до третього розділу 56
ВИСНОВКИ 57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 59
ДОДАТКИ 61
ДОДАТОК А 62
ДОДАТОК Б 65
ДОДАТОК В 70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – Бази даних.

MVC – Model-view-controller. Моде́ль–вигляд–контро́лер

UML – Unified Modeling Language. уніфікована мова моделювання

ПВ – План виробництва

HMVC - Hierarchical model–view–controller — Ієрархична Модель-Вид-Контролер

АСУ – Автоматизована система управління

ЛМІ – Людино-машинний інтерфейс

ВСТУП

З розвитком економічного та науково-технічного прогресу у XXI ст. виник новий четвертинний сектор економіки, пов'язаний з інформаційними технологіями, що активно впливає на інші суміжні сектори. Поява цього сектора цілком логічна, адже сьогодні інформація є стратегічним ресурсом та капіталом, а інформаційні технології – інструментом вдосконалення та підвищення ефективності діяльності підприємств будь-якої сфери господарювання. Серед широкого спектра IT, які сьогодні активно використовуються в комерції, важливе місце займають автоматизовані системи управління, ефективність яких пояснюється комплексністю та системністю автоматизації усіх бізнес-процесів на підприємстві. Не є винятком і підприємства, що займаються виробництвом продукції. Кінцевою метою комплексної автоматизації є створення інформаційної бази, яка б сприяла прийняттю управлінських рішень, вибору стратегії на основі аналізу господарської діяльності за різними показниками та досягненню конкурентноздатності підприємства.

Автоматизація підприємств - уже давно не питання: усі розуміють її необхідність і окупність. Головним питанням залишається тільки вибір програмного забезпечення для відповідного типу закладу.

Компанія Lomany займається виробництвом панчішної продукції. Потреба в простій і доступній як за ціною, так і по впровадженню і використанню програмі була давно. Для задоволення цієї потреби було вирішено розробити програму, що представляється в даній випускній роботі. Дана програма готова запропонувати власникам малого і середнього бізнесу рішення для оптимізації управління та автоматизації процесу побудови плану роботи.

Метою випускної роботи є проектування та розробка модулів Веб додатку, системи автоматизації процесів виробництва.

Встановлена мета обумовлює наступні завдання:

– проведення аналізу задач автоматизації закладу;
– визначення архітектури та узагальненої структури системи;
– обґрунтування та вибір засобів реалізації системи;

– розробка інформаційного та математичного забезпечення системи;

– проектування структурних складових та алгоритмів роботи системи;

– реалізація програмного комплексу Веб додатку, системи автоматизації виробничих процесів компанії Lomany.

Об’єктом дослідження є процес управління та функціонування закладу.

Предметом дослідження є використання веб орієнтованих технологій для реалізації проекту Веб додатку, системи автоматизації виробничих процесів компанії Lomany.

В роботі було використано методи об’єктно-орієнтованого проектування, об’єктно-орієнтованого програмування та дизайну інтерфейсів.

Розроблена система може бути використана для малого та середнього бізнесу.

Пояснювальна записка до випускної роботи містить 58 сторінок, 37 ілюстрацій та 11 таблиць, список літературних джерел містить 12 найменувань.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ ДОСЛІДЖЕНЬ ПО АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ВИРОБНИЦТВА

1.1 Обґрунтування актуальності теми досліджень по розробці системи автоматизації процесів виробництва

Проблему використання ІТ у бізнесі висвітлено у наукових розробках вітчизняних та зарубіжних вчених. Так, питання впровадження інформаційних систем та технологій у процеси управління вивчали А. Гринберг, Н. Горбачев, А. Бондаренко, О. Губанова, Т. Новгородцев, С. Чупров, В. Гужва, Л. Дубчак, В. Косарев [1–5]. Роль програмного забезпечення в управлінні сучасними підприємствами досліджують Ж. Пророчук, А. Журавлёва [6], доцільність застосування інформаційних технологій у бізнесі – С. Степова, А. Когут [7]. Автори О. Оліфіров, А. Лутай та К. Федусова вивчали проблему використання сучасних ІТ у бізнесі [8; 9;10]. Крім цього, науковці К. Єсіпова [11], Д. Еймор [12] окремі дослідження присвятили питанню використання інтернет-технологій у комерції, у тому числі в господарстві. Проте наразі проблема автоматизації виробничих процесів шляхом комплексного впровадження АСУ, незважаючи на її актуальність, в не достатньому полі зору вітчизняних науковців та вчених.

Щоб задовольнити всі вимоги виробництва треба організувати ведення виробничого, оперативного, складського та регламентованого обліку в єдиному інформаційному просторі. Крім того, для ефективного управління компанією необхідно планувати продажі, виробничі програми і завантаження потужностей, оперативно складати і підтримувати в актуальному стані платіжний календар, завдяки чому знижуються ризики касових розривів. Маючи під рукою, актуальну інформацію про поточної діяльності компанії, з'являється можливість більш гнучкого оперативного управління діяльністю, планування різних показників оцінки роботи, а також побудови адекватних і більш точних планів на перспективу.

Основною метою провадження Веб додатку, системи автоматизації процесів виробництва є: оптимізація процесів виробництва, автоматизоване

планування, складання графіку роботи робочих та контроль над виробничими процесами.

1.2 Аналіз аналогів програмного продукту

Управління виробничими операціями - це набір систем для управління наскрізними виробничими процесами з метою оптимізації ефективності.

Існує багато типів програмного забезпечення управління виробничими операціями, включаючи управління виробництвом, аналіз продуктивності, якість і відповідність вимогам, а також інтерфейс людина-машина (ЛМІ). Програмне забезпечення для управління виробництвом в режимі реального часу надає інформацію про робочі місця і замовленнях, робочій силі і матеріалах, стан устаткування і відвантаження продукції. Програмне забезпечення для аналізу продуктивності відображає метрики на рівні машини, лінії, заводу і підприємства для ситуаційного або історичного аналізу. Програмне забезпечення якості та відповідності використовується для забезпечення відповідності стандартам і специфікаціям для операційних процесів і процедур. Програмне забезпечення ЛМІ - це форма програмного забезпечення для управління виробничими операціями, яке дозволяє операторам керувати промисловими і технологічними механізмами управління за допомогою комп'ютерного інтерфейсу.

Для аналізу програмного забезпечення було розглянуто наступні системи автоматизації:

1. CRM DeLight - власна розробка ТОВ Авангард Консалтинг - програмний комплекс, що дає можливість комплексно автоматизувати виробництво будь-якого мірного товару.

У програмі зберігається інформація про всі контакти - план робіт на період, які проводяться роботи, нагадування на конкретну дату і час.

Програма DeLight для автоматизації виробництва дозволяє використовувати додаткові набори полів, а не вписувати всю інформацію, якої бракує в одне поле «Коментар».

Будь-який товар можна уявити як набір комплектуючих, склад й властивості яких залежать від параметрів готового виробу. Це, в свою чергу, дозволить точно розрахувати собівартість готового виробу.

Гнучка модульна структура CRM Delight дозволить Вам отримати тільки той функціонал, який необхідний. Найпростіший (базовий) варіант системи дозволяє вести простий облік замовлень, але при необхідності система може бути розширена наступними модулями і функціональністю: комплекти, послуги, модуль оптимізації, контакт-менеджер, фінансовий модуль, модуль складського обліку, генератор звітів.

2. Програма «Універсальна Система Обліку» для автоматизації обліку виробництва.

Програма «Універсальна Система Обліку» можуть користуватись:

Виробнича компанія і виробнича фірма;

Виробниче підприємство;

Виробничо-комерційне підприємство;

Торгово-виробнича компанія, фірма;

Торгово-промислова компанія;

Торгова фірма;

Торгова компанія;

Операційна система: Windows

Мова базової Версії програми: українська, російська.

Основні можливості програми:

– У вас сформується єдина база клієнтів з усіма необхідними контактними даними;
– Ви можете виготовляти будь-яку продукцію, використовуючи при цьому будь-які матеріали або напівфабрикати;
– По кожній продукції можна скласти калькуляцію, щоб програма автоматично розраховувала кількість необхідної сировини;
– Є можливість подивитися собівартість кожної продукції, що випускається;

– Відділ постачання може оприбутковувати сировину, передавати у виробничий відділ і списувати;
– Програма може працювати з будь-якою кількістю складів і підрозділів;
– Всі замовлення будуть зберігатися в базі даних, за їх кількістю можна буде планувати обсяг виробництва;
– Замовлення можна компонувати за маршрутами прямування, які будуть друкуватися водіям;
– Кожен день можна оприбутковувати вироблену продукцію на склад готової продукції, порівнюючи плановане споживання сировини з фактичним;
– Програма веде реєстрацію будь-яких платежів;
– Договори, бланки, специфікації замовлень і багато іншого можуть формуватися автоматично;
– До кожного замовлення можна прикріплювати будь-які файли, наприклад, з макетом замовлення;
– Кожну послугу можна розділити на певні роботи і етапи і легко їх контролювати;
– Ви зможете відслідковувати виконання робіт по вашим співробітникам і контролювати їх ефективність;
– Всі відділи будуть взаємодіяти між собою як єдиний успішний механізм;
– Система проводить нарахування відрядної зарплати співробітникам залежно від їх вироблення;
– Програма вміє по середнім виробничими показниками показувати, на скільки днів безперервної роботи вистачить наявної на складі сировини.

3. РосБізнесСофт - це комплексна система автоматизації бізнес-процесів для малого і середнього бізнесу. У пакет входить: CRM, ERP, а також Help Desk. Рішення підходить для вже "сталих на ноги" компаній, які зацікавлені в глибокому впровадження своїх бізнес-процесів і серйозно націлені на масштабуванні свого бізнесу. Серед клієнтів компанії найбільший інтернет-

магазин «Озон.ру», виробник миючої техніки Karcher, онлайн-бухгалтерія «Моя справа», італійський виробник котлів «Immergas» та інші відомі бренди.

Основні можливості РосБізнесСофт:

Єдина платформа для CRM, управління продажами, складським обліком, послугами

Управління виробництвом (ERP), закупівлями, робота з постачальниками

Автоматизація сервісного центру, Help Desk, тікетна система

Відкрита платформа для самостійного доопрацювання системи

Гнучке налаштування прав доступу, хронологія всіх змін в системі

Потужна система аналітики (більше 40 звітів вже в базовій поставці)

Повна двостороння інтеграція з усіма продуктами 1С по «однієї кнопки» (у інших такого не зустрічали)

Маркетплейс з готовими інтеграціями + наявність вбудованого API-інтерфейсу для створення будь-яких інтеграцій

Більше 20 галузевих рішень.

Всі розглянуті системи не є безкоштовними. Вартість від 1800 грн за модуль до 100000 грн. Також всі системи це готові рішення з базовим функціоналом. Для автоматизації нестандартних рішень та функцій виникає необхідність в розробці власної системи автоматизації.

1.3 Аналіз основних вимог до додатку системи автоматизації процесів виробництва

В випускній роботі необхідно розробити веб додаток, системи автоматизації процесів виробництва. Для реалізації поставленого завдання, основними етапами є:

Результатом реалізації поставленого завдання є веб-додаток, системи автоматизації процесів виробництва.

1. Розробити дизайн користувацького інтерфейсу.
   1.1. Створити загальну концепцію дизайну користувацького інтерфейсу.

1.2. За допомогою графічного редактора Adobe Photoshop створити графічні зображення основних елементів керування (кнопок, панелей та ін.)

1.3. Зверстати за стандартами GUI основні сторінки веб-додатку:

1.3.1. Сторінку авторизації

1.3.2. Сторінку складання плану виробництва

1.3.3. Сторінку роботи з персоналом

1.3.4. Сторінку профілю робітника (графік роботи робітника, сторінка фіксації виконаної роботи, сторінка перегляду виконаної роботи)

1.3.5. Сторінку складання графіку роботи персоналу

1.3.6. Сторінку роботи із в’язальними машинами

1.3.7. Сторінку роботи із продукцією

1.3.8. Сторінку роботи окремого робітника

2. Проектування та розробка основних модулів

2.1. Модуль роботи із продукцією

2.2. Модуль роботи із пряжею

2.3. Модуль складання графіку роботи персоналу

2.4. Модуль планування виробництва

2.5. Модуль роботи із персоналом

2.6. Модуль контролю над виробництвом

2.7. Модуль авторизації

2.8. Модуль фіксації виконаної роботи робітником

2.9. Модуль нагадування паролю

2.10. Модуль друку звітів

2.11. Модуль формування звітів в excel

2.12. Модуль збереження даних, щодо роботи робітника

Результатом реалізації поставленого завдання є веб додаток, системи автоматизації процесів виробництва.

1.4 Постановка завдання

Повне найменування розроблюваної програмної системи: «Система автоматизації компанії Lomany з модулем складання плану роботи» (надалі «програма»). Коротка назва програмної системи – «Lomany».

Програма «Lomany» призначена для збереження, обробки та відображення інформації для автоматизації роботи компанії з виробництва панчішних виробів. В програмі передбачено можливість побудови плану роботи виробництва орієнтованого на окремі в'язальні машини та відповідну продукцію. Додатковою функціональною можливістю є складання графіку роботи персоналу та облік робочого часу.

Система може використовуватись організаціями, сфера діяльності яких пов'язана з виробництвом.

Розроблена програмна система «Система автоматизації компанії Lomany з модулем складання плану роботи» дозволить підвищити ефективність, швидкість та якість роботи працівників компанії. Програма значно спростить процес планування виробництва та зробить прозорим процес обліку виконаної роботи працівниками компанії.

Програмна система повинна надавати наступні функціональні можливості:

- можливість дистанційної роботи з робочих станцій, що відключаються з локальної та глобальної мереж;
- зручний та інтуїтивно зрозумілий користувацький інтерфейс;
- оптимальне збереження даних, що забезпечить мінімальний час доступу та економію дискового простору;
- надійне збереження та можливість відновлення даних;
- модульну архітектуру, що гарантуватиме можливість легкої модернізації системи через зміну функціональних потреб користувача або оновлення програмного забезпечення.

Розробити веб орієнтований додаток для автоматизації роботи компанії Lomany, що забезпечить виконання наступних транзакцій:

1. Занесення даних про працівників компанії, посади, шаблони роботи, продукцію компанії, пряжу, постачальників, продукцію, що виробляє компанія;
2. Зберігання інформації про в'язальні машини;
3. Додавання, видалення та редагування всієї довідникової інформації з пункту 1;
4. Пошук та сортування даних;
5. Побудова плану виробництва компанії на визначений період за кожною машиною;
6. Фіксація та зберігання даних про виконану роботу працівником в розрізі план виробництва, конкретна продукція, машина, людина.
7. Облік виконаного плану виробництва;
8. Облік залишків продукції;
9. Складання графіку роботи працівників;
10. Формування звітності

Вхідними даними є інформація про робітників, номенклатура продукції, в'язальні машини, шаблони графіків роботи та пряжа, що використовується при виробництві продукції.

Введення даних повинно виконуватися за допомогою екранних форм, побудованих з використанням сучасних візуальних компонентів.

Час реакції програми на дії користувача (маніпуляції с пристроями введення даних) не повинен перевищувати 0,25 с.

Час виконання команд меню не більше 3 с.

Відображення масивів даних за запитами не більше 3 хвилин.

Операції з'єднання з базою даних не більше 1 хвилини.

Обсяг оперативної пам'яті, необхідний для роботи розробленої системи не може перевищувати 512 МБ (при низькому навантаженні).

Дисковий простір, необхідний для збереження програми і файлів бази даних не повинні перевищувати 300 МБ для робочої станції та 5 ГБ для серверу.

Розмір інсталяційного пакету програми, що міститься у складі бази даних не повинен перевищувати 100 МБ.

Висновки до першого розділу

Проведений аналіз предметної області дозволив визначити основні аспекти розробки веб додатку. Огляд наявних аналогів та виявлення протиріч між наявними можливостями та потребами показав, що головними функціями майбутньої системи має бути: зручність GUI (у зв'язку з тим, що програмою будуть користуватися робітники підприємства), обов'язковий модуль планування виробництва для майстра цеху, модуль складання графіку роботи та модуль обліку залишків продукції. Також мають бути реалізовані модулі авторизації та модуль зберігання щоденного виконання забов'язань за планом виробництва для окремих працівників.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ КОМПАНІЇ LOMANY

2.1 Розробка загальної структури системи

Веб додаток, системи автоматизації компанії Lomany створюється з метою максимально автоматизувати роботу управлінського персоналу компанії. Робота із системою поділяється на три частини: робота технічного директора, робота адміністратора та робота окремих працівників.

Вимоги користувачів.

Технічний директор має доступ до наступних функцій:

1. Авторизація в системі;
2. Складання плану виробництва на визначений період;
3. Формування звітності за планом виробництва;
4. Пошук за параметрами планів виробництва;
5. Складання графіку роботи працівників;
6. Друк графіку роботи за визначений період та за різними параметрами;
7. Оптимізація графіку роботи;
8. Слідкування за виконанням графіку роботи;
9. Розцінки за оплату праці;
10. Перегляд звітів, щодо виконання робіт робітниками;
11. Перегляд звітів, щодо виконання плану.

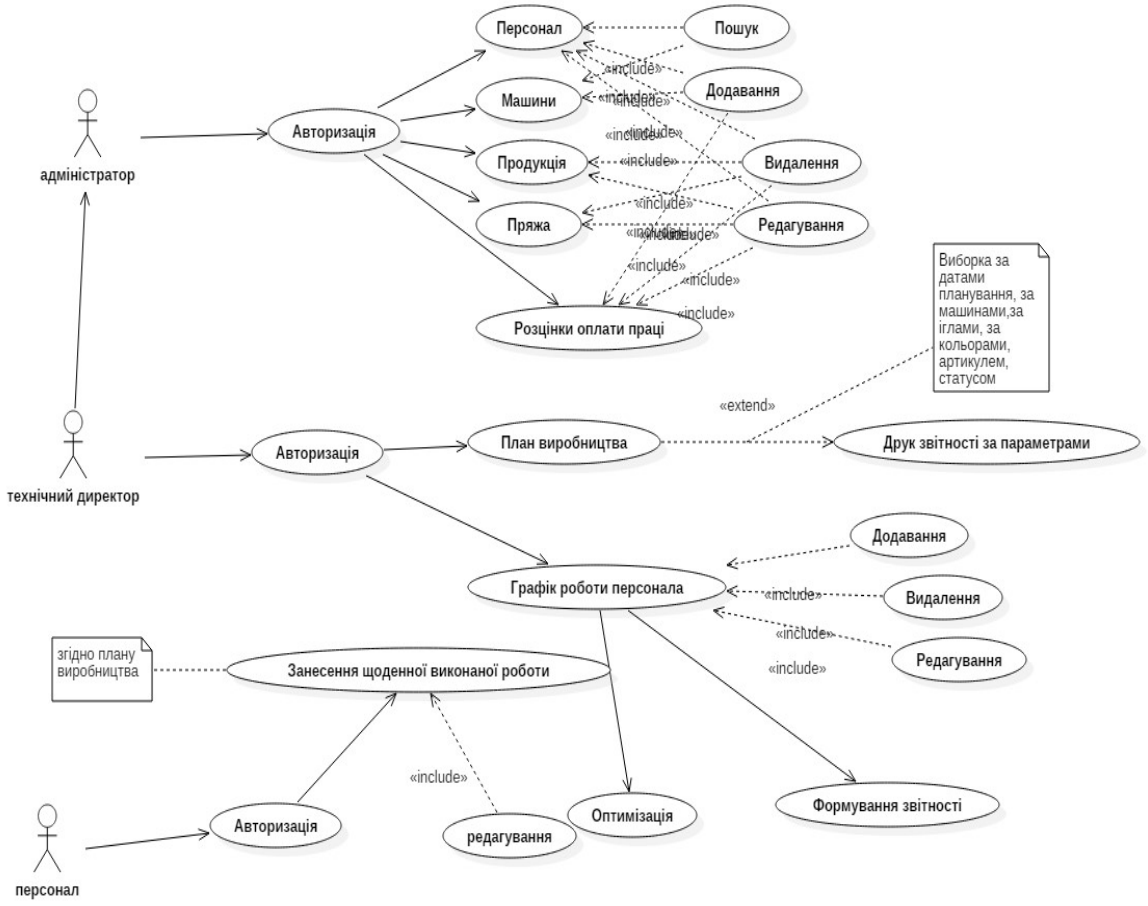


Рис. 2.1. Діаграма варіантів використання

Вимоги адміністратора:

1. Авторизація за логіном та паролем.
2. Робота із довідниками (машини, робітники, постачальники, пряжа, шаблони графіків роботи, асортимент продукції);
3. Друк інформації.

Вимоги робітника компанії:

1. Авторизація за логіном та паролем.
2. Зберігання інформації, щодо щоденно виконаної роботи згідно графіку виробництва;
3. Перегляд власного графіку виробництва
4. Перегляд та фільтрація виконаної роботи
5. Друк інформації.

Нефункціональні вимоги:

1. Сприйняття:

– Час, потрібний для навчання роботи з програмою для звичайних користувачів — 30 хвилин, для досвідчених — 10 хвилин.

– Час відповіді системи для звичайних запитів не повен перевищувати 0.016 секунди, а для більш складних запитів — 0.03 секунди.

– Інтерфейс програми повинен бути інтуїтивно зручним для користувача та не вимагати від нього додаткової підготовки.

– Надійність:

– Доступність - час, потрібний для обслуговування системи не повинен перевищувати 10 % від загального часу роботи.

2. Продуктивність. Система повинна підтримувати безперебійну роботу користувача.

Аналіз функціональних вимог дозволив виділити наступні сутності, що забезпечать реалізацію програмного комплексу системи.

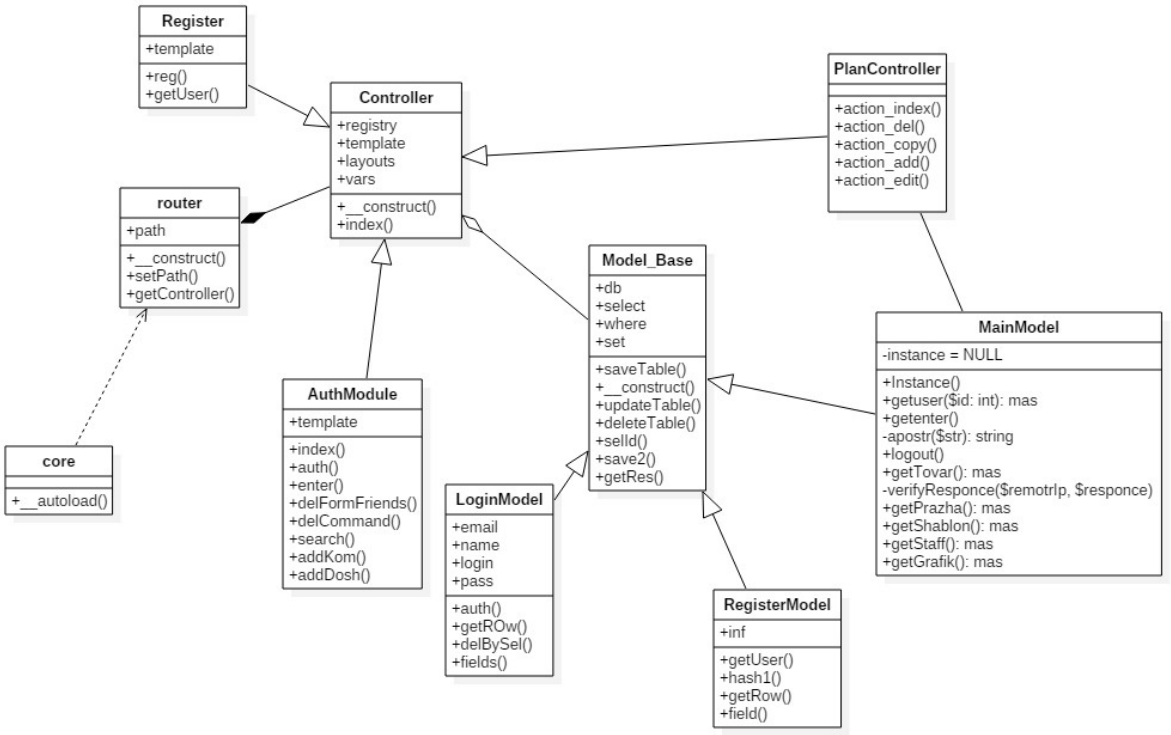


Рис. 2.2. Діаграма класів системи

На рис.2.2. наведено діаграму класів, які використовуються для керування додатком системою автоматизації компанії Lomany.

В даній діаграмі можна виділити класи, які спроектовано, використовуючи патерн проектування HMVC.

Так клас router забезпечує роутинг в системі. Роутинг - це маршрутизація: вхідний URL розбирається спеціальним чином і по його результату виконується певний код. З роутингом прямо пов'язане поняття ЛЗУ (Людинозрозумілий урли), яке дозволяє виключити в адресах складні параметри. Наприклад замість http: // сайт / admin / new-page ми мали б використовувати http: //сайт/admin.php? Action = new-page.

Будь-який URL на сервері розбирається за єдиним стандартом. в адресі передається параметр path (шлях на сервері), якого на сервері реально може не бути. Наприклад в адресі http: // сайт / admin каталога admin реально може не існувати. Тобто сервер, отримавши таку адресу, спробує знайти каталог admin, але не знайшовши його, видає 404-сторінку (not found).

Щоб виключити такий варіант, сервера вказується, що для всіх неіснуючих каталогів і файлів, підключати php-файл (зазвичай index.php).

Частина коду класу router наведена нижче.

```
class Route{
    private $components=[];
    private $controller_name=NULL, $action_name=NULL,$params=[];
    const PARAM_PATTERN="/\<([a-z0-9]+)\>\??/i";
    const PARAM_TMP_PATTERN="/\<([a-z0-9]+)\>\?/i";
    public function __construct($route,array $params)
    {
       $this->components = explode("/",$route);
   $this->controller_name=@$params["controller"];
        $this->action_name=@$params["action"];
        $this->params=$params;
          }
          private function isTempParam($name){
        return preg_match(self::PARAM_TMP_PATTERN,$name);
          }
          private function isParam($param,$value){
        if(!preg_match(self::PARAM_PATTERN,$param,$arr)) return false;
        $name=$arr[1];

        if($name=="controller"){
            $this->controller_name = $value;
        }
        else if($name=="action"){
            $this->action_name = $value;
        }else{
            $this->params[$name]=$value;
        }
        return true;
}
          public function exec($realrouteparams){
```

```
   $count=max(count($realrouteparams),count($this->components));
                for($i=0;$i<$count;$i++){
                    if(!$this->components[$i]) return false;
                    if(empty($realrouteparams[$i]) && $this-
>isTempParam($this->components[$i])) return true;
                    elseif(empty($realrouteparams[$i])) return false;
                    if($this->isParam($this-
>components[$i],$realrouteparams[$i]))continue;
                       if($this-
>components[$i]!=$realrouteparams[$i])return false;
        }
        return true;
           }
           public function getData(){

                if($this->controller_name===NULL || $this-
>action_name===NULL) throw new RoutExeption("Invalid Route",404);
                return [
                    "controller"=>$this->controller_name,
                    "action"=>$this->action_name,
                    "params"=>$this->params
                ];
           }
}
```

Клас Template – відповідає за підключення базового шаблону. Спеціалізовані контролери у зв’язку із відповідними класами моделей відповідають за окремі функціональності. Так клас PlanController – відповідає за модуль роботи із планом виробництва продукції. Даний клас має наступні action: action_index() – дія по замовчуванню; action_del() – видаляє створений раніше план виробництва; action_add() – додає в базу даних інформацію про план виробництва. Action_edit() – редагує існуючу інформацію.

2.2 Проектування та реалізація бази даних системи

Реалізована база даних MySQL, що складається з 16 таблиць, які містять у собі всі дані для роботи програми. База даних має назву Nosok.

Таблиця 2.1

Таблиця таблиць бази даних

Назва таблиці	Призначення
posada	Довідник посад
shablon	Шаблон графіку роботи
grafik	Реальний графік роботи робітників
plan	Звіт виробництва
const	Костанти роботи
planColor	План за кольорами
catTov	Категорія продукції

Продовження таблиці 2.1

deti	Діти робітників
Db_user	Користувачі програми
tovar	Номенклатура товару
nameStolb	Назви стовбчиків для плану виробництва
sezon	Сезон
staff	Персонал
prazha	Пряжа
mash	Машини

Розглянемо структуру бази даних.

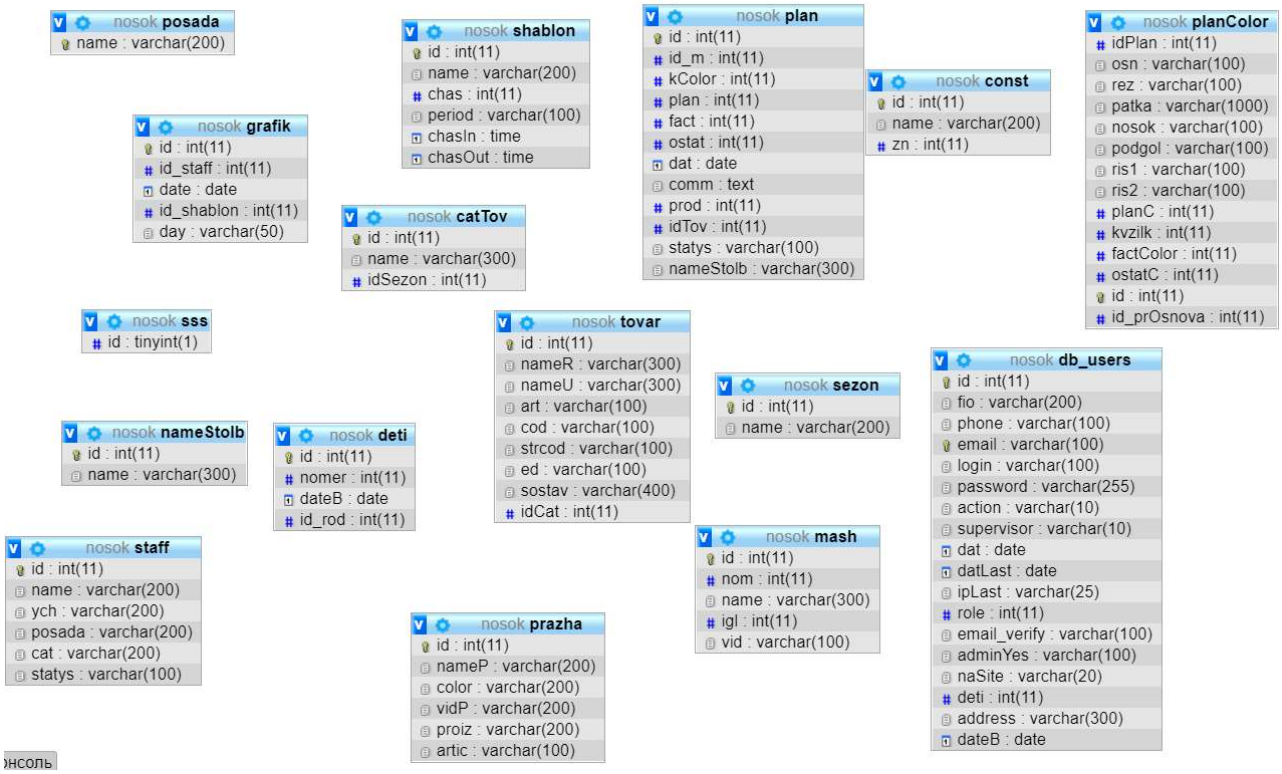


Рис. 2.3. Структурна схема БД

Далі наведемо опис основних таблиць бази даних.

Таблиця «tovar» - таблиця містить інформацію про продукцію компанії. Структура таблиці наведена нижче:

Таблиця 2.2

Таблиця tovar

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
nameR	Varchar	-	-	Назва продукції
nameU	Varchar	-	-	Назва українською
art	Varchar	-	-	Артикуль
cod	int	-	-	Код
strcod	Varchar	-	-	Строчний код

Продовження таблиці 2.2

Idcat	int	-	+	Ідентифікатор категорії
ed	Varchar	-	-	Одиниця виміру
sostav	Varchar	-	-	Склад

Таблиця plan призначена для зберігання інформації про план виробництва. Структура таблиці наведена нижче:

Таблиця 2.3

Таблиця plan

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
Id_m	Int	-	-	Ідентифікатор машини
kColor	Int	-	-	Кількість кольорів
plan	Int	-	-	Кількість в плані
fact	int	-	-	Фактично зроблено
ostat	int	-	+	Залишок
dat	date	-	-	Дата
comm	text	-	-	Коментар
prod	Int	-	-	Продукція
idTov	Int	-	+	Ідентифікатор продукції
statys	varchar	-	-	Статус
nameStolb	varchar	-	-	Назва частини продукції за кольорами

Таблиця «shablon» - таблиця містить інформацію про шаблони графіків роботи персоналу компанії. Структура таблиці наведена нижче:

Таблиця 2.4

Таблиця shablon

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
name	Varchar	-	-	Назва шаблону
chas	Varchar	-	-	Час роботи
period	Varchar	-	-	Період шаблону
chasIn	int	-	-	Час початку роботу
chasOut	Varchar	-	-	Час кінця роботи

Таблиця db_users призначена для зберігання інформації про користувачів програми. Структура таблиці наведена нижче:

Таблиця 2.5

Таблиця db_users

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
fio	Varchar	-	-	ПІБ

Продовження таблиці 2.5

Phone	Varchar	-	-	Телефон
email	Varchar	-	-	Пошта
login	int	-	-	Логін
password	Varchar	-	-	Пароль
action	varchar	-	-	Активність
supervisor	varchar	-	-	Додаткові права
dat	date	-	-	Дата реєстрації
datLast	date	-	-	Дата останнього заходу
ipLast	varchar	-	-	Ip адреса
role	int	-	-	Роль
Email_verify	varchar	-	-	Флаг чи підтверджено пошту
adminYes	varchar	-	-	Флаг чи підтвердив адміністратор
deti	int	-	-	Кількість дітей
address	varchar	-	-	Адреса
dateB	date	-	-	Дата народження

Таблиця planColor призначена для зберігання інформації про кількість та колори продукції, що має бути зв'язана згідно плану . Структура таблиці наведена нижче:

Таблиця 2.6

Таблиця planColor

Назва	Тип даних	ПК	ЗК	Опис поля
IdPlan	Int	-	+	Ідентифікатор плана виробництва
osn	Varchar	-	-	Колір основи
rez	Varchar	-	-	Колір резинки
patka	Varchar	-	-	Колір п'ятки
nosok	Varchar	-	-	Колір носка
podgol	Varchar	-	-	Колір підголенка
Ris1	Varchar	-	-	Малюнок 1
Ris2	Varchar	-	-	Малюнок 2
planC	int	-	+	Ідентифікатор кольорів
kvzilk	int	-	-	Кількість циклів
factColor	int	-	-	Фактично зроблено
ostatC	int	-	-	Залишок
id	int	+	-	Ідентифікатор
Id_prOsnova	int	-	+	Ідентифікатор пряжі під основу

Таблиця «prazha» - таблиця містить інформацію про нитки, що будуть використовуватися при виробництві продукції. Структура таблиці наведена нижче:

Таблиця 2.7

Таблиця prazha

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
nameP	Varchar	-	-	Назва пряжі
Color	Varchar	-	-	Колір
vidP	Varchar	-	-	Вид
Proiz	Varchar	-	-	Виробник
artic	Varchar	-	-	Артикул

Таблиця «mash» - таблиця містить інформацію про в'язальні машини, що працюють на виробництві. Структура таблиці наведена нижче:

Таблиця 2.8

Таблиця mash

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
nom	int	-	-	Номер машини
name	Varchar	-	-	Назва
igl	int	-	-	Кількість голок
vid	Varchar	-	-	Вид виробу

Таблиця «workStaff» - таблиця містить інформацію про виконану роботу згідно плану виробництва окремим працівником. Структура таблиці наведена нижче:

Таблиця 2.9

Таблиця workStaff

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
idStaff	Int	-	-	Ідентифікатор робітника
idPlanColor	Int	-	-	Ідентифікатор плану продукції
kol	Int	-	-	Кількість
date	date	-	-	Дата

Таблиця «staff» - таблиця містить інформацію про персонал компанії. Структура таблиці наведена нижче:

Таблиця 2.10

Таблиця staff

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
name	Varchar	-	-	ПІБ
ych	Varchar	-	-	Ділянка роботи
posada	Varchar	-	-	Посада
cat	Varchar	-	-	Категорія
statys	Varchar	-	-	Статус

2.3. Вибір архітектури системи

Для розробки даного проекту було використано патерн проектування HMVC. Архітектура Ієрархічна Модель-Вид-Контролер (англ. Model-View-Controller) дуже важлива в контексті фреймворків, оскільки вона є типовою для більшості web-додатків. HMVC – схема використання декількох шаблонів проектування, за допомогою яких модель , інтерфейс і взаємодія з користувачем розділені на три окремі компоненти таким чином, щоб модифікація одного з компонентів мала мінімальний вплив на інші . Дана схема проектування часто використовується для побудови архітектурного каркаса, коли переходять від теорії до реалізації в конкретній предметній області.

Архітектурний шаблон HMVC поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них; вигляд (View) відповідальний за відображення цих даних користувачеві; контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача і повідомляє про зміни компонента Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами. HMVC поділяє цю частину системи на три самостійні: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація.

У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле). Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

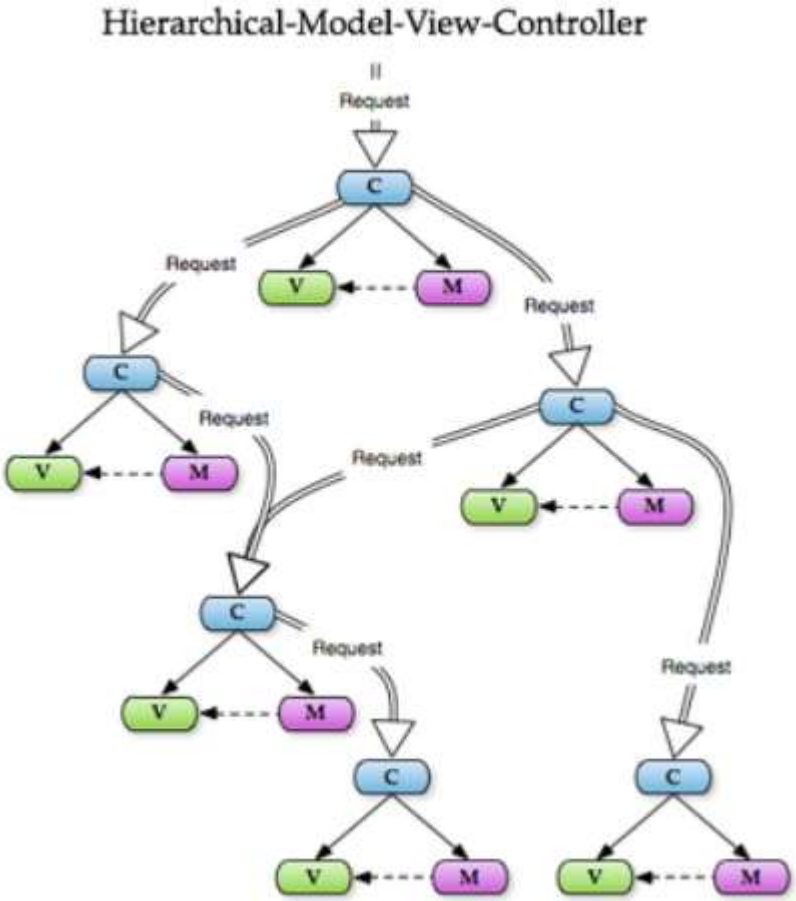


Рис. 2.3 Архітектура Hierarchical Model–View–Controller

Таким чином, можна зробити висновок, що даний архітектурний шаблон забезпечує всі технологічні можливості для побудови системи, що дозволяє реалізувати поставлені завдання.

2. 4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для створення веб-додатку, доцільно застосувати: мову розмітки гіпертекстових документів HTML, каскадні таблиці стилів CSS, браузерну мову програмування JavaScript, мову програмування PHP, веб-сервер Apache з підтримкою PHP та MySQL-баз даних, графічний редактор Adobe Photoshop CS6 та HTML-редактор Adobe Dreamweaver CS6.

Для побудови Web-документів в WWW використовується спеціальна мова **HTML**, що означає HyperText Markup Language - мова гіпертекстової розмітки, мова форматування даних. Заснована на мові SGML (Standard Generalized Markup Language), мова HTML визначає форматування і організацію даних в Web-документах. Вона не визначає те, як точно буде розміщений текст на екрані, ця мова визначає структуру даних. Web-документ може містити не тільки текстову інформацію, і тому мова HTML правильніше було б називати HyperMedia Markup Language, однак в літературі практично завжди вживається абревіатура HTML. Документ, створений на мові HTML - це звичайний файл в ASCII-форматі. В його основі лежать спеціальні дескриптори (теги), які і визначають форматування даних в будь-якому Web-документі. Природно, для перегляду HTML-документів в World Wide Web необхідне спеціальне програмне забезпечення. Такі програми називаються браузерами (від англ. Browse - перегортати, переглядати). З їх допомогою можна завантажувати і переглядати Web-сторінки, здійснювати навігацію по WWW і т.д. На даний момент існує досить велика кількість браузерів, з яких найпопулярнішими є браузери Microsoft Internet Explorer, Netscape Navigator і NCSA Mosaic. Браузер, прочитавши HTML-файл, за допомогою дескрипторів інтерпретує що міститься в документі дані і відповідним чином відображає їх на екрані комп'ютера.

JavaScript − спочатку створювався для того, щоб зробити web-сторінки «живими». Програми на цій мові називаються скриптами. У браузері вони підключаються безпосередньо до HTML і, як тільки завантажується сторінка -

тут же виконуються. Програми на JavaScript - звичайний текст. Вони не вимагають якоїсь спеціальної підготовки.

Сучасний JavaScript - це «безпечна» мова програмування загального призначення. Вона не надає низькорівневих засобів роботи з пам'яттю, процесором, так як спочатку була орієнтований на браузери, в яких це не потрібно. Що ж стосується інших можливостей - вони залежать від оточення, в якому запущений JavaScript. У браузері JavaScript вміє робити все, що відноситься до маніпуляції зі сторінкою, взаємодії з відвідувачем і, в якійсь мірі, з сервером:

Створювати нові HTML-теги, видаляти існуючі, змінювати стилі елементів, ховати, показувати елементи і т.п.

Реагувати на дії відвідувача, обробляти кліки миші, переміщення курсора, натискання на клавіатуру і т.п.

Посилати запити на сервер і завантажувати дані без перезавантаження сторінки (ця технологія називається "AJAX").

Отримувати і встановлювати cookie, запитувати дані, виводити повідомлення.

PHP (англ. PHP: Hypertext Preprocessor - «PHP: препроцесор гіпертексту») - мова програмування, створена для генерування HTML-сторінок на веб-сервері і роботи з базами даних. В даний час підтримується переважною більшістю хостинг-провайдерів. Входить в LAMP - поширений набір для створення веб-сайтів (Linux, Apache, My SQL, PHP (Python або Perl)).

Код PHP може об'єднуватися з тегами XHTML. PHP є вбудовуваним мовою - це означає, що можна переміщатися між чистим кодом HTML і PHP, не жертвуючи можливістю читання тексту. Однак хорошим стилем програмування є відділення програмного коду на мові PHP від подання до вигляду HTML - це найчастіше реалізується за допомогою шаблонізації. Щоб вбудувати код PHP в XHTML, PHP повинен задаватися відокремлено, за допомогою початкового і кінцевого тегів PHP. Теги PHP кажуть серверу Web, де починається і

закінчується код PHP. Аналізатор PHP розпізнає три варіанти початкового і кінцевого тегів.

JetBrains PhpStorm - комерційне кроссплатформене інтегроване середовище розробки для PHP. Розробляється компанією JetBrains на основі платформи IntelliJ IDEA. PhpStorm є інтелектуальним редактором для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок в коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповненя коду в PhpStorm підтримує специфікацію PHP 5.3, 5.4, 5.5 і 5.6 (сучасні та традиційні проекти), включаючи генератори, спів програми, простори імен, замикання, типажі і синтаксис коротких масивів, повноцінний SQL - редактор з можливістю редагування отриманих результатів запитів, PHP UML.

Apache - це HTTP сервер, що володіє високою надійністю і гнучкістю, під HTTP сервером слід розуміти програмне забезпечення для обробки HTTP запитів. Основна робота Apache це обробка і відповідь на HTTP запити і генерування динамічний зміст сторінок. Гнучкість досягається шляхом використання файлу .htaccess, завдяки якому можна перевизначати глобальні настройки сервера Apache. Виглядає це наступним чином. Apache має свої глобальні настройки, але при зверненні до сайту він шукає файл .htaccess, зчитує з нього директиви і застосовує їх, а не глобальні. Зазвичай файл .htaccess лежить в корені сайту, і визначає настройки сервера для всього сайту, проте він може розташовуватися і в будь-якому внутрішньому каталозі сайту, визначаючи таким чином настройки сервера саме для цього каталогу. Веб-майстрами цей файл використовується для настройки редиректів, обробки помилок, безпеки, доступу, кодування і т.д. Apache має механізм для визначення віртуальних хостів, завдяки чому за однією IP адресою може знаходиться необмежену кількість сайтів, саме таким чином працює віртуальний хостинг. Також існує велика кількість модулів, що дозволяють йому працювати з більшістю популярних мов програмування і розширювати функціональність. На сьогоднішній день Apache є одним з найпопулярніших ПО для веб-сервера. Його використовують хостинг-провайдери по всьому світу.

Порівняльний аналіз СКБД дозволяє раціонально вибрати систему керування базами даних для проекту. У якості альтернатив будуть розглянуті наступні СКБД: MySQL, Microsoft SQL Server (далі MS SQL), PostgreSQL. Дві з них безкоштовні – MySQL, PostgreSQL.

Усі обрані системи керування базами даних підходять для проведення аналізу й порівняння тому що реалізують реляційну модель даних.

Таблиця 2.10.

Експертна оцінка багатокористувацьких СУБД

СУБД	Продук-тивність	Конкурентний доступ	Кількість користувачів	Великі БД	Готовність
Microsoft SQL Server	6	7	5	5	7
Oracle	6	9	8	8	8
MySql	5	8	4	5	6

За критерієм тригери й збережені процедури всі альтернативи ідентичні. Усі підтримують тригери, процедури й функції.

Таблиця 2.11.

Аналіз підтримуваних альтернативами операційних систем

	Windows	Linux	Unix	Android	Symbian
MySQL	+	+	+	+	+
MS SQL	+	-	-	-	-
Oracle	+	+	+	-	+

Інформаційна система організації містить у собі конфіденційну інформацію, для запобігання несанкціонованого доступу застосовуються різні способи захисту. Проведемо порівняння систем забезпечення безпеки даних у таблиці

Проведений аналіз таких СКБД, як MySQL, MS SQL та Oracle показав, що для виконання поставлених задач найбільш підходящими є MySQL та Oracle. MySQL має ряд суттєвих переваг над Oracle, зокрема MySQL звичайно набагато перевершує Oracle по швидкості роботи. Крім того, в MySQL 5.0 реалізований кеш запитів. Він дозволяє в багато разів збільшити швидкість обробки запитів для сайтів, на яких переважають неодноразово повторювані запити на читання.

Отже нами було обрано для створення програмного комплексу такі інструментальні засоби та технології:

– Клієнтські інструменти: HTML, як мову розмітки гіпертекстових документів, CSS - каскадні таблиці стилів, браузерну мову програмування JavaScript;

– Серверні інструменти: PHP - мову програмування, середовище керування базами даних MySQL;

– веб-сервер Apache;

– OpenServer

Використано такі програмні продукти та інтегровані середовища розробки: графічний редактор Adobe Photoshop CS6; середовище розробки Php Storm 16.0.1.

2.5 Реалізація додатку системи автоматизації компанії

Розглянемо основні методи написані для роботи програми:

Для вибору данних з основних таблиць бази даних (mash, shablon, plan,worker) в класі MainModel написано відповідні методи:

```
public function getPrazha(){
    $tmp=DatabaseModule::instance()->tovar->query('Select * from
prazha');

    return $tmp;
}
public function getCat(){
    $tmp=DatabaseModule::instance()->catTov->select();

    return $tmp;
}
public function getMash(){
    $tmp=DatabaseModule::instance()->tovar->query('Select * from
mash order by nom ');

    return $tmp;
}
```

Для стандартних операцій додавання видалення та редагування в файлі FunctionModel прописані відповідні методи:

Наприклад, метод додавання виробничого плану у відповідну таблицю:

```
public function addplan($id_m,$nameStolb,$id_t,
$kilColor,$zagPlan,$zagFact,$zagZal,$date,$statys,$comm,$osn,$rez,
$patka,$nosok,$podgol,$ris1,$ris2,$plan,
    $kolz,$fact,$zal,$osnhid){
$nameStolb1=implode(',',$nameStolb);

        $idPlan=DatabaseModule::instance()->plan->insert(
            ["id_m"=>$id_m,
                "idTov"=>$id_t,
                "kColor"=>$kilColor,
                "plan"=>$zagPlan,
                "fact"=>$zagFact,
                "ostat"=>$zagZal,
                "dat"=>$date,
                "statys"=>$statys,
                "nameStolb"=>$nameStolb1,
                "comm"=>$comm,
            ]
        );
        for($i=0;$i<$kilColor;$i++){
            $str="INSERT INTO `planColor`(`idPlan`, `osn`, `rez`,
`patka`, `nosok`, `podgol`, `ris1`, `ris2`,
 `planC`, `kvzilk`, `factColor`, `ostatC`,`id_prOsnova`)
VALUES({$idPlan},'{$osn[$i]}','{$rez[$i]}','{$patka[$i]}','{$nosok
[$i]}','{$podgol[$i]}'

,'{$ris1[$i]}','{$ris2[$i]}','{$plan[$i]}','{$kolz[$i]}','{$fact[$
i]}','{$zal[$i]}',$osnhid[$i])";
            DatabaseModule::instance()->planColor->query($str);
        }
    }
```

Для редагування виробничого плану з врахуванням планування виробництва за кольорами написано метод editPlan

```
public function editplan($idPlan,$nameStolb,$id_m,$id_t,
$kilColor,$zagPlan,$zagFact,$zagZal,$date,$statys,$comm,$osn,$rez,
$patka,$nosok,
                            $podgol,$ris1,$ris2,$plan,
                           $kolz,$fact,$zal,$osnhid){
        $nameStolb1=implode(',',$nameStolb);
        DatabaseModule::instance()->plan->updatePlan(
            ["id_m"=>$id_m,
                "idTov"=>$id_t,
                "kColor"=>$kilColor,
                "plan"=>$zagPlan,
                "fact"=>$zagFact,
                "ostat"=>$zagZal,
                "dat"=>$date,
                "statys"=>$statys,
                "nameStolb"=>$nameStolb1,
                "comm"=>$comm,
```

```
            ],"id={$idPlan}"
        );
        $ss="delete from `planColor` where idPlan={$idPlan}";
        DatabaseModule::instance()->planColor->query($ss);

        for($i=0;$i<$kilColor;$i++){

            $str="INSERT INTO `planColor`(`idPlan`, `osn`, `rez`,
`patka`, `nosok`, `podgol`, `ris1`, `ris2`,
`planC`, `kvzilk`, `factColor`, `ostatC`,`id_prOsnova`)
VALUES({$idPlan},'{$osn[$i]}','{$rez[$i]}','{$patka[$i]}','{$nosok
[$i]}','{$podgol[$i]}'
,'{$ris1[$i]}','{$ris2[$i]}','{$plan[$i]}','{$kolz[$i]}','{$fact[$
i]}','{$zal[$i]}',$osnhid[$i])";

            DatabaseModule::instance()->planColor->query($str);

        }

    }
```

Всі основні методи, що прописані в моделях використовують базові методи роботи з базою даних. Базові методи прописані в класі DatabaseModule. Код класу DatabaseModule наведено в додатку А.

Для швидкого пошуку даних в таблиці використовуємо код js:

```
$('.table-filters input').on('input', function () {
    filterTable($(this).parents('table'));
});
function filterTable($table) {
    console.log("jkjhk");
    var $filters = $table.find('.table-filters td');
    var $rows = $table.find('.table-data');
    $rows.each(function (rowIndex) {
        var valid = true;
        $(this).find('td').each(function (colIndex) {
            if ($filters.eq(colIndex).find('input').val()) {
                if ($(this).html().toLowerCase().indexOf(

$filters.eq(colIndex).find('input').val().toLowerCase()) == -1) {
                    valid = valid && false;
                }
            }
        });
        if (valid === true) {
            $(this).css('display', '');
        } else {
            $(this).css('display', 'none');
        }
    });
}
```

Для занесення щоденно виконаної роботи працівником компанії та розрахунку залишків та фактичного виконання за планом виробництва написано наступні запити.

```
$allKol=0;
  for($i=0;$i<count($kol);$i++){
$sql="Update planColor set  factColor=factColor+".$kol[$i].",
ostatC=ostatC-".$kol[$i]." where id=".$idPlan[$i];
      $tmp=DatabaseModule::instance()->planColor->query($sql);
      $st="Insert into workStaff(`idStaff`, `idPlanColor`, `kol`,
`date`) values({$idStaff},{$idPlan[$i]},{$kol[$i]},curdate())";
      $tmp=DatabaseModule::instance()->workStaff->query($st);
    $allKol=$allKol+ $kol[$i];
  }
        $ss="Update plan set fact=fact+".$allKol.", ostat=ostat-
".$allKol." where id=".$Plan;
        $tmp=DatabaseModule::instance()->plan->query($ss);
    }
    public function getAllWork($id){
        $sql="SELECT staff.name,staff.ych,workStaff.kol,
workStaff.date,tovar.nameU,mash.nom FROM `db_users` inner join
staff on staff.id=db_users.idStaff inner join workStaff on
workStaff.idStaff=db_users.id inner join planColor on
planColor.id=workStaff.idPlanColor inner join
plan on plan.id=planColor.idPlan inner join mash on
mash.id=plan.id_m inner join tovar on tovar.id=plan.idTov where
db_users.id=".$id;
        $tmp=DatabaseModule::instance()->workStaff->query($sql);
           return $tmp;
```

Для формування звітів в Excel використовуємо бібліотеку PHPExcel - відмінна бібліотека з величезним функціоналом по роботі з форматами xls, xlsx. Можна зчитувати, записувати, змінювати форматування, задавати формули, а з xlsx можна і картинки витягувати. Код формування звітів в Excel наведено в додатку Б.

Для відправки листів користувачам програми викоритовуємо бібліотеку PHPMailer - дуже зручна і популярна бібліотека для відправки e-mail повідомлень з вашого сайту. PHPMailer - клас для передачі електронної пошти. Сьогодні він є одним з найпопулярніших способів відправки листів у PHP-розробників.

Для роботи звичайної пошти використовується підсистема електронної пошти сервера. Це означає, що якщо ви захочете змінити деякі настройки,

наприклад, SMTP-сервер або параметри аутентифікації, вам потрібно буде зробити це в масштабах всієї системи.

Якщо у вас немає виділеного сервера або ж ви використовуєте локальну середовище розробки PHP, то цей процес може виявитися досить складним і у вас, ймовірно, навіть не буде дозволу на цю операцію.

Одночасне використання різних конфігурацій, наприклад, використання декількох облікових записів SMTP і програмного перемикання між ними також є практично неможливим.

PHPMailer дозволяє з легкістю змінити будь-який параметр прямо в вашому PHP-скрипт.

Електронна пошта (функція mail) не забезпечує додатковими функціями і відмінно підходить для відправки простих текстових листів. Однак, якщо вам знадобиться, додати вкладення або відправити електронний лист в форматі HTML, простіше виконати це з PHPMailer. Код відправки пошти наведено у додатку В.

Висновки до другого розділу

Веб додаток, системи автоматизації виробництва компанії Lomany реалізує функціонал роботи компанії, що пов’язаний з виробництвом продукції. Система веде облік робочого часу робітників згідно запланованого плану виробництва.

Для збереження користувацьких даних було реалізовано базу даних з 16 таблиць основними з яких є таблиця для збереження плану виробництва, а також таблиця графіку роботи персоналу.

Було розглянуто алгоритми основних процесів системи та проаналізовано взаємодію класів системи під час замовлення користувачів, а також розглянуто порядок дій адміністратора системи.

В останньому пункті наводиться детальний розгляд деяких аспектів реалізації: описано методи головного скрипта-контролера, методи скрипта для реалізації плану виробництва, детально описано процедуру складання графіку виробництва.

РОЗДІЛ 3. ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ КОМПАНІЇ LOMANY

3.1 Порядок встановлення та налаштування системи

Веб додаток, система автоматизації компанії Lomany це веб орієнтоване програмне забезпечення, що було створено на мові PHP. Після побудови релізної версії та заливання її на хостинг веб додаток доступний в будь якому браузері, що відповідає встановленій цілі. Перед перенесенням системи на хостинг необхідно змінити дані в конфігураційних файлах. Система використовує MySQL базу, отже потрібно перенести базу даних на сервер і оновити конфігураційні файли. Порядок дій в такому випадку наступний:

1. Створюння копії (бекап) бази даних на старому сервері. Якщо поточний провайдер надає інструмент для створення MySQL копій, тоді можна скористатись ними. В PhpMyAdmin можна вибрати базу даних і обрати команду 'Експорт'. В діалоговому вікні Зберегти як (Зберегти як) слід вказати параметр 'архівована'. Після цього має початися скачування бази.

2. Створення бази даних на сервері. Як тільки скачування закінчиться, визначити назву файл. Перед відновленням, потрібно створити базу на сервері та користувача.

Імпорт даних бази на сервер. Після створення порожньої база даних, в PhpMyAdmin на сервері слід обрати команду 'Імпорт' та вибрати копію бази. Почнеться процес відновлення бази даних. Цей процес може зайняти деякий час, якщо розмір бази занадто великий. Тепер, коли база даних відновлена, потрібно відкрити конфігураційний файл скрипта і змінити значення хоста бази даних, ім'я бази даних, ім'я користувача бази даних і пароль бази даних.

Параметри налаштування системи зберігаються у конфігураційному файлі:

```
define("URLROOT","");
define("DOCROOT",$_SERVER['DOCUMENT_ROOT']."/".URLROOT);
define("APP_PATH",DOCROOT."app/");
define("CORE_PATH",DOCROOT."core/");
define("HELPER_PATH",DOCROOT."helpers/");
define("MODULES_PATH",DOCROOT."modules/");
```

```
define("CONTROLLERS_PATH",APP_PATH."controllers/");
define("MODELS_PATH",APP_PATH."models/");
define("VIEWS_PATH",APP_PATH."views/");
define("TEMPLASE_PATH",APP_PATH."templase/");
define("CORE_BASE_PATH",CORE_PATH."base/");
define("CONFIGS_PATH",APP_PATH."configs/");
define("CORE_CLASSES_PATH",CORE_PATH."classes/");
define("EXEPTION_PATH",CORE_PATH."classes/Exeptions/");
```

З метою забезпечення можливості дистанційної роботи і незалежності від операційної системи користувача, даний програмний комплекс потребує лише комп'ютер користувача та браузер.

Розроблена програма буде використовуватися на IBM PC/AT сумісних комп'ютерів. Дана підсистема працює під такими операційними системами: Windows 7, 10. Також можливе встановлення під *nix системи, типа Ubuntu, Fedora, RedHat.

Зважаючи на те, що підсистема для своєї роботи не потребує значної кількості апаратних ресурсів, конфігурація комп'ютера, що необхідна буде залежати головним чином від вибору операційної системи.

Наприклад в системі Windows 10 для роботи програми буде достатньою наступна конфігурація:

32-розрядний (x86) або 64-розрядний (x64) процесор * з тактовою частотою 1 ГГц або вище.

1 ГБ (для 32-розрядної процесора) або 2 ГБ (для 64-розрядної процесора) ОЗУ.

16 ГБ (для 32-розрядної системи) або 20 ГБ (для 64-розрядної системи) вільного місця на жорсткому диску.

Графічний пристрій DirectX 9 з драйвером WDDM 1.0 або більш пізньої версії.

Таким чином розроблений програмний код не потребує спеціалізованого апаратного забезпечення, додаткових налаштувань та засобів для розгортання крім стандартних для більшості сучасних сайтів.

3.2 Розробка рекомендацій щодо застосування програмного забезпечення системи автоматизації виробничих процесів компанії Lomany

Для початку роботи з програмою в браузері вводимо домене ім’я нашої програми https://nosok.pp.ua/. Завантажується початкова сторінка програми з авторизацією користувачів в програмі (рис.3.1)

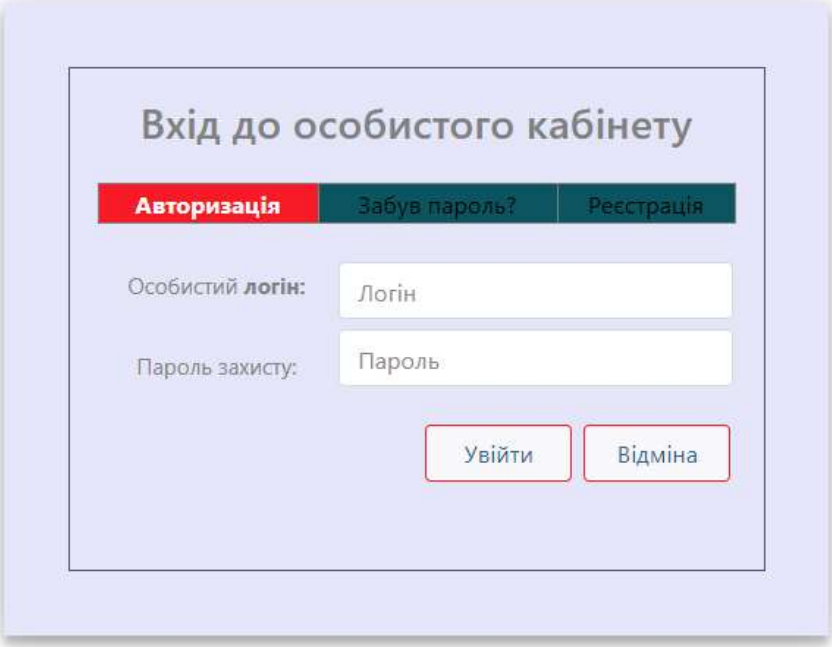


Рис. 3.1 Форма авторизації в системі

При введені вірних даних для користувача завантажується головна сторінка програми (рис.3.2)

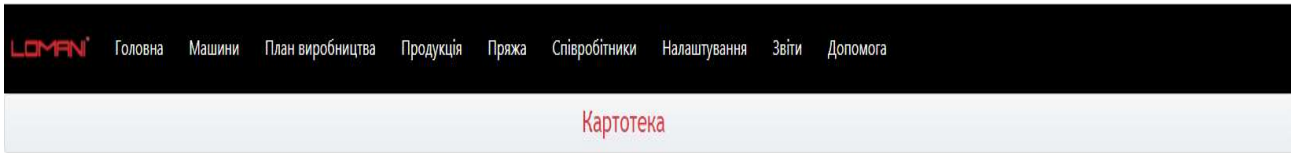


Рис. 3.2 Головне меню програми

Для початку роботи з програмою необхідно заповнити довідникові таблиці на основі яких планується та виконується виробництво продукції. Так першою заповнюється таблиця інформації про машини, що виконують виробничий процес (рис.3.3)

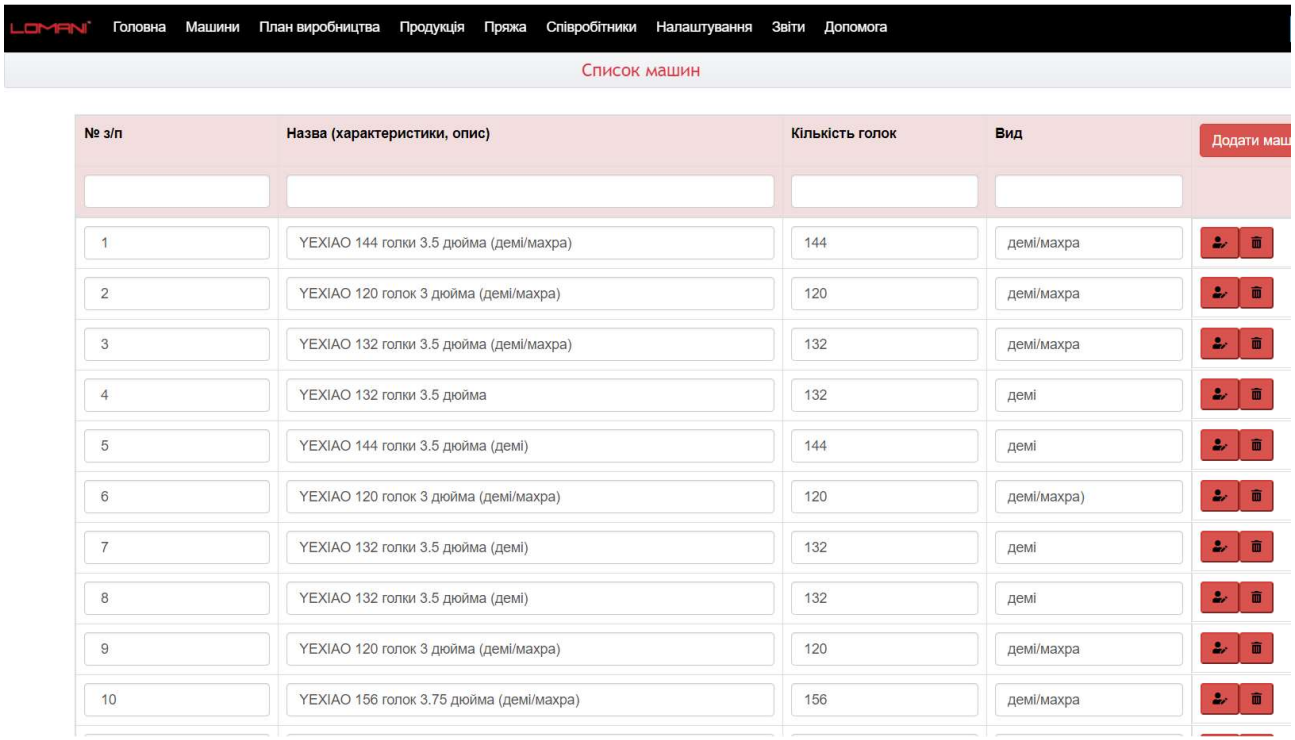


Рис.3.3 Сторінка роботи із машинами

На даній сторінці реалізовано функціонал: - пошуку машин за всіма параметрами; видалення машин; редагування та збереження інформації про машину, а також додавання нової машини. Наступною довідниковою таблицею для заповнення є асортимент продукції (рис.3.4)

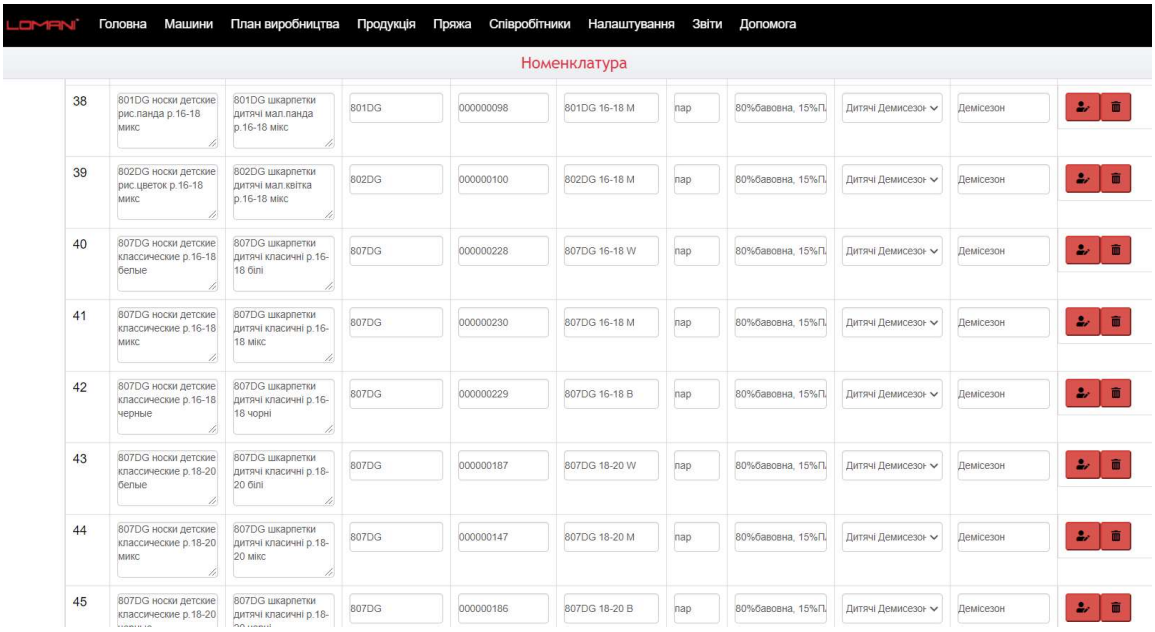


Рис.3.4 Сторінка роботи із продукцією

Додавання продукції виконано через завантаження даних з файлу excel (рис.3.5)

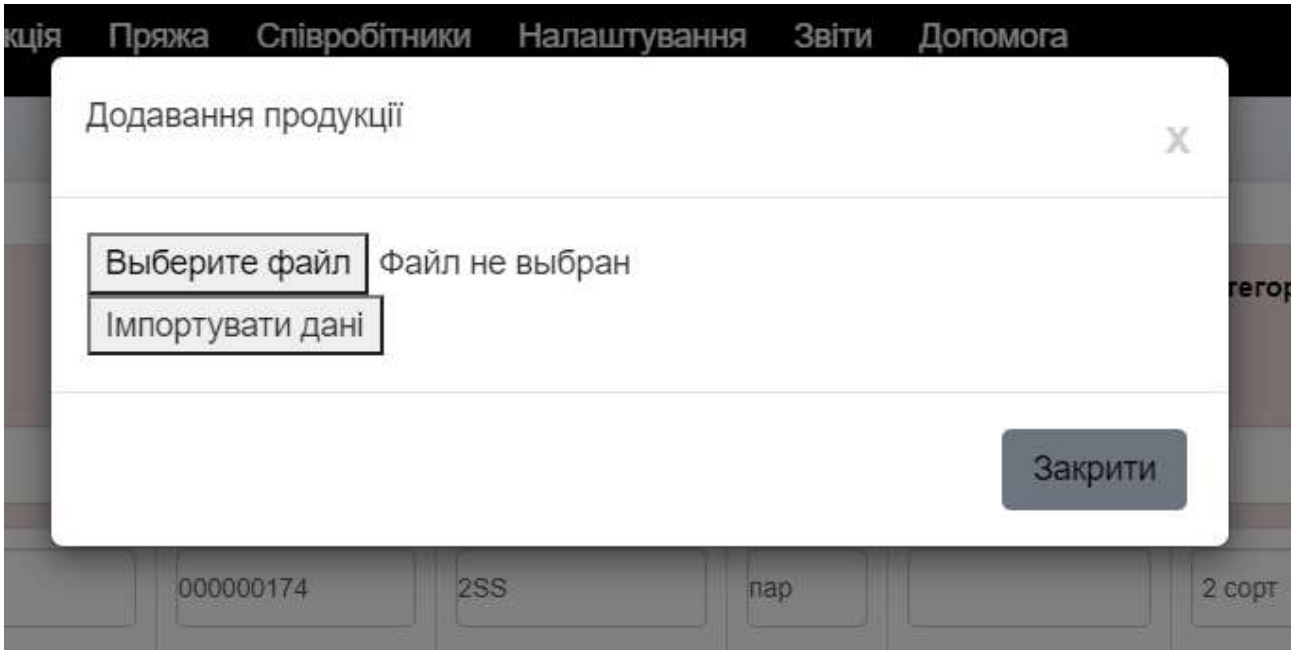
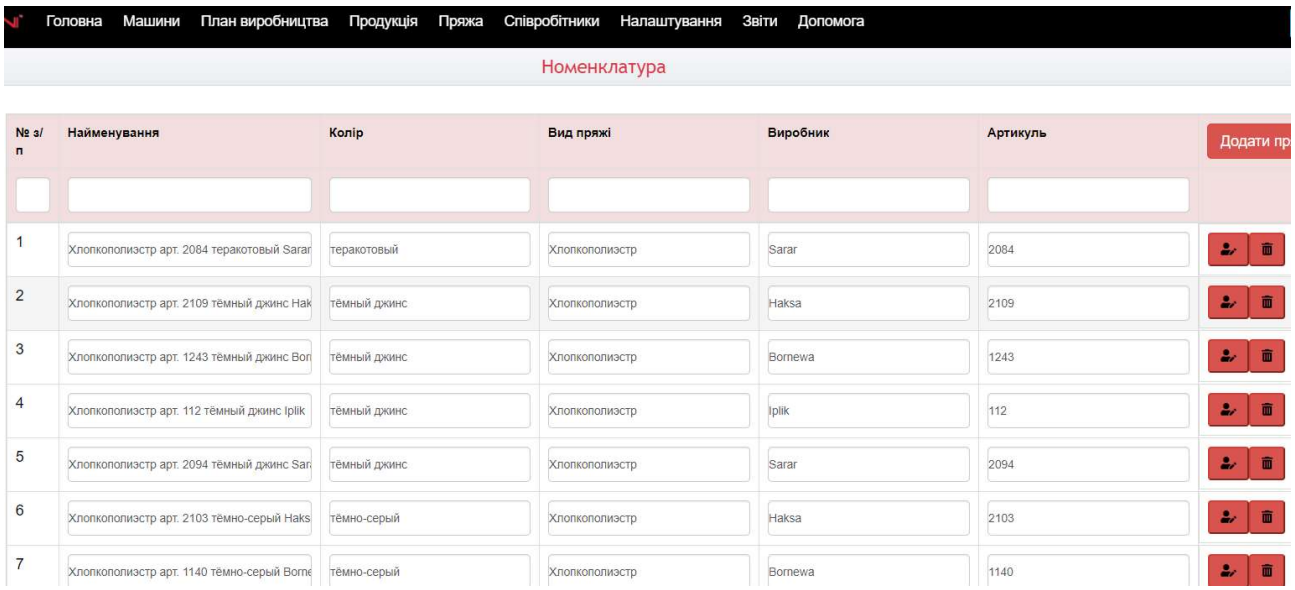


Рис.3.5 Додавання продукції

Наступними довідниковими таблицями є пряжа з якої виробляється продукція та співробітники (рис.3.6)



Головна Машини План виробництва Продукція Пряжа Співробітники Налаштування Звіти Допомога

Список співробітників

№ з/п	ПІБ	Участок	Посада	Категорія	Статус	Додати співр
1	Уборщица	Офис	уборщица	стандарт	работает	
2	Алла Денисевич	Формировка	Формировщица	стандарт	работает	
3	Алла Максимович	Вязальный	начальник смен	стандарт	работает	
4	Антипчук	Вязальный	стажировщик	стандарт	работает	
5	Ахмад Али Султанали	Руководство	Собственник	стандарт	работает	
6	Ашраф Танин Али	Офис	администратор	стандарт	работает	
7	Батюк Таня	Зашивка	Зашивщица	стандарт	работает	
8	Бражук Таня	Упаковка	Упаковщица	стандарт	работает	

Рис.3.6. Довідникові дані для роботи програми

Після внесення довідникових даних складається план виробництва продукції. Для цього заходимо на сторінку план виробництва (рис.3.7)

Головна Машини План виробництва Продукція Пряжа Співробітники Налаштування Звіти Допомога anna

Список машин

Новий план Excel

Машина	Ігли	Артикул	Вид	Кількість кольорів	Основа	План загальний	План за кольорами	Факт по кольорам	Факт загальний	Залишок за кольорами	Залишок загальний	Дата	Статус	Коментар
1	144	0002WG	демі/махра	4	лён	400	100	0	0	100	400	07.03.2021	Поточний	
					темно синий / марине		100	0		100				
					темно зелёный (бутылка)		100	0		100				
					белый		100	0		100				
2	120	819DL	демі/махра	6	персик	360	60	0	0	60	360	17.03.2021	Поточний	
					беж		60	0		60				
					голубой		60	0		60				
					св. серый		60	0		60				
					красный		60	0		60				
					св. розовый		60	0		60				
3	132	909DG	демі/махра	2	чёрный	240	120	0	0	120	240	07.03.2021	Поточний	
					серый шифер		120	0		120				
3	132	831DL	демі/махра	2	оранж	30	15	0	0	15	30	17.03.2021	Новий1	

Рис.3.7 Список планів виробництва

Кожен план в системі можна видалити або редагувати, також можна сформувати excel документ та скопіювати план виробництва для продовження роботи на інший період (рис. 3.8)

Рис.3.8 Дублювання планів виробництва

Для створення нового плану виробництва натискаємо кнопку «Новий план» та завантажуємо сторінку побудови плану (рис.3.9)

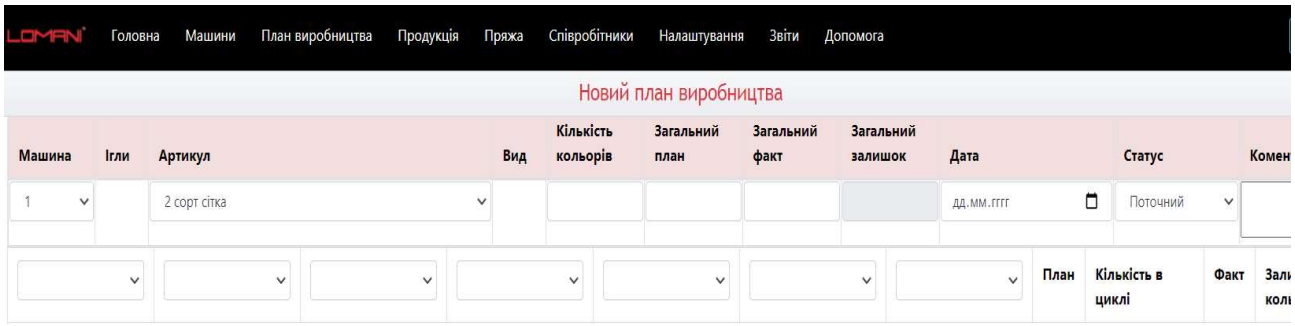


Рис.3.9 Сторінка побудови плану виробництва

Спочатку заповнюємо верхній рядок обираючи номер машини та атикул товару поля ігли та вид запонюються автоматично. Далі необхідно поставити число в поле кількість кольорів обраної продукції буде вироблятися і машина автоматично додасть відповідну кількість рядків до таблиці конкретизації плану виробництва (рис.3.10)

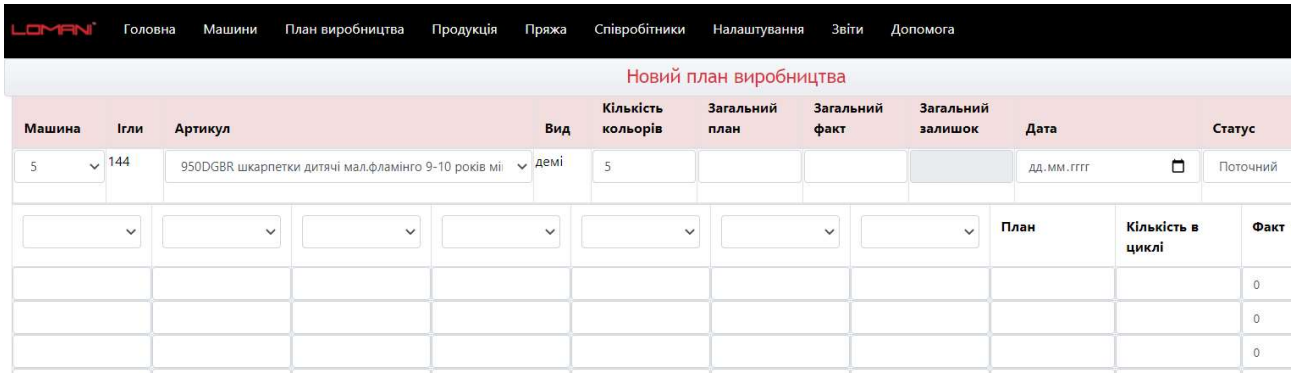


Рис.3.10 Формування плану виробництва

Далі запонюються види частини продукції для формування кольорового складу товару. Види частини продукції обираються з випадаючих списків. Під кожним видом частини обирається пряжа та колір цієї частини продукції (рис.3.11)

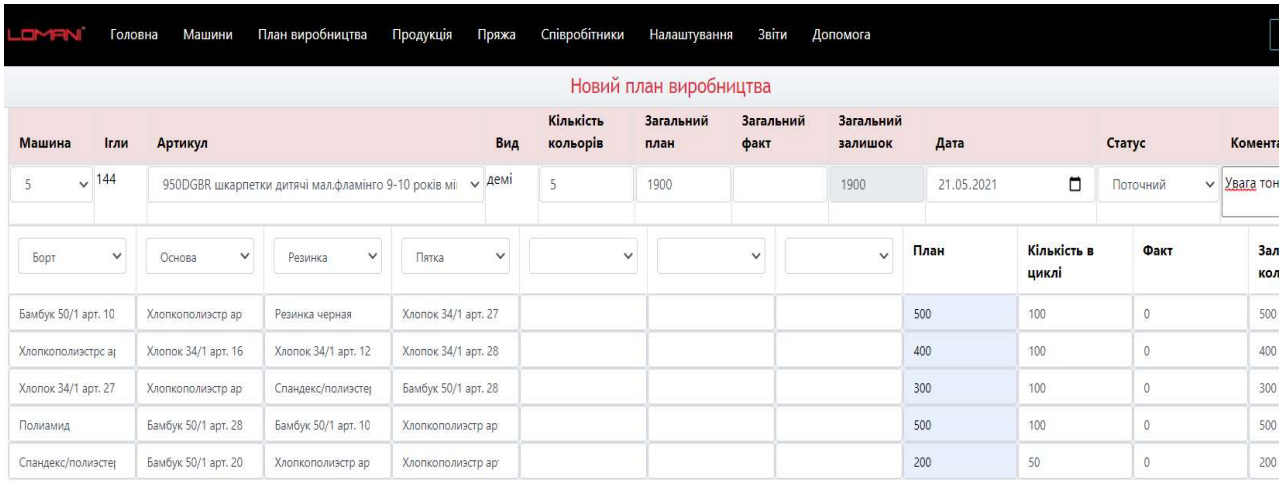


Рис.3.11 Заповнення плану виробництва

Поля Загальний план та факт та залишок заносяться в верхню таблицю автоматично. Після збереження плану виробництва він потрапляє до загального списку планів. Плани сортуються по машинам для зручного перегляду всіх планів за вибраною машиною. План що виконано змінює статус. Взагалі плани мають декілька статусів (Новий1, Новий 2, Поточний, Виконаний, Закритий) (рис.3.12)

3	132	831DL	демі/махра	2	оранж	30	15	0	0	15	30	17.03.2021	Новий1	
					персик		15	0		15				
5	144	0001WG	демі	3	пудра	100	100	0	0	100	100	07.03.2021	Виконаний	
5	144	950DGBR	демі	5	беж	1900	500	0	0	500	1900	21.05.2021	Поточний	Увага тонка нитка
					абрикосовый		400	0		400				
					св. розовый		300	0		300				
					белый		500	0		500				
					черный		200	0		200				

Рис.3.12 Плани виробництва за машинами

Кожен план можна передати в excel для детального опрацювання(рис.3.13)

A	B	C	D	E	F	G	H	I	J	K
План виробництва на 5 машину										
Номер машини	Ігли	Артикул	Вид	Кількість кольорів	План загальний	Факт загальний	Залишок загальний	Дата	Статус	Коментар
5	144	950DGBR	демі	5	1900	0	1900	2021-05-21	Поточний	Увага тонка
Борт	Основа	Резинка	Пятка				План	Кількість в	Факт	Залишок по кольорам
Бамбук 50/1 арт. 1022 беж Osborn	Хлопкополиэстр арт. 102 кремовый Iplik	Резинка черная	Хлопок 34/1 арт. 2795 яр. салатовый Osborn				500	100	0	500
Хлопкополиэстрс арт. 1118 абрикосовый Bornewa	Хлопок 34/1 арт. 1681 яр. мята Osborn	Хлопок 34/1 арт. 1261 св. фиолет Osborn	Хлопок 34/1 арт. 2874 сред. серый Osborn				400	100	0	400
Хлопок 34/1 арт. 2708 св. розовый Osborn	Хлопкополиэстр арт. 1122 лиловый Bornewa	Спандекс/полиэстер Sp+PES 20/150 черный Китай	Бамбук 50/1 арт. 2818 тем. фиолет Osborn				300	100	0	300
Полиамид	Бамбук 50/1 арт. 2818 тем. фиолет Osborn	Бамбук 50/1 арт. 1072 коралл Osborn	Хлопкополиэстр арт. 102 кремовый Iplik				500	100	0	500

Рис.3.13 Друк плану виробництва

Для роботи співробітників компанії необхідно скласти графік роботи. Для цього необхідно заповнити шаблони роботи компанії (рис.3.14)

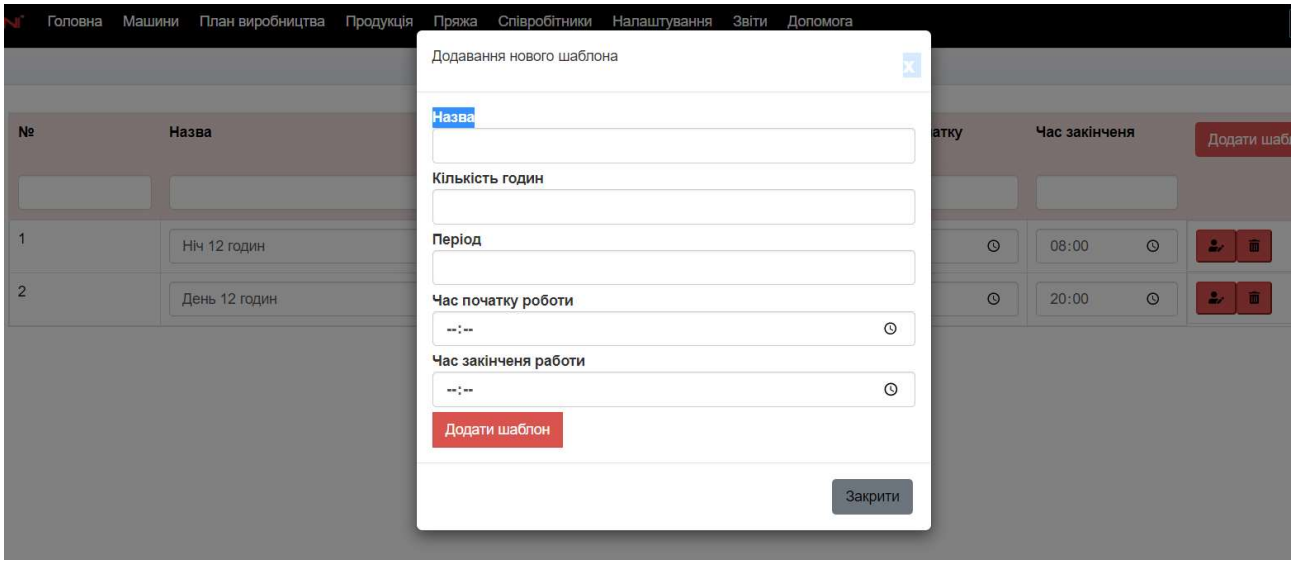


Рис.3.13 Шаблон графіку роботи

Далі формується сам графік роботи співробітників (рис.3.14)

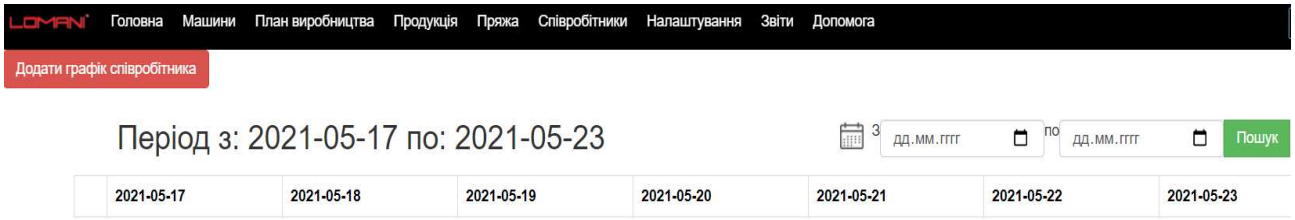
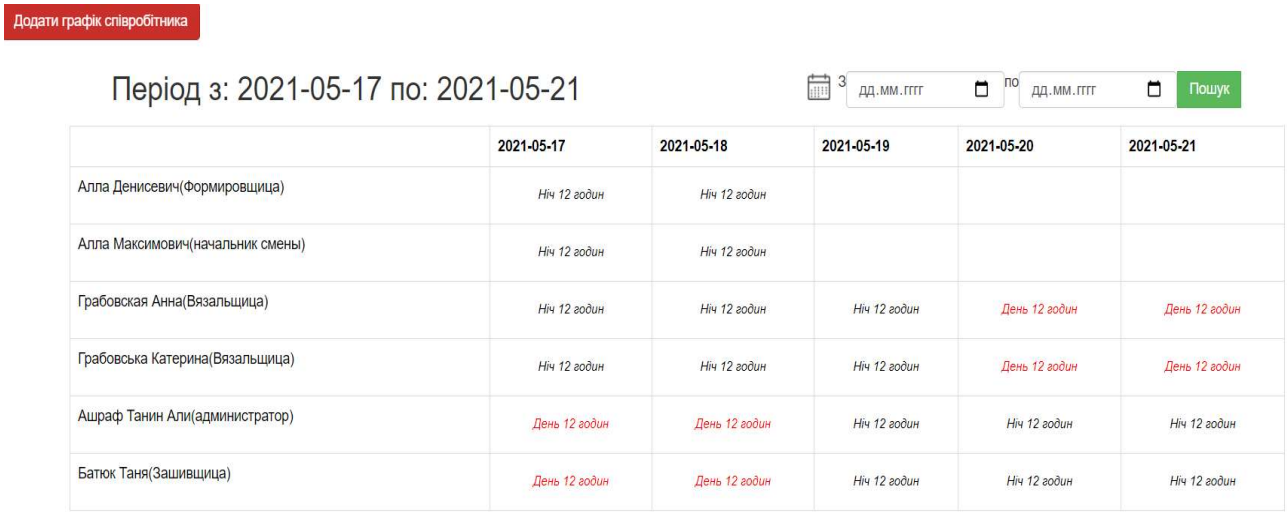


Рис.3.14 Сторінка графіку роботи робітників

Скориставшись пошуком можна переглянути графік роботи робітників за довільний період (рис.3.15)



	2021-05-17	2021-05-18	2021-05-19	2021-05-20	2021-05-21
Алла Денисевич(Формировщица)	Ніч 12 годин	Ніч 12 годин			
Алла Максимович(начальник смены)	Ніч 12 годин	Ніч 12 годин			
Грабовская Анна(Вязальщица)	Ніч 12 годин	Ніч 12 годин	Ніч 12 годин	День 12 годин	День 12 годин
Грабовська Катерина(Вязальщица)	Ніч 12 годин	Ніч 12 годин	Ніч 12 годин	День 12 годин	День 12 годин
Ашраф Танин Али(администратор)	День 12 годин	День 12 годин	Ніч 12 годин	Ніч 12 годин	Ніч 12 годин
Батюк Таня(Зашивщица)	День 12 годин	День 12 годин	Ніч 12 годин	Ніч 12 годин	Ніч 12 годин

Рис.3.15 Пошук за графіком роботи

Для додавання нового графіку для робітника натискаємо кнопку «Додати графік співробітника» (рис.3.16)

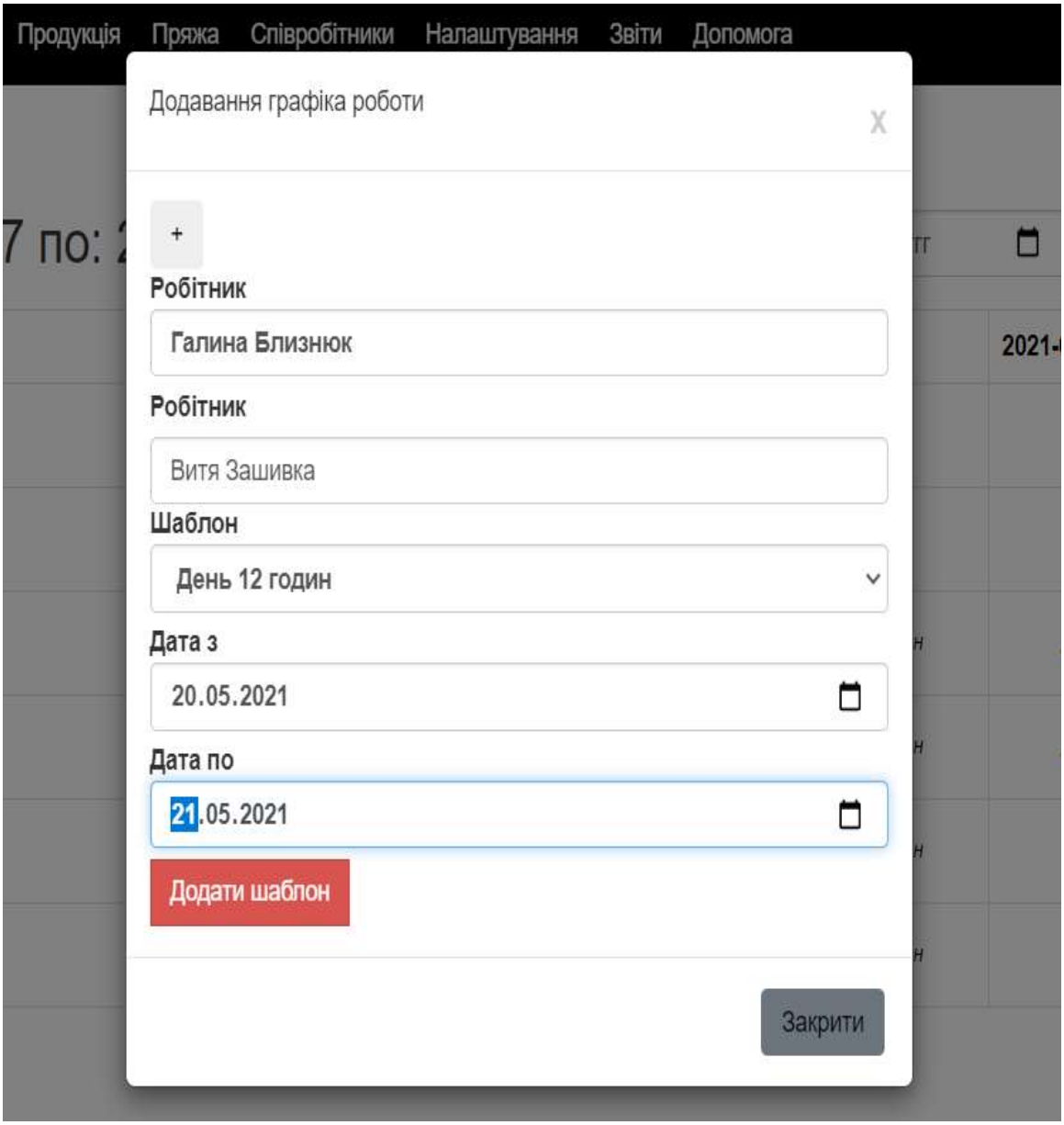


Рис.3.16 Формування графіку роботи робітників

За допомогою кнопки «+» можна Додати декількох робітників до однакового графіку роботи. Для зміни графіку роботи у робітника необхідно клікнути на ячейку з його графіком та в модальному вікні змінити графік (рис.3.17)

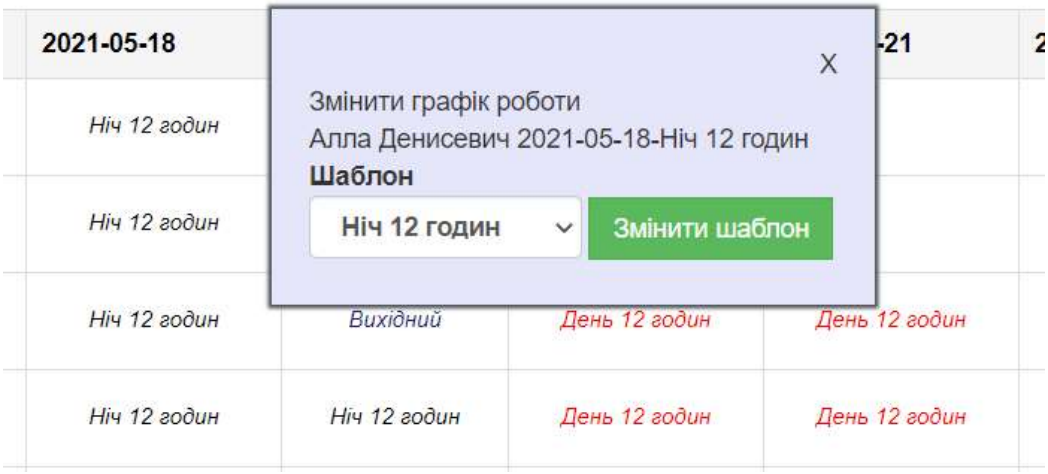


Рис.3.17 Зміна графіку роботи робітника

Кожен робітник реєструється в програмі окремо згідно даних, що занесені в програму адміністратором (рис.3.18)

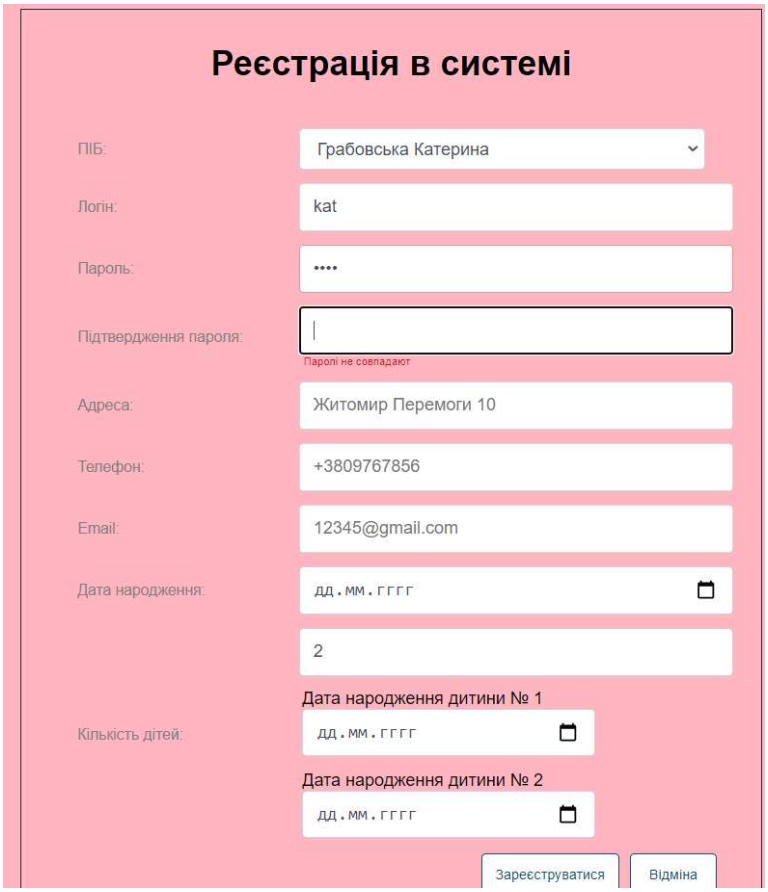


Рис.3.18 Реєстрація робітника в системі

Для робітника в системі відображається окреме меню (рис.3.19)

Рис.3.19 Меню робітника

При виборі пункта меню «Графік роботи» робітник може переглянути та відфільтрувати по датах власний графік роботи (рис.3.20)

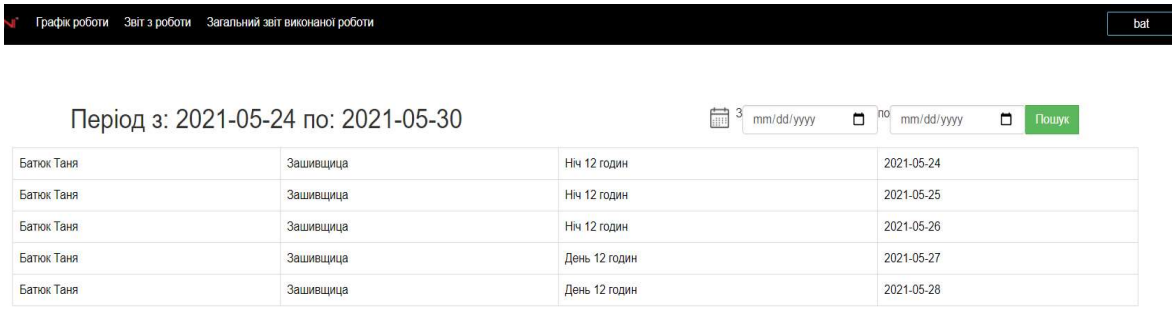


Рис.3.20 Графік роботи робітника

При виборі пункта меню «Звіт з роботи» завантажується форма для внесення результатів роботи за зміну для визначеного робітника (рис.3.21). Робітнику необхідно обрати зі списку номер машини на якій він працював та програма підтягне дані з бази даних про план та продукцію, далі робітник вносить кількість виготовлених виробів.

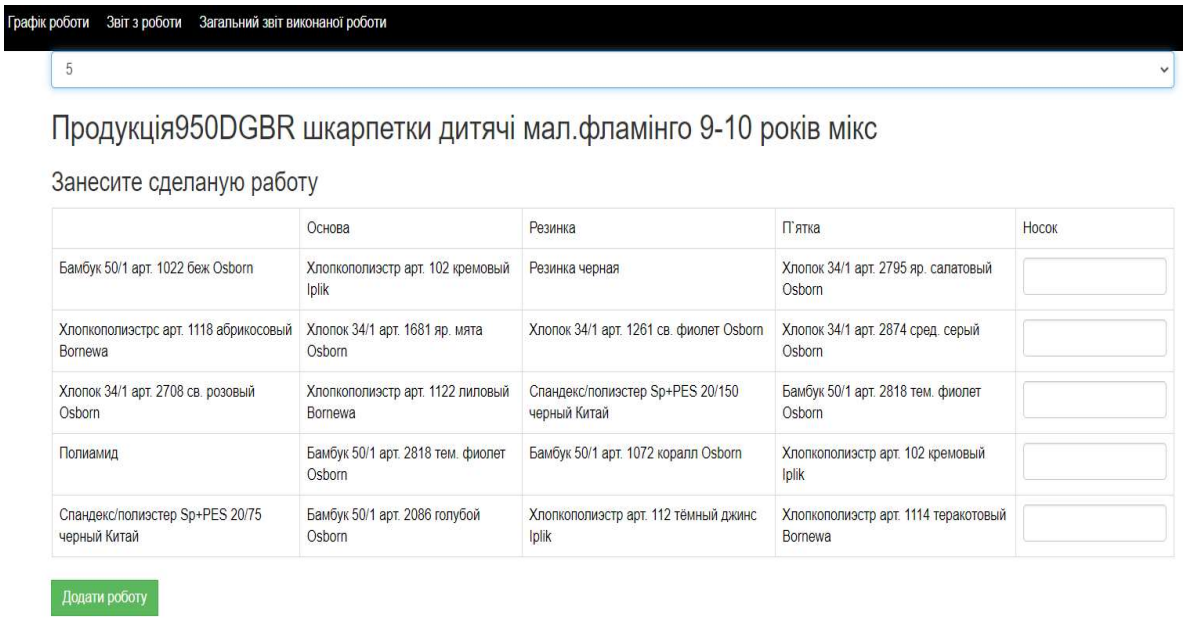


Рис.3.21. Звіт з виконаної роботи

Загальний звіт за період по власній діяльності робітник може переглянути на вкладці «Загальний звіт виконаної роботи» рис.3.22

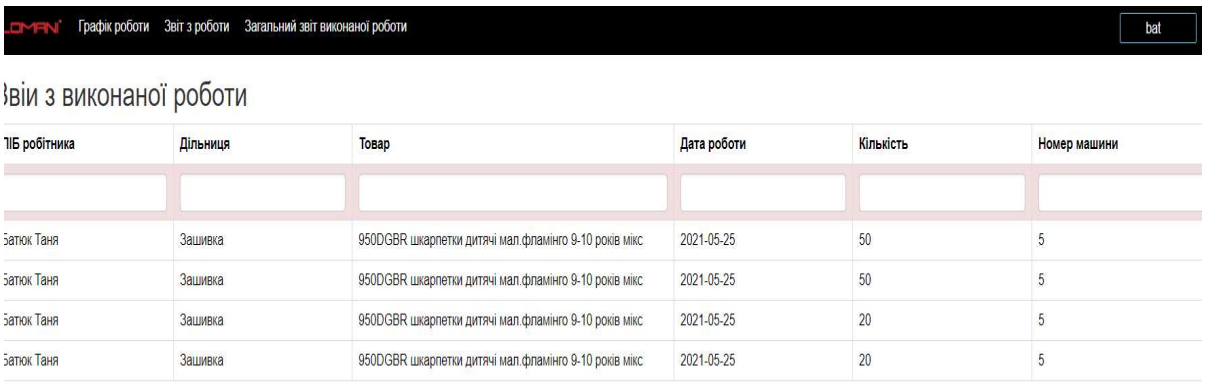


Рис.3.22 Загальний звіт з виконаної роботи

Після внесення робітником даних про виконану роботу дані автоматично заносяться в базу даних та розраховуються залишки по фактичному виробництву та залишки за планом.

3.3 Дослідження працездатності програмного забезпечення системи автоматизації виробничих процесів компанії Lomany

Перевірка на працездатність – невід'ємний процес під час розробки програмного коду системи.

Планується три етапи проведення процесу тестування. Перший етап полягає в аналізі ТЗ, складання критичного чек-листу, а також частковий прогін функціональних тестів. Другий етап буде присвячений зручності використання. На третьому етапі буде проведено кросбраузерне тестування, для перевірки адекватного відображення системи в різних браузерах. Таким чином, досягається максимальна деталізація глибини тестування, що, в свою чергу, дозволяє більш точно визначити витрачені ресурси, а так само дозволяє почати виправляти дефекти на ранніх етапах.

Функціональне тестування Цілі: Виявлення функціональних помилок, невідповідності технічному завданню і очікуванням користувача шлях реалізація стандартний, а також нетривіальний тестові сценарії. Класифікація функцій:

1. Реєстрація / Авторизація:

– авторизація адміністратора/робітника;

2. Робота адміністратора:

– робота з довідниковими таблицями (додавання видалення редагування);

– робота із складанням плану виробництва;

– розрахунок залишків за планом виробництва

– робота із складанням графіку робіт;

– друк інформації;

– фомування звітів.

3. Робота робітника:

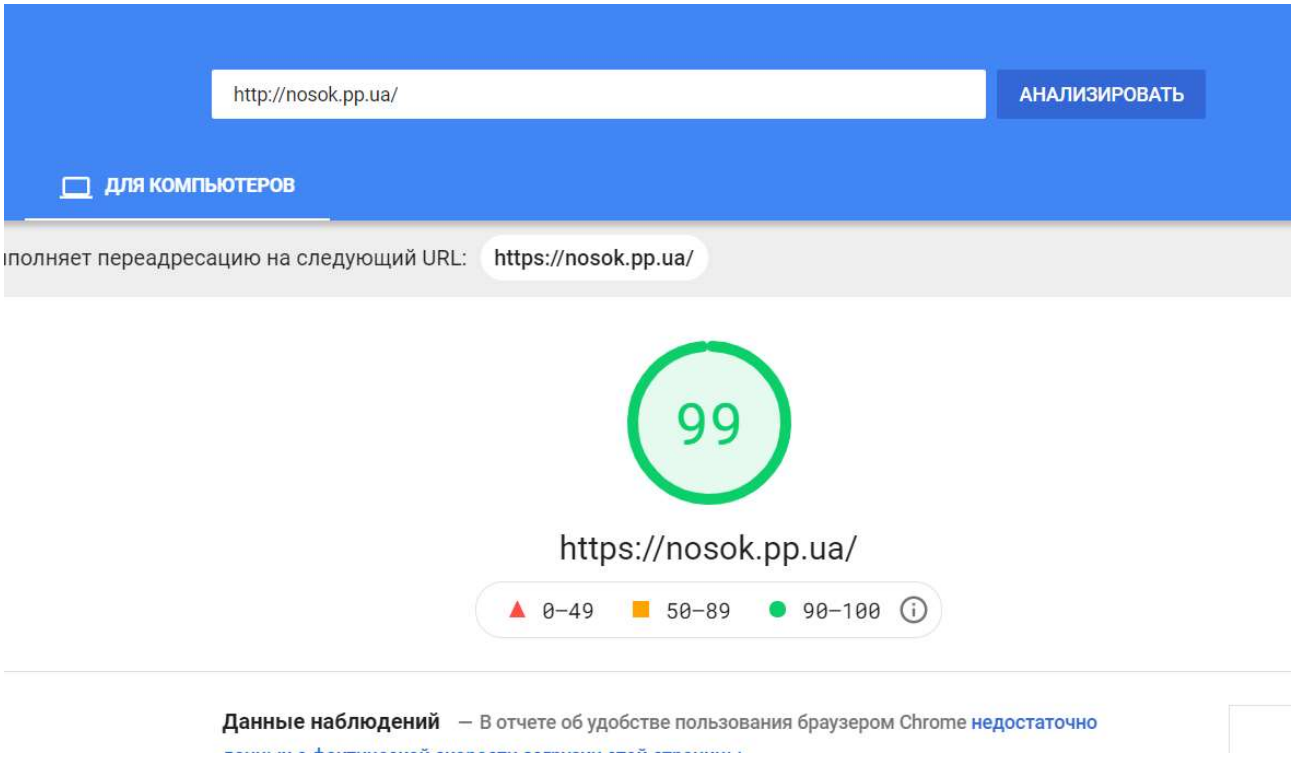
– реєстрація/авторизація;

– зберігання інформації, щодо виконаної роботи;

Тестування в певному середовищі Мета: Перевірити коректну роботу і дизайн Проекту в різних браузерах і при різних розрішеннях монітора Опис конфігурацій

Інструменти для проведення тестування:

1. GooglePageSpeed – сервіс від google для перевірки швидкості та коректності завантаження веб ресурсів (рис.3.19)



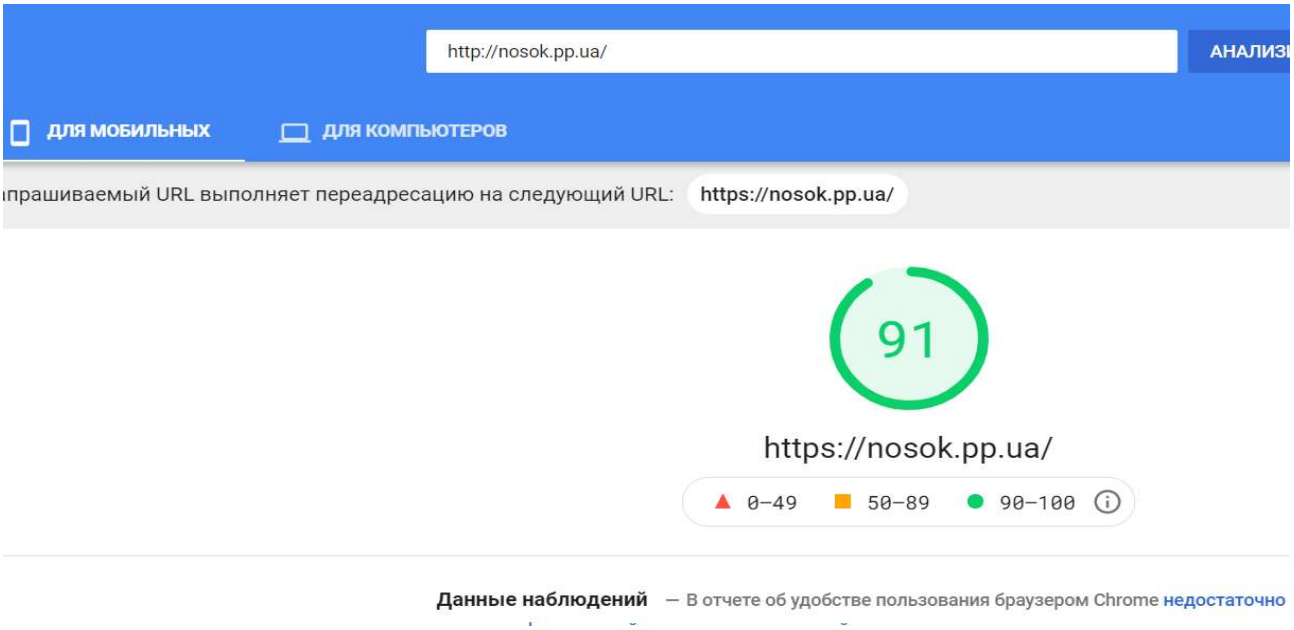


Рис.3.19 Швидкість завантаження сайту

2. Тестування безпеки буде проводитись за допомогою Acunetix веб Vulnerability Scanner. Які уразливості виявляє Acunetix веб Vulnerability Scanner: Cross Site Scripting (виконання шкідливого сценарію в браузері користувача при зверненні і в контексті безпеки довіреної сайту); SQL-ін'єкції (SQL-виконання запитів з браузера для отримання несанкціонованого доступу до даних);

3. Для кросбраузерного тестування використовуймо Інтернет сервіси, які можуть надати реальні пристрої з необхідними браузерами в роздільне користування. Browserstack.com - хмарний симулятор популярних операційних систем і браузерів.

Statistics for managers

Correct internal URLs, by MIME type:

MIME type	count	% count	Σ size	Σ size (KB)	% size	min size	max size	Ø size	Ø size (KB)	Ø time
text/html	102 URLs	45.95%	861267 Bytes	(841 KB)	30.99%	0 Bytes	56161 Bytes	8443 Bytes	(8 KB)	0.029
text/css	5 URLs	2.25%	267739 Bytes	(261 KB)	9.63%	2166 Bytes	142303 Bytes	53547 Bytes	(52 KB)	
image/jpeg	27 URLs	12.16%	1073888 Bytes	(1048 KB)	38.64%	4245 Bytes	174488 Bytes	39773 Bytes	(38 KB)	
image/png	9 URLs	4.05%	458272 Bytes	(447 KB)	16.49%	112 Bytes	364109 Bytes	50919 Bytes	(49 KB)	
application/x-javascript	1 URLs	0.45%	3122 Bytes	(3 KB)	0.11%	3122 Bytes	3122 Bytes	3122 Bytes	(3 KB)	
application/msword	1 URLs	0.45%	41472 Bytes	(40 KB)	1.49%	41472 Bytes	41472 Bytes	41472 Bytes	(40 KB)	
image/gif	77 URLs	34.68%	73291 Bytes	(71 KB)	2.64%	44 Bytes	5229 Bytes	951 Bytes	(0 KB)	
Total	222 URLs	100.00%	2779051 Bytes	(2713 KB)	100.00%					

All pages, by result type:

ok	387 URLs	94.16%
no info to return	1 URLs	0.24%
not found	16 URLs	3.89%
skip type	3 URLs	0.73%
mail host ok	3 URLs	0.73%
no such host	1 URLs	0.24%

Рис.3.20. Загальна статистика перевірки системи

Звіт сканера безпеки Acunetix WVS містить:

1. Деталі сканування (Scan деталі - дату, час, конфігурацію програми та сервера, відповідний профіль тестування тощо).
2. Загальний рівень загроз (Рівень загрози) з коментарем.
3. Узагальнену інформацію про потенційні вразливості (агент резюме).
4. Деталізовану технічну інформацію про вразливості (подробиці Пильні).
5. Рекомендації щодо усунення слабких місць (рекомендація).

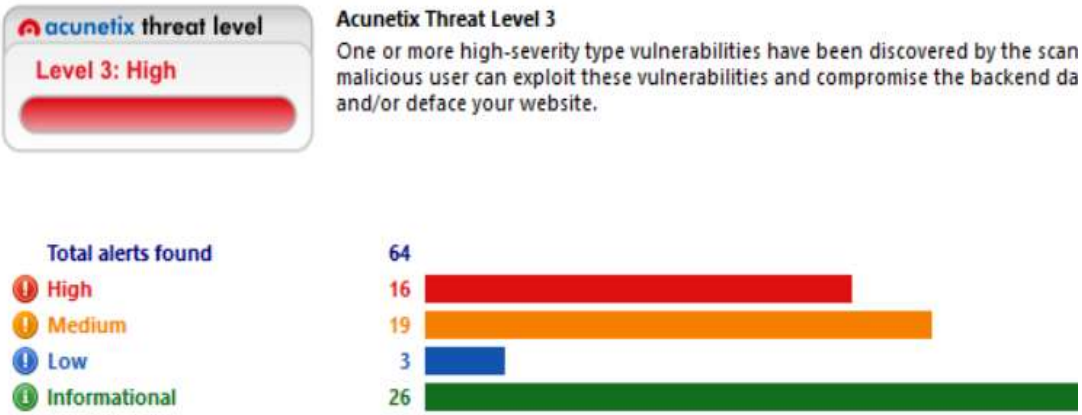


Рис.3.21. Загальний звіт про тестування безпеки системи

Під час тестування доступного функціоналу було виявлено декілька помилок не критичного рівня які було виправлено.

Висновки до третього розділу

В даному розділі було наведено порядок встановлення та налаштування додатку та визначено мінімальну конфігурацію системи для комфортної роботи системи.

Реалізована система містить інтерфейси головного вікна, меню, інтерфейс та шаблон адміністратора та технічного директора, а також робітника компанії.

Проаналізовано та перевірено працездатність системи. Проведено тестування веб системи, виконано функціональне тестування, тестування швидкості, продуктивності, кроссбраузерності, адаптивності та безпеки. Функціональних помилок високого рівня не знайдено.

ВИСНОВКИ

Під час виконання випускної роботи бакалавра було спроектовано та реалізовано веб додаток автоматизації компанії Lomany для цього було вирішено наступні завдання:

1. Проведений аналіз предметної області дозволив визначити основні аспекти роботи компанії. Огляд наявних аналогів та виявлення протиріч між наявними можливостями та потребами користувачів дали результат, що головними функціями майбутньої системи мають бути: зручність GUI, наявність двох інтерфейсів адміністратора та робітників, реалізація модулю планування виробництва, формування звітів та графіка робіт. Визначившись з оптимальною архітектурою обрано необхідний та достатній набір програмних продуктів для реалізації поставлених задач: OpenServer, PHP, MySQL, HTML5, CSS3, JS.

2. Було проаналізовано функціональні та нефункціональні вимоги до веб-додатку, системи автоматизації компанії, для чого побудовано діаграму варіантів використання та на її основі розроблено діаграму класів для аналізу структури системи.

3. Реалізовано базу даних у відповідності до вимог реляційної моделі, яка забезпечує збереження та колективний доступ до інформації системи. База даних складається з 16 таблиць. Основними є Shablon, Grafik, Plan, PlanColor, Prazha, Mash.

4. Було розглянуто алгоритми основних процесів системи та проаналізовано взаємодію класів системи під час роботи.

5. Було наведено детальний розгляд деяких аспектів реалізації: описано методи головного скрипта-контролера, методи скрипта роботи із планами виробництва.

6. Реалізована система містить інтерфейси головного вікна, меню, інтерфейс та шаблон адміністратора та робітників компанії.

7. Також було проведено функціональне, безпеки та кросбраузерне тестування додатку. Знайдені помилки було виправлено, система працює коректно та виконує поставлені задачі.

Реалізований програмний комплекс готовий до використання. Система може бути використана для роботи виробничих компаній.

В подальшому дана система може бути розширена шляхом впровадження нових функціональностей. Завдяки обраній архітектурі розширення системи відбуватиметься значно швидше.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гринберг А. С. Информационные технологии управлення / А. С. Гринберг, Н. Н. Горбачев, А. С. Бондаренко. — М. : ЮНИТИ-ДАНА, 2012. — 479 с.
2. Губанова О. В. Управление, прогнозирование, информационные технологии в сервисной деятельности / О. В. Губанова, Т. Ю. Новгородцев, С. В. Чупров. — Иркутск : ИГЗА, 2011. — 286 с.
3. Гужва В. М. Інформаційні системи і технології на підприємствах / В. М. Гужва. — К., 2015. — 400 с.
4. Дубчак Л. В. Інформаційні системи та технології на підприємствах / Л. В. Дубчак. — Ірпінь : Нац. акад. ДПС України, 2016. — 84 с.
5. Косарев В. М. Информационные системы в менеджменте / В. М. Косарев, Е. А. Паршина, Ю. И. Паршин. — Днепропетровск : ДУЗП, 2011. — 256 с.
6. Пророчук Ж. А. Роль програмного забезпечення в управлінні підприємством / Ж. Пророчук // Современные информационные технологии. — 2016. — № 1. — С. 18 – 20.
7. Єсіпова К. А. Єдність і відмінність понять електронний бізнес та електронна комерція / К. А. Єсіпова [Електронний ресурс]. — Режим доступу : http://intkonf.org/esipova-k-a-ednist-i-vidminnosti-ponyat-elektronniybiznes-ta-elektronna-komertsiya.
8. Эймор Д. Электронный бизнес : эволюция и/или революция / Д. Эймор. — М. : Вильямс, 2015. — 752 с.
9. Пономаренко В. С. Інформаційні системи і технології в економіці : посібник / за ред. В. С. Пономаренка. — К. : Академія, 2017. — 544 с.
10. С.В. Синицын, Н.Ю. Налютин, Верификация программного обеспечения. М.: БИНОМ, 2010, 368 с.
11. Довідниково-інформаційний портал htmlbook [Електроний ресурс]. Доступ за посиланням: http://htmlbook.ru/. [Дата звернення 15.05.2021 р.]
12. PHPFAQ [Електроний ресурс]. Доступ за посиланням: http://phpfaq.ru/pdo. [Дата звернення 15.05.2021 р.]

13. E.Wurzer, Why you Should be using PHP's PDO for Database Access, 2012. [Електроний ресурс]. Доступ за посиланням: https://code.tutsplus.com/tutorials/why-you-should-be-using-phps-pdo-for-database-access--net-12059. [Дата звернення 15.05.2021 р.]
14. В.В. Кириллов, Г.Ю. Громов, Структуризованный язык запросов (SQL): учебн. пособ.. СПб: Санкт-Петерб. госуд. техн. универ., 1998. [Електроний ресурс]. Доступ за посиланням: http://www.citforum.ru/database/sql_kg/. [Дата звернення 15.05.2021 р.]
15. Роббинс Д. Н. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Д. Н. Роббинс. — М. : Эксмо, 2014. — 515 с.
16. Фельке-Моррис Т. Большая книга веб-дизайна / Т. Фельке-Моррис. — М. Эксмо, 2012 — 608 с.
17. Бенкс А. React и Redux: функциональная веб-разработка. – СПб.: "Издательский дос "Питер"", 2017. – 336 с.
18. Вілсон Е. MERN Quick Start Guide. – Бірмінгем: Packt Publishing Ltd, 2018. – 302с.
19. HTML5 [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/HTML/HTML5
20. Самоучитель CSS [Електронний ресурс] – Режим доступу до ресурсу: http://htmlbook.ru/samcss
21. JavaScript [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/JavaScript
22. Databases and Collections [Електронний ресурс] – Режим доступу до ресурсу: https://docs.mongodb.com/manual/core/databases-and-collections/
23. HTML5 Canvas Shape Events [Електронний ресурс] – Режим доступу до ресурсу: https://konvajs.org/docs/events/Binding_Events.html
24. Структура випускної кваліфікаційної роботи бакалавра [Електронний ресурс] – Режим доступу до ресурсу: https://learn.ztu.edu.ua/course/view.php?id=3126

ДОДАТКИ

ДОДАТОК А

Код класу DatabaseModule

```
<?php
class DatabaseModule
{
private $dbh,$tables,$table=NULL;
private static $instance = NULL;
public static function instance(){
    return self::$instance===NULL
        ? self::$instance=new self()
        : self::$instance;
}
private function __construct()
{
    $config = ConfigModule::load("Database/Database");
       $opt=[
        PDO::ATTR_ERRMODE=>PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE=>PDO::FETCH_ASSOC
    ];
    $this->dbh=new PDO("mysql:host=localhost;dbname=
nosok;charset=utf8","nosok","nosok",$opt
         );
    $this->dbh->setAttribute(PDO::MYSQL_ATTR_USE_BUFFERED_QUERY,
false);
    $this->tables = $this->dbh->query("SHOW TABLES;")-
>fetchAll(PDO::FETCH_COLUMN);
}
public function __get($name)
{
if(!in_array($name,$this->tables)) throw new Exception("Table
{$name} not exists");
$this->table= $name;
return $this;
    // TODO: Implement __get() method.
}
public function exec($query, array $data=[]){
    $stmt = $this->dbh->prepare($query);
    $stmt->execute($data);
}
    public function exec1($query, array $data=[]){
        return $this->query($query,$data);
    }
    public function queryInsert($str){
        $stmt = $this->dbh->prepare($str);
        $stmt->execute();
$id=$this->dbh->lastInsertId();

return $id;
    }
public function query($query,array
$data=[],$mode=PDO::FETCH_ASSOC){
    $stmt = $this->dbh->prepare($query);
```

```
    $stmt->execute($data);
    return $stmt->fetchAll($mode);
}
public function select($where="1",array $data=[],array
$columns=NULL){
    if($this->table===NULL)throw new Exception("Table was not be
selected");
    $collist = $columns===NULL ?
"*":"`".implode("`,`",$columns)."`";
    $q="SELECT {$collist} FROM `{$this->table}` WHERE {$where}";
      $this->table=NULL;
    return $this->query($q,$data);
}
public function selectColumn($column,$where="1",array $data=[]){
    if($this->table===NULL)throw new Exception("Table was not be
selected");
    $q="SELECT {$column} FROM `{$this->table}` WHERE {$where}";
       $this->table=NULL;
    return $this->query($q,$data,PDO::FETCH_COLUMN);
}
public function selectOne($where="1",array $data=[]){
    if($this->table===NULL)throw new Exception("Table was not be
selected");
    $ss="SELECT * FROM `{$this->table}` WHERE {$where}";
    $stmt=$this->dbh->prepare($ss);
    $stmt->execute($data);
    $this->table=NULL;
    return $stmt->fetch();
}
    public function selectAdmin($where="1",array $data=[]){
        if($this->table===NULL)throw new Exception("Table was not
be selected");
        $ss="SELECT * FROM `db_users` WHERE {$where}";

        $stmt=$this->dbh->query($ss);
$ss=$stmt->fetch();

        return $ss;
    }
public function getById($id){
    return $this->selectOne("id=?",[$id]);
}
    public function getByIdk($id){
        return $this->selectOne("id_k=?",[$id]);
    }
public function insert(array $data){
     if($this->table===NULL)throw new Exception("Table was not be
selected");
    $fields = array_keys($data);
    $q="INSERT INTO `{$this->table}`(`".implode("`,`",$fields)."`)
VALUES(:".implode(",:",$fields).");";

       $this->exec($q,$data);
```

```
    $this->table=NULL;
    return $this->dbh->lastInsertId();
}

    public function insert1(array $data){
                if($this->table===NULL)throw new Exception("Table
was not be selected");
        $fields = array_keys($data);
        $values = array_values($data);

        $q="INSERT INTO `{$this-
>table}`(`".implode("`,`",$fields)."`)
VALUES('".implode("','",$values)."');";
           $this->exec($q,$data);
        $this->table=NULL;
        return $this->dbh->lastInsertId();
    }
    public function insertz($st){
            $this->exec($st,$data);
              return $this->dbh->lastInsertId();
    }
public function deleteWhere($where="0",array $data=[]){
    if($this->table===NULL)throw new Exception("Table was not be
selected");
    $this->exec("DELETE FROM `{$this->table}` WHERE
{$where}",$data);
    $this->table=NULL;
    }
    public function delete($id){
    $this->deleteWhere("`id`=?",[$id]);
    }
    public function updateWhere(array $values, $where="0",array
$data=[]){
        if($this->table===NULL)throw new Exception("Table was not
be selected");
        $new_values=[];
        $q="UPDATE `{$this->table}` SET ";
        foreach ($values as $field=>$value){
            $q.="`{$field}`=:update_prefix_{$field}";
            $new_values["update_prefix_{$field}"]=$value;
        }
        $q.=" WHERE {$where}";

        $data = array_merge($data,$new_values);
        $this->exec($q,$data);
        $this->table=NULL;
    }
    public function updatePlanCopy(array $values, $where="0",array
$data=[]){
        if($this->table===NULL)throw new Exception("Table was not
be selected");
        $new_values=[];
```

```
        $q="UPDATE `{$this->table}` SET ";
        $k=1;
        foreach ($values as $field=>$value){
            $q.="`{$field}`='{$value}'";
            if($k<3){
                $q.=",";
            }
            $k=$k+1;
        }
        $q.=" WHERE {$where}";
        $data = array_merge($data,$new_values);
        $this->exec($q,$data);
        $this->table=NULL;
    }
    public function updatePlan(array $values, $where="0",array
$data=[]){
        if($this->table===NULL)throw new Exception("Table was not
be selected");
        $new_values=[];
        $q="UPDATE `{$this->table}` SET ";
        $k=1;
        foreach ($values as $field=>$value){
            $q.="`{$field}`='{$value}'";
            if($k<10){
                $q.=",";
            }
            $k=$k+1;
        }
        $q.=" WHERE {$where}";
        $data = array_merge($data,$new_values);
        $this->exec($q,$data);
        $this->table=NULL;
    }
```

ДОДАТОК Б

Код формування звітів в Excel.

```
$titles = array(
    array(
        'name'=>'Номер машини',
        'cell'=>'A'
    ),
    array(
        'name'=>'Ігли',
        'cell'=>'B'
    ),
    array(
        'name'=>'Артикул',
        'cell'=>'C'
    ),
    array(
        'name'=>'Вид',
```

```
            'cell'=>'D'
        ),
        array(
            'name'=>'Кількість кольорів',
            'cell'=>'E'
        ),
        array(
            'name'=>'Основа',
            'cell'=>'F'
        ),
        array(
            'name'=>'План загальний',
            'cell'=>'G'
        ),
        array(
            'name'=>'План за кольорами',
            'cell'=>'H'
        ),
        array(
            'name'=>'Факт по кольорам',
            'cell'=>'I'
        ),
        array(
            'name'=>'Факт загальний',
            'cell'=>'J'
        ),
        array(
            'name'=>'Залишок за кольорами',
            'cell'=>'K'
        ),
        array(
            'name'=>'Залишок загальний',
            'cell'=>'L'
        ),
        array(
            'name'=>'Дата',
            'cell'=>'M'
        ),
        array(
            'name'=>'Статус',
            'cell'=>'N'
        ),
        array(
            'name'=>'Коментар',
            'cell'=>'O'
        )
    );
    require_once ('Classes/PHPExcel.php');
    require_once('Classes/PHPExcel/Writer/Excel5.php');
    $phpexcel = new PHPExcel();
    $sheet = $phpexcel->getActiveSheet();
    for($i=0;$i<count($titles);$i++){
        $string = $titles[$i]['name'];
```

```
        //  $string = mb_convert_encoding($string,'UTF-8','Windows-
1251');
        $cellLetter = $titles[$i]['cell'].'2';
        $phpexcel->getActiveSheet()-
>setCellValueExplicit("$cellLetter",$string,PHPExcel_Cell_DataType
::TYPE_STRING);
        $phpexcel->getActiveSheet()->getStyle($cellLetter)-
>getFont()->setBold( true );
        $phpexcel->getActiveSheet()->getStyle($cellLetter)-
>getAlignment()->setWrapText(true);
        $phpexcel->getActiveSheet()->getStyle($cellLetter)-
>getAlignment()->applyFromArray(
            array(
                'horizontal' =>
PHPExcel_Style_Alignment::HORIZONTAL_CENTER,
                'vertical'   =>
PHPExcel_Style_Alignment::VERTICAL_CENTER,
                'rotation'   => 0,
                'wrap'       => TRUE
            )
        );
    }
    $i=3;
    $styleArray = array(
        'borders' => array(
            'allborders' => array(
                'style' => PHPExcel_Style_Border::BORDER_THIN

            )
        )
    );

    $phpexcel->getActiveSheet()->getColumnDimension('A')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('B')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('C')-
>setWidth(30);
    $phpexcel->getActiveSheet()->getColumnDimension('D')-
>setWidth(15);
    $phpexcel->getActiveSheet()->getColumnDimension('E')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('F')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('G')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('H')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('I')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('J')-
>setWidth(10);
    $phpexcel->getActiveSheet()->getColumnDimension('K')-
```

```
>setWidth(10);
  $phpexcel->getActiveSheet()->getColumnDimension('L')-
>setWidth(10);
  $phpexcel->getActiveSheet()->getColumnDimension('M')-
>setWidth(10);
  $phpexcel->getActiveSheet()->getColumnDimension('N')-
>setWidth(10);
  $phpexcel->getActiveSheet()->getColumnDimension('O')-
>setWidth(10);
  $page=$phpexcel->setActiveSheetIndex();

  $page->setTitle('Plan');
   $i = 3;
          while($row=$protokol->fetch()){
              $str1="Select * from planColor where
idPlan={$row['id']}";

              $color = $pdo->query($str1);
              $resColor=$color->fetchAll();
              $osnov='';
              $planCol='';
              $factCol='';
              $ostatC='';
              foreach($resColor as $item){
                $osnov=$osnov." -".$item['osn'];
                  $planCol=$planCol."\r\n - ".$item['planC'];
                  $factCol=$factCol."\r\n - ".$item['factColor'];
                  $ostatC=$ostatC."\r\n - ".$item['ostatC'];
              }

$phpexcel->getActiveSheet()-
>setCellValueExplicit("A{$i}",$row['nom'],PHPExcel_Cell_DataType::
TYPE_STRING);

              $phpexcel->getActiveSheet()-
>setCellValueExplicit("B{$i}",$row['igl'],PHPExcel_Cell_DataType::
TYPE_STRING);
              $phpexcel->getActiveSheet()-
>setCellValueExplicit("C{$i}",$row['art'],PHPExcel_Cell_DataType::
TYPE_STRING);
              $phpexcel->getActiveSheet()-
>setCellValueExplicit("D{$i}",$row['vid'],PHPExcel_Cell_DataType::
TYPE_STRING);
              $phpexcel->getActiveSheet()-
>setCellValueExplicit("E{$i}",$row['kColor'],PHPExcel_Cell_DataTyp
e::TYPE_STRING);
              $phpexcel->getActiveSheet()-
>setCellValueExplicit("F{$i}",$osnov,PHPExcel_Cell_DataType::TYPE_
STRING);
              $phpexcel->getActiveSheet()-
>setCellValueExplicit("G{$i}",$row['plan'],PHPExcel_Cell_DataType:
:TYPE_STRING);
              $phpexcel->getActiveSheet()-
```

```
>setCellValueExplicit("H{$i}",$planCol,PHPExcel_Cell_DataType::TYP
E_STRING);

                $phpexcel->getActiveSheet()-
>setCellValueExplicit("I{$i}",
$factCol,PHPExcel_Cell_DataType::TYPE_STRING);
                $phpexcel->getActiveSheet()-
>setCellValueExplicit("J{$i}",$row['fact'],PHPExcel_Cell_DataType:
:TYPE_STRING);
                $phpexcel->getActiveSheet()-
>setCellValueExplicit("K{$i}",$ostatC,PHPExcel_Cell_DataType::TYPE
_STRING);
                $phpexcel->getActiveSheet()-
>setCellValueExplicit("L{$i}",$row['ostat'],PHPExcel_Cell_DataType
::TYPE_STRING);
                $phpexcel->getActiveSheet()-
>setCellValueExplicit("M{$i}",$row['dat'],PHPExcel_Cell_DataType::
TYPE_STRING);
                $phpexcel->getActiveSheet()-
>setCellValueExplicit("N{$i}",$row['statys'],PHPExcel_Cell_DataTyp
e::TYPE_STRING);
                $phpexcel->getActiveSheet()-
>setCellValueExplicit("O{$i}",$row['comm'],PHPExcel_Cell_DataType:
:TYPE_STRING);


                $i++;


            }
            for($ii=2;$ii<$i;$ii++)  {
        $page->getStyle("A{$ii}")->applyFromArray($styleArray);

        $page->getStyle("B{$ii}")->applyFromArray($styleArray);
        $page->getStyle("C{$ii}")->applyFromArray($styleArray);
        $page->getStyle("D{$ii}")->applyFromArray($styleArray);
        $page->getStyle("E{$ii}")->applyFromArray($styleArray);
        $page->getStyle("F{$ii}")->applyFromArray($styleArray);
                $page->getStyle("F{$ii}")->getAlignment()-
>setWrapText(true);
        $page->getStyle("G{$ii}")->applyFromArray($styleArray);
        $page->getStyle("H{$ii}")->applyFromArray($styleArray);
         $page->getStyle("H{$ii}")->getAlignment()-
>setWrapText(true);
                $page->getStyle("I{$ii}")-
>applyFromArray($styleArray);
                $page->getStyle("I{$ii}")->getAlignment()-
>setWrapText(true);
                $page->getStyle("J{$ii}")-
>applyFromArray($styleArray);
                $page->getStyle("K{$ii}")-
>applyFromArray($styleArray);
                $page->getStyle("K{$ii}")->getAlignment()-
```

```
>setWrapText(true);
                $page->getStyle("L{$ii}")-
>applyFromArray($styleArray);
                $page->getStyle("M{$ii}")-
>applyFromArray($styleArray);
                $page->getStyle("N{$ii}")-
>applyFromArray($styleArray);
                $page->getStyle("O{$ii}")-
>applyFromArray($styleArray);
  }

$objWriter=PHPExcel_IOFactory::createWriter($phpexcel,'Excel2007')
;
  $filename='zvit.xlsx';
  if(file_exists($filename)){
      unlink($filename);
  }
  echo "Документ zvit.xlsx сформовано в поточній теці";
  $objWriter->save($filename);

      function file_force_download($file) {
          if (file_exists($file)) {
                // сбрасываем буфер вывода PHP, чтобы избежать
переполнения памяти выделенной под скрипт
                // если этого не сделать файл будет читаться в
память полностью!
                if (ob_get_level()) {
                    ob_end_clean();
                }
                // заставляем браузер показать окно сохранения файла
                header('Content-Description: File Transfer');
                header('Content-Type: application/octet-stream');
                header('Content-Disposition: attachment; filename='
. basename($file));
                header('Content-Transfer-Encoding: binary');
                header('Expires: 0');
                header('Cache-Control: must-revalidate');
                header('Pragma: public');
                header('Content-Length: ' . filesize($file));
                // читаем файл и отправляем его пользователю
                readfile($file);
                exit;
            }
      }
      echo "<br><a href='https://rssnews.pp.ua/admin'>Назад до
програми</a>";
      file_force_download("zvit.xlsx");
```

ДОДАТОК В

Код відправки листів.

```
  if(isset($_POST['email'])){
        include "PHPMailer/PHPMailerAutoload.php";
        function smtpmailer($to, $from, $from_name, $subject,
$body)
        {
            $mail = new PHPMailer();
            $mail->IsSMTP();
            $mail->SMTPAuth = true;
            $mail->CharSet = "utf-8";
            $mail->SMTPSecure = 'ssl';
            $mail->Host = 'uavip04.twincervers.net';
            $mail->Port = 465;
            $mail->Username = 'nosok@nosok.com.ua';
            $mail->Password = 'nosok';
            $mail->IsHTML(true);
            $mail->From="nosok@nosok.com.ua";
            $mail->FromName=$from_name;
            $mail->Sender=$from;
            $mail->AddReplyTo($from, $from_name);
            $mail->Subject = $subject;
            $mail->Body = $body;
            $mail->AddAddress($to);
            if(!$mail->Send())
            {
                $error ="Please try Later, Error Occured while
Processing...";
                return $error;
            }
            else
            {
                $error = "Thanks You !! Your email is sent.";
                return $error;
            }
        }
        $to    = $email;
        $from = 'nosok@nosok.com.ua';
        $from_name='nosok';
        $name = 'Art';
       $subj = "Зміна пароля";
        $datt=date('yy-m-d');
        $msg= "Доброго дня! Якщо ви хочете поміняти пароль на
сайті LOMANI!
     Для цього перейдіть за посиланням
https://nosok.pp.ua/changepass?email=".$_POST['email']." З
повагою, Адміністрація";
        $error=smtpmailer($to,$from, $name ,$subj, $msg);
        $res='Вам на пошту буде надіслано посилання для зміни
паролю';
    }

    else{
        $res='';
    }
```