

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
Кафедра кібернетики та інформатики

КВАЛІФІКАЦІЙНА РОБОТА

освітній ступінь - «Бакалавр»

на тему: **«РОЗРОБКА ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ**
ОЦІНЮВАННЯ ПРАЦЕЗДАТНОСТІ
КОМП'ЮТЕРИЗОВАНИХ РОБОЧИХ МІСЦЬ»

Виконав: студент спеціальності
126 «Інформаційні системи та технології»
ОП «Інформаційні системи та технології»
Дубінчин Володимир Сергійович

Науковий керівник: **Руденко Юлія Олександрівна**
доцент кафедри кібернетики та інформатики

Рецензент: **Дегтярьова Неля Валентинівна**
доцент кафедри інформатики Сумського державного
педагогічного університету

Суми – 2024

СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет	Економіки і менеджменту
Кафедра	Кібернетики та інформатики
Освітній ступінь	«Бакалавр»
Спеціальність	126 «Інформаційні системи та технології» ОП «Інформаційні системи та технології»

Затверджую:
Завідувач кафедри _____ Світлана АГАДЖАНОВА
« ____ » « _____ » 2023 ____ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Дубінчину Володимиру Сергійовичу

1. Тема роботи: Розробка інформаційної підсистеми оцінювання працездатності комп'ютеризованих робочих місць

(на прикладі Управління Державної казначейської служби України у Середино-Будському районі Сумської області).

2. База практичного дослідження: Сумський НАУ.

керівник роботи Руденко Юлія Олександрівна

затверджена наказом по університету від « ____ » _____ 2023 р. № ____ -н

3. Строк подання студентом закінченого проекту (роботи) «9» червня 2024 року

4. Вихідні дані до проекту (роботи):

Результати аналітичних досліджень зарубіжних та вітчизняних вчених, нормативні документи та чинне законодавство, Інтернет-джерела.

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- виконати постановку задачі;
- вибрати середовище для реалізації програмного додатку;
- провести аналіз наявних рішень для вирішення проблеми;
- розробити проект;
- розробити програмне забезпечення проекрованої інформаційної системи;
- розробити програмний продукт .

6. Дата видачі завдання: « ____ » _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту(роботи)	Строк виконання етапів проекту	Примітка
1	Визначення об'єкту, предмету дослідження, формулювання мети та задач кваліфікаційної роботи, складання плану	січень 2024 р.	виконано
2	Підбір та вивчення літературних джерел	січень 2024 р.	виконано
3	Узагальнення теоретичного матеріалу з обраної теми дослідження та представлення чорнового варіанту першого розділу кваліфікаційної роботи	січень 2024 р.	виконано
4	Збір та обробка фактичного матеріалу, узагальнення аналізу застосування досліджуваного питання в установі (переддипломна практика)	січень - березень 2024 р.	виконано
5	Оформлення теоретичної частини кваліфікаційної роботи, узагальнення аналітичної частини, захист звіту по переддипломній практиці	березень 2024 р.	виконано
6	Оформлення практичної частини кваліфікаційної роботи, розробка проєкту інформаційної підсистеми.	квітень 2024 р.	виконано
7	Підготовка висновків, пропозицій та узгодження з керівником	травень 2024 р.	виконано
8	Підготовка кінцевого варіанту кваліфікаційної роботи	до 3 червня 2024 р.	виконано
9	Опрацювання зауважень керівника, перевірка на автентичність	до 6 червня 2024 р.	виконано
10	Перевірка з академічної доброчесності кваліфікаційної роботи	9 червня 2024 р..	виконано
11	Представлення кваліфікаційної роботи на кафедру, деканат	23 червня 2024 р..	виконано
12	Підготовка доповіді, презентації до кваліфікаційної роботи та її попередній захист	до 27 червня 2024 р..	виконано
13	Захист кваліфікаційної роботи	_ червня 2024 р.	виконано

Здобувач вищої освіти _____

Керівник роботи _____

Перевірку на автентичність проведено _____

Дипломна робота допущена до захисту _____

АНОТАЦІЯ

Дубінчин В.С. Розробка інформаційної підсистеми оцінювання працездатності комп'ютеризованих робочих місць (на прикладі Управління Державної казначейської служби України у Середино-Будському районі Сумської області).

Кваліфікаційна робота зі спеціальності 126 «Інформаційні системи та технології» ОП Інформаційні системи та технології, СНАУ, Суми-2024 р. – Рукопис.

Кваліфікаційна робота присвячена дослідженню актуальних питань, пов'язаних з розробкою інформаційної підсистеми оцінювання працездатності комп'ютеризованих робочих місць (на прикладі Управління Державної казначейської служби України у Середино-Будському районі Сумської області).

Актуальність теми обумовлена зростанням кількості використовуваної обчислювальної техніки і програмних засобів в роботі організації і необхідністю підтримки їх в працездатному стані задля запобіганню простоїв в роботі, ефективного використання як комп'ютерних систем, так і трудових ресурсів, уникненню загроз безпеці інформації, пов'язаних з несанкціонованим доступом до даних.

Розглянуто фактори, що прямо або опосередковано можуть впливати на працездатність комп'ютеризованих робочих місць,

Основними завданнями дослідження є вивчення та аналіз існуючих рішень, розробка та програмна реалізація інформаційної підсистеми оцінювання працездатності комп'ютеризованих робочих місць технології.

Результати дослідження сприятимуть покращенню комфорту працівників і підвищенню продуктивності їх роботи, підвищенню ефективності роботи організації.

Розроблена інформаційна підсистема може бути використана організаціями для підвищення надійності і стабільності роботи парку

комп'ютерної техніки та зменшення кількості її простоїв, підвищення рівня безпеки даних, своєчасному реагуванню на виникнення проблем з апаратним і програмним забезпеченням комп'ютеризованих робочих місць організації та запобіганню їм.

Ключові слова: комп'ютеризоване робоче місце, PHP, Yii2, MySQL, Python.

Зміст

ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ ДОСЛІДЖУВАНОЇ ТЕМАТИКИ	9
1.1 Поняття інформаційних систем для оцінювання працездатності комп'ютеризованих робочих місць	9
1.2 Огляд існуючих підходів до оцінювання працездатності	14
1.3. Аналіз потреб та вимог Державної казначейської служби України у Середино-Будському районі	18
1.4. Визначення основних завдань	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ	22
2.1 Визначення функціональних вимог	22
2.2 Архітектура інформаційної підсистеми	24
2.3 Вибір технологій та інструментів розробки	25
2.3.1. Вибір СУБД	25
2.3.2 Вибір інструментальних засобів розробки	28
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ.....	37
3.1 Розробка бази даних.....	37
3.2 Створення програмного коду	40
3.3 Впровадження інформаційної підсистеми в роботу.....	53
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
ДОДАТКИ.....	60
Додаток А.....	60
Додаток Б	62
Додаток В.....	65
Додаток Г	67
Додаток Д.....	70
Додаток Є.....	71
Додаток Ж.....	72
Додаток З.....	73
Додаток І	74

ВСТУП

Впровадження і використання інформаційно-комунікаційних технологій в усіх сферах діяльності людини докорінно змінює простір існування людини. Їх використання у фінансовій сфері, сфері послуг, виробничій діяльності тощо, дозволяє прискорити всі бізнес-процеси. При цьому швидкими темпами зростає парк цифрових пристроїв і, відповідно, обсяг програмного забезпечення, що використовується суб'єктами господарювання. Водночас, для ефективного використання комп'ютерної техніки і програмного забезпечення, підприємства і організації потребують підтримки належного стану комп'ютерної техніки і програмного забезпечення, оскільки виникнення проблем як з апаратним забезпеченням, так і з програмним забезпеченням, можуть привести до втрати або пошкодження даних, порушення їх цілісності, простоїв в роботі. Для запобігання виникнення проблем з апаратним та/або програмним забезпеченням комп'ютеризованих робочих місць або вчасного на них реагування і швидкого усунення, необхідно здійснювати постійний моніторинг апаратного і програмного забезпечення.

Таким чином, актуальність роботи зумовлена необхідністю підтримки належного стану комп'ютерної техніки і програмного забезпечення та додаткового контролю за використанням ресурсів кожного комп'ютеризованого робочого місця, що в кінцевому рахунку впливає як на ефективність використання матеріально-технічних і кадрових ресурсів організації, так і ефективність роботи самої організації.

Об'єкт дослідження – теоретичні та практичні аспекти створення комп'ютеризованих робочих місць.

Предметом дослідження є комп'ютеризовані робочі місця працівників Управління Державної казначейської служби України у Середино-Будському районі Сумської області.

Метою роботи є розробка інформаційної підсистеми оцінювання працездатності комп'ютеризованих робочих місць працівників Управління Державної казначейської служби України у Середино-Будському районі Сумської області.

Для досягнення мети роботи необхідно вирішити наступні *задачі*:

- визначити фактори, що впливають на працездатність комп'ютеризованих робочих місць;
- розробити інформаційну систему, що здійснює моніторинг та оцінювання працездатності комп'ютеризованих робочих місць;
- розробити програмне забезпечення інформаційної системи.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ ДОСЛІДЖУВАНОЇ ТЕМАТИКИ

1.1 Поняття інформаційних систем для оцінювання працездатності комп'ютеризованих робочих місць

Докорінні зміни в усіх сферах життєдіяльності людини, викликані широким впровадженням і застосуванням інформаційно-комунікаційних технологій, не залишили в стороні і Державну казначейську службу України. З огляду на трансформаційні процеси розвитку казначейського обслуговування бюджетних коштів актуальними є питання створення нових та вдосконалення існуючих інструментів стандартизації та оптимізації передових інформаційних технологій, спрямовані на модернізацію та уніфікацію казначейських процедур. Застосування сучасних інформаційно-комунікаційних технологій та комп'ютерних систем відкриває перед казначейською службою широкі горизонти для підвищення якості фінансових послуг та зростання ефективності роботи казначейської системи з обслуговування бюджетів усіх рівнів [1].

Відповідно, цифровізація роботи будь-якого підприємства взагалі, і казначейської служби зокрема, вимагає використання комп'ютерної техніки, широкого спектру периферійних пристроїв та необхідного програмного забезпечення на робочих місцях працівників, що в свою чергу, покращує умови їх праці, але, разом з тим, вимагає від працівників опановувати, поряд з професійними, нові навички в сфері використання інформаційно-комунікаційних технологій. Слід зауважити, що кількість комп'ютерів і спектр використовуваного програмного забезпечення постійно зростає.

Водночас, ефективне використання сучасних інформаційно-комунікаційних технологій неможливе без ефективного використання кадрового потенціалу підприємства [2].

Таким чином, цифровізація роботи підприємства спирається, в першу чергу, на два взаємопов'язані і взаємозалежні фактори – впровадження і застосування інформаційно-комунікаційних технологій в свою діяльність і ефективне використання людських ресурсів.

Ефективне використання людських ресурсів на підприємстві включає в себе ряд аспектів, що сприяють досягненню цілей організації та мають на меті зростання її конкурентоспроможності. До основних принципів ефективного використання кадрового потенціалу відноситься:

1. Правильна організація робочого простору. Цей принцип має на меті покращення загального комфорту працівників, що, в свою чергу, позитивно впливає на їхню продуктивність.
2. Відповідність виконуваним обов'язкам. Працівники повинні мати необхідний набір знань, навичок та досвіду для виконання своїх обов'язків.
3. Мотивація персоналу. Система мотивації повинна стимулювати працівників до досягнення високих результатів.
4. Комунікація. Ефективна комунікація сприяє розумінню працівниками цілей і завдань організації, а також сприяє вирішенню конфліктів та підвищенню рівня кооперації працівників.
5. Розвиток. Підприємство повинно інвестувати у розвиток своїх працівників задля вдосконалення їх навичок та знань, набуття нового досвіду.
6. Ефективний менеджмент. Ефективні менеджери можуть надихати та мотивувати свою команду до досягнення великих результатів.
7. Адаптивність. Компанія повинна бути готовою до змін і адаптації до нових умов, що дозволяє підтримувати високу ефективність в умовах ринкової нестабільності [3].

Досліджувана в роботі тематика стосується правильної організації робочого простору комп'ютеризованих робочих місць працівників казначейства в частині апаратного і програмного забезпечення робочого місця. Зазвичай працездатність робочого місця оцінюється за стандартними критеріями, а саме, ергономікою, освітленням, мікрокліматом, шумом і

вібрацією, безпекою та зручністю [4]. Разом з тим, коли мова йде про комп'ютеризовані робочі місця, цей перелік вимагає врахування і специфічних критеріїв таки, як наявне комп'ютерне обладнання і периферійні пристрої, наявні необхідні програмні засоби та забезпечення безпеки і захисту даних в системі. Водночас, посадові обов'язки, наприклад, бухгалтера або фахівця планово-фінансового відділу, не передбачають наявності навичок встановлення програмного забезпечення або аналіз використання пристроїв комп'ютера з метою оптимізації роботи системи. Відповідно, ці функції мають виконувати фахівці IT-відділу.

Комп'ютерне обладнання, підключені периферійні пристрої та встановлене програмне забезпечення мають забезпечувати виконання всіх функціональних обов'язків працівника. Окрім того, пристрої самого комп'ютера мають забезпечити прийнятну швидкість виконання всіх процесів і задач для встановленого програмного забезпечення, інакше кажучи, пристрої комп'ютера повинні відповідати вимогам до апаратного забезпечення, що зазначається виробниками програмного забезпечення на своїх продуктах, оскільки це має суттєвий вплив на ефективність і продуктивність роботи [5].

Ще одним важливим фактором забезпечення працездатності комп'ютеризованих робочих місць є безпека і захист даних. Для забезпечення безпеки і захисту даних мають застосовуватися комплексні заходи, оскільки існує багато шляхів, що можуть привести до крадіжки даних, порушення їх цілісності або втрати, як через навмисні злочинні дії, так і ненавмисні дії користувачів, що не мають достатніх знань в цьому аспекті. Відповідно, зазначені заходи мають включати (рис.1.1.):

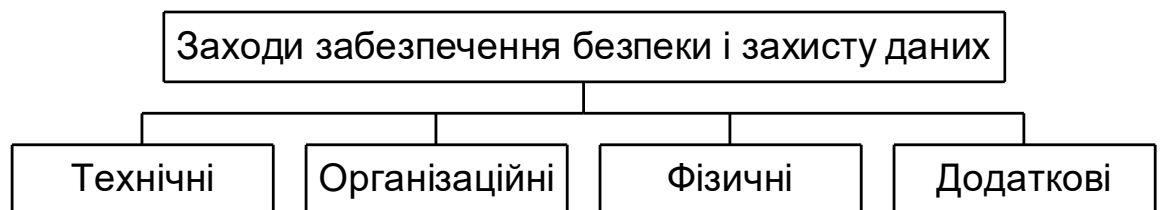


Рисунок 1.1 - Заходи забезпечення безпеки і захисту даних

Джерело: [6]

I. Технічні заходи:

1. Встановлення антивірусного програмного забезпечення та його регулярне оновлення з метою захисту від вірусів та шкідливих програм.
2. Контроль та фільтрація вхідного та вихідного трафіку за допомогою апаратних та програмних фаєрволів.
3. Використання шифрування даних як під час зберігання, так і під час передачі.
4. Регулярне створення резервних копій даних для відновлення у разі їх втрати або пошкодження.
5. Регулярне оновлення операційної системи та програмного забезпечення для усунення вразливостей та захисту від нових загроз.
6. Налаштування прав доступу до файлів і папок для обмеження доступу лише для авторизованих користувачів.

II. Організаційні заходи:

1. Розробка та впровадження політик інформаційної безпеки, що регламентують використання комп'ютерів та обробку даних.
2. Проведення регулярних тренінгів для працівників з метою підвищення обізнаності про ризики інформаційної безпеки та способи їх уникнення.
3. Вимога використовувати складні паролі та регулярна їх зміна.
4. Проведення регулярних аудитів інформаційної безпеки та моніторинг систем для виявлення і реагування на загрози.

III. Фізичні заходи:

1. Контроль фізичного доступу задля обмеження фізичного доступу до комп'ютерів та серверів як сторонніх осіб, так і співробітників, що не мають відповідної підготовки або не пройшли відповідне навчання.
2. Забезпечення належного захисту комп'ютерного обладнання від крадіжки або пошкодження.

IV. Додаткові заходи

1. Розробка та впровадження плану дій для швидкого відновлення систем та даних у разі аварії або кібер-атаки.

2. Використання багатофакторної автентифікації (MFA) для підвищення безпеки доступу до критичних систем і даних [6].

Впровадження, виконання та/або контроль за виконанням цих заходів має здійснюватися виключно фахівцями IT-відділу, які, задля цього, мають постійно тримати на контролі стан кожного комп'ютеризованого робочого місця.

З метою здійснення постійного контролю за станом комп'ютеризованих робочих місць має бути забезпечений постійний моніторинг апаратного і програмного забезпечення місця, який дозволить своєчасно реагувати на зміни в роботі системи і вживати відповідних заходів для попередження порушень в роботі системи або усунення наслідків таких порушень.

Серед факторів, що можуть свідчити про порушення в роботі системи комп'ютеризованого робочого місця або можливі порушення в майбутньому, можна виділити:

- несанкціоноване встановлення програмного забезпечення, в т.ч. і піратського, оскільки таке програмне забезпечення може містити шкідливі програми [7];
- відсутність останніх оновлень програмного забезпечення, в т.ч. і операційної системи, оскільки застарілі системи виявляються, в першу чергу, більш вразливими до загроз безпеці [8, 9];
- недостатній обсяг вільного місця на зовнішніх накопичувачах, як правило, менше 10% від загального обсягу носія, оскільки це може уповільнювати роботу системи загалом, унеможливлювати встановлення нового програмного забезпечення або, навіть, його оновлень, тощо;
- заміна пристроїв комп'ютера.

Таким чином, моніторинг наведених вище факторів дозволить фахівцям IT-відділу швидко і своєчасно оцінювати працездатність комп'ютеризованих

робочих місць організації та, за необхідності, своєчасно реагувати і приймати відповідні рішення у разі виявлених змін. Це має суттєве значення для роботи підприємства, оскільки дозволяє підтримувати комп'ютерну техніку та програмне забезпечення в належному стані та контролювати своєчасність і актуальність їх оновлення, запобігає нерегламентованому їх використанню, є додатковим заходом забезпечення захисту даних.

1.2 Огляд існуючих підходів до оцінювання працездатності

Оцінка працездатності комп'ютеризованих робочих місць вимагає врахування багатьох різнопланових факторів, що стосуються різних напрямків і суб'єктів, задіяних в досліджуваній сфері. Наприклад, оцінка безпеки і захисту даних вимагає врахування як наявності технічних заходів, що стосуються обладнання, так і організаційних заходів, які, в свою чергу стосуються працівників.

Враховуючи специфічність оцінки, загальновідомих комплексних рішень та програмних засобів не існує. Для продуктових ІТ-компаній ринок для реалізації таких рішень замалий, а з врахуванням рівня фінансування підприємств і організацій державного сектору, взагалі видається безперспективним. Відповідно, знайти готове рішення, що формує комплексну інтегральну оцінку комп'ютеризованих робочих місць або здійснює постійний моніторинг і аналіз стану комп'ютерних систем, видається малоімовірним.

На сьогоднішній день існують рішення для аналізу конфігурації та моніторингу ПК, що дозволяють здійснювати збір та відображення технічних даних про комп'ютер, його апаратні засоби, встановлене програмне забезпечення тощо. Серед найбільш розповсюджених:

1. CPU-Z – безкоштовна утиліта для відображення параметрів роботи обладнання комп'ютера та його характеристик. CPU-Z працює під всіма версіями операційних систем Microsoft Windows, включаючи Windows 95 і

Windows 11. Також наявна версія для мобільних пристроїв під Android. Вікно програми CPU-Z наведено на рисунку 1.2.

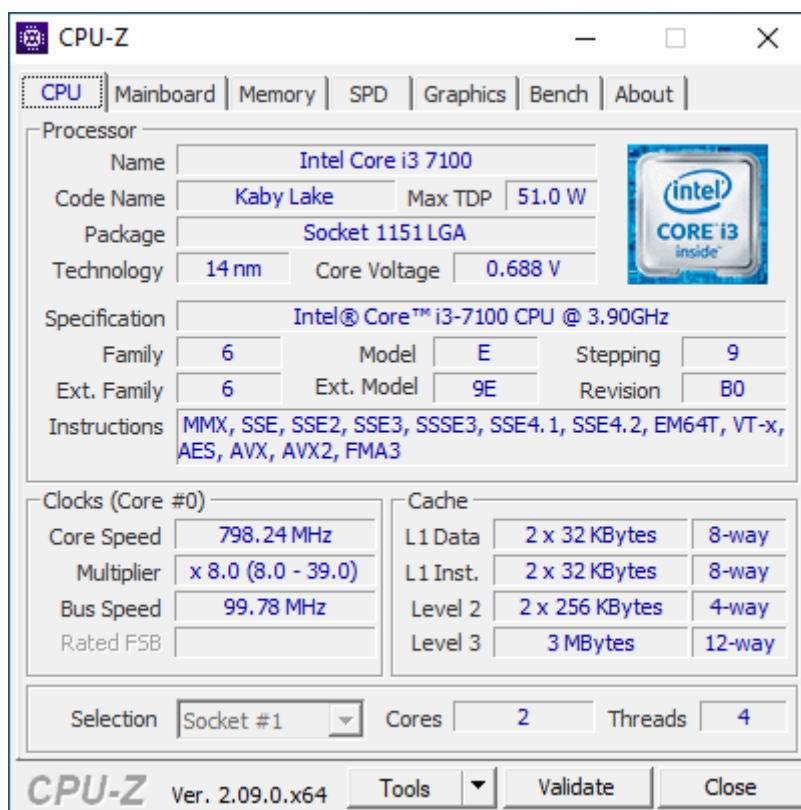


Рисунок 1.2 - Вікно програми CPU-Z

Джерело: Запропоновано автором

2. AIDA64 – безкоштовна програма-утиліта від виробника FinalWire Ltd. для тестування складових компонентів персонального комп'ютера та відображення його характеристик, в т.ч. і про програмне забезпечення. Програма випускається тільки під Windows. Вікно програми AIDA64 наведено на рисунку 1.3.

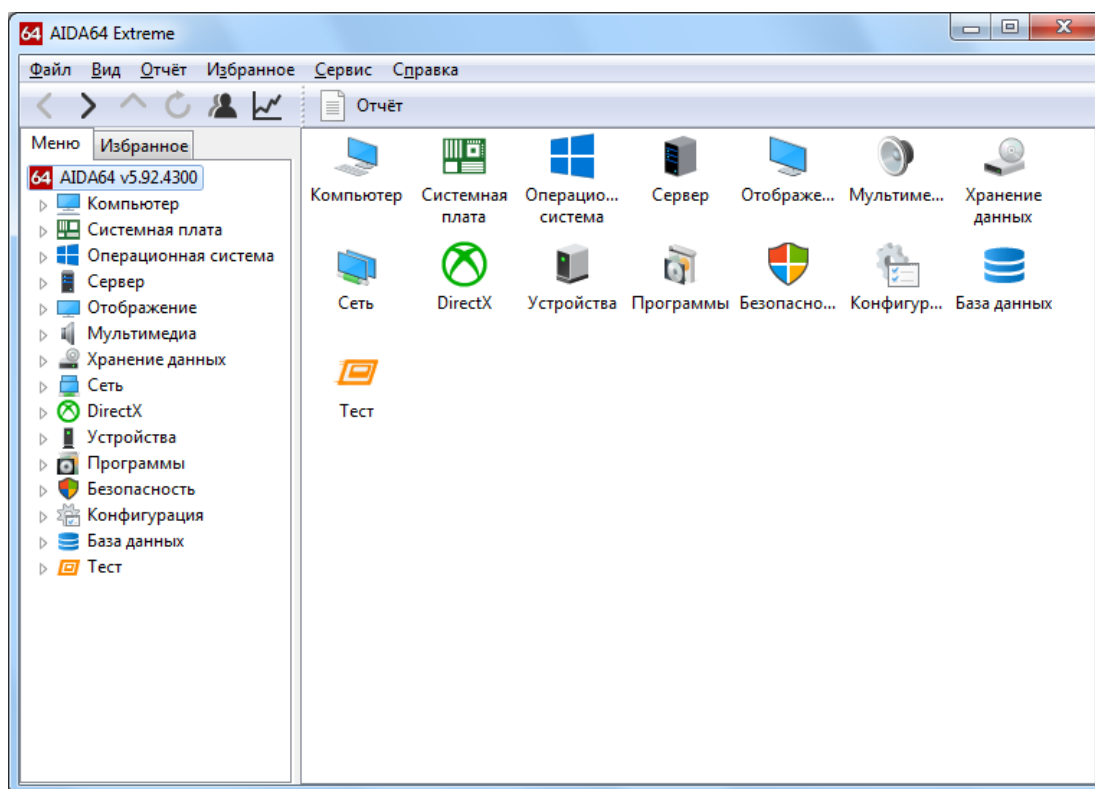


Рисунок 1.3 - Вікно програми AIDA64

Джерело: Запропоновано автором

3. HWiNFO (Hardware Info) є потужним інструментом для моніторингу та/або діагностики стану апаратного забезпечення комп'ютера. Цей інструмент доступний у двох версіях: HWiNFO32 для 32-бітових операційних систем та HWiNFO64 для 64-бітних систем. Він дозволяє отримати детальну інформацію про всі компоненти системи, включаючи процесор, оперативну і зовнішню пам'ять, материнську плату, відеокарти, мережеві пристрої та інші. Вікно програми HWiNFO наведено на рисунку 1.4.

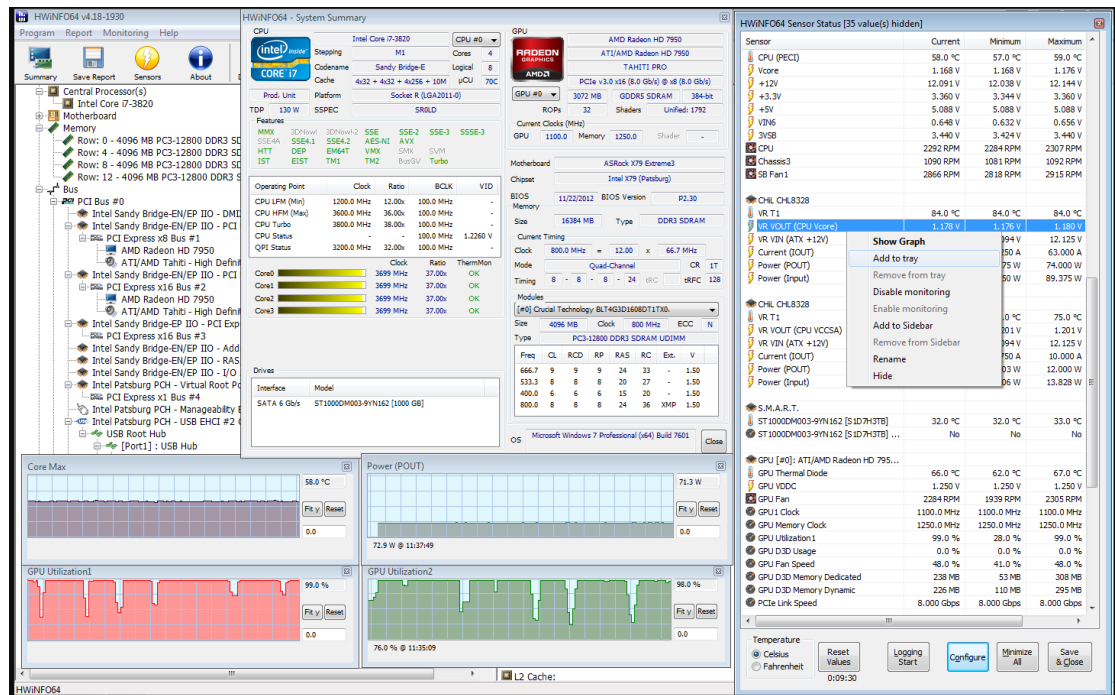


Рисунок 1.4 - Вікно програми HWiNFO

Джерело: Запропоновано автором

Перелік таких засобів можна продовжувати до нескінченості. Їх велика кількість обумовлена простотою програмної реалізації та тим, що більшість популярних мов програмування мають в своєму складі засоби, як з «коробки», так і у вигляді додаткових пакетів, для реалізації функціоналу зазначених додатків.

Але проблема їх використання полягає в тому, що відсутня можливість їх інтеграції в проєктовану інформаційну систему з метою використання даних для комплексної оцінки працездатності комп'ютеризованого робочого місця, оскільки в багатьох з них навіть не передбачено виведення інформації в файл, а, якщо і передбачено, то це має бути здійснено безпосередньо користувачем системи і безпосередньо з робочого місця. Водночас, лише невелика кількість даних з цих додатків потрібні при реалізації системи.

1.3. Аналіз потреб та вимог Державної казначейської служби України у Середино-Будському районі

Управління Державної казначейської служби України у Середино-Будському районі Сумської області є ключовим органом, що відповідає за ефективне управління фінансовими ресурсами держави на місцевому рівні та здійснює контроль за їх використанням. Управління забезпечує проведення фінансових операцій, контроль за виконанням бюджету, а також здійснює координацію та підтримку роботи фінансових установ та організацій району. Управління відповідає за виконання різноманітних завдань, включаючи контроль використання бюджетних коштів, ведення обліку фінансових операцій, а також надання консультацій та підтримки організаціям та громадянам у сфері фінансового управління.

Управління працює згідно з принципами відкритості та транспарентності, забезпечуючи доступ до даних про фінансові операції та діяльність управління для громадськості та зацікавлених сторін. Управління Державної казначейської служби у Середино-Будському районі Сумської області відіграє важливу роль у розвитку району, забезпечуючи його фінансову стабільність, сприяє ефективному управлінню фінансовими ресурсами та підтримці фінансової дисципліни.

Персонал управління складають висококваліфіковані фахівці в області переважно фінансів та бухгалтерського обліку. Їх робота спрямована на забезпечення ефективного управління фінансовими ресурсами та дотримання вимог законодавства. Оскільки управління активно впроваджує і використовує інформаційно-комунікаційні технології та відповідне програмне забезпечення як загального призначення, так і спеціалізоване, для автоматизації, зокрема, бухгалтерського обліку та контролю за фінансовими операціями, то для забезпечення роботи всієї ІТ-інфраструктури управління, в т.ч. і забезпечення безпеки даних, в структурі Управління було створено ІТ-відділ.

З огляду на безпеку і захист даних та з метою ефективного використання апаратного і програмного забезпечення, управлінням були розроблені і встановлені правила, згідно яких на комп'ютерах управління дозволено використовувати лише програмне забезпечення з переліку, складеного фахівцями IT-відділу. Також діє заборона на встановлення програмного забезпечення для всіх фахівців, окрім фахівців IT-відділу, що, в свою чергу дозволяє уникати загроз для даних і використання комп'ютерної техніки для виконання сторонніх задач.

До переліку дозволеного для використання на комп'ютеризованих робочих місцях програмного забезпечення, серед інших, відносяться:

- операційні системи сімейства Windows версій 7, 10 та 11;
- офісні пакети MS-Office, Open Office, Libre Office різних версій;
- антивірусне програмне забезпечення Зілля, McAfee, Avast Antivirus та ін.;
- програмний комплекс «Фіндокументи»;
- хмарна версія бухгалтерської програми QuickBooks;
- та інші.

1.4. Визначення основних завдань

Виходячи з викладеного вище, можна зробити висновок, що задля підтримки працездатності і ефективного використання комп'ютеризованого робочого місця варто враховувати цілу низку факторів, а саме:

1. Необхідність моніторингу стану обладнання, що включає відстеження стану ключових компонентів комп'ютера, таких як процесор, оперативна пам'ять, жорсткий диск тощо, а також контроль за станом операційної системи, встановлених програм та їх своєчасного оновлення.

2. Необхідність моніторингу використання ресурсів задля відстеження використання дискового простору та інших ресурсів.

3. Необхідність збору і аналізу даних використання апаратних і програмних ресурсів з метою виявлення закономірностей і планування оновлення систем або їх заміни.

4. Необхідність регулярного технічного обслуговування для забезпечення безперервної працездатності системи.

5. Необхідність відстеження потенційних загроз і вразливостей у системі, забезпечення захисту даних від несанкціонованого доступу та зловмисних дій.

Безумовно, забезпечувати врахування цих факторів у роботі підприємства мають ІТ-фахівці при цьому, бажано, не відволікаючи інших фахівців від виконання своїх посадових обов'язків на своїх робочих місцях. Для цього ІТ-фахівцям необхідно забезпечити постійний доступ до інформації про стан системи, що дозволить їм:

- контролювати стан апаратного і програмного забезпечення кожного комп'ютеризованого робочого місця організації;
- своєчасно реагувати на проблеми, що виникають при експлуатації комп'ютерної техніки та використанні програмного забезпечення;

Таким чином, доцільно розробити інформаційну систему оцінювання працездатності комп'ютеризованих робочих місць, що забезпечить виконання таких завдань:

1. Моніторинг стану апаратного забезпечення комп'ютеризованого робочого місця.
2. Моніторинг стану програмного забезпечення комп'ютеризованого робочого місця.
3. Моніторинг використання апаратних і програмних ресурсів комп'ютеризованого робочого місця.
4. Генерація звітів про стан системи комп'ютеризованого робочого місця.
5. Забезпечення заходів збереження і захисту даних.
6. Аналіз ефективності робочих місць.

Очікувані результати:

1. Підвищення ефективності роботи.
2. Покращення продуктивності користувачів.
3. Зниження витрат на технічне обслуговування.
4. Підвищення надійності та стабільності системи.
5. Поліпшення безпеки даних.
6. Зручність використання та прийняття рішень.
7. Підвищення комфортних умов для користувачів.
8. Зменшення кількості простоїв.

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ

2.1 Визначення функціональних вимог

Для розробки програмного забезпечення інформаційної підсистеми оцінювання працездатності комп'ютеризованих робочих місць, необхідно визначити функціональні вимоги безпосередньо до програмного забезпечення на підставі сформульованих завдань до інформаційної підсистеми, визначених в попередньому пункті роботи.

Перш за все, застосунок має забезпечувати моніторинг апаратного і програмного забезпечення кожного комп'ютеризованого робочого місця. При цьому моніторинг має відбуватись без втручання фахівця, що працює на комп'ютеризованому робочому місці. Окрім того, моніторинг не має відволікати фахівця від виконання своїх посадових обов'язків. Таким чином, застосунок, що здійснює збір інформації, має працювати в фоновому режимі і мати незначний обсяг, щоб додатково не навантажувати систему. Збір інформації про комп'ютеризоване робоче місце має відбуватись щогодини протягом робочого часу

Для виконання поставлених завдань фахівцям ІТ-відділу необхідно мати інформація про апаратне і програмне забезпечення кожного комп'ютеризованого робочого місця, а саме:

- процесор;
- обсяг оперативної пам'яті;
- повний і доступний обсяг зовнішніх запам'ятовуючих пристроїв;
- операційна система та її версія;
- встановлені програми та їх версії;

Зібрана інформація про програмне і апаратне забезпечення комп'ютеризованого робочого місця має бути збережена з міткою часу в базі даних і у файлі на пристрої на випадок, якщо база даних недоступна.

Таким чином, проєктована система має забезпечувати виконання таких функцій:

1. Авторизований доступ до функціоналу програмного забезпечення.
2. Віддалений доступ до даних про комп'ютеризовані робочі місця.
3. Створення і редагування переліку комп'ютеризованих робочих місць організації: робоче місце, розташування, відповідальний працівник, встановлена операційна система та її версія, встановлене програмне забезпечення, апаратне забезпечення.
4. Створення і редагування переліку дозволеного програмного забезпечення: тип програмного забезпечення, назва, остання версія, інформація про виробника.
5. Моніторинг апаратного забезпечення комп'ютеризованого робочого місця: процесор, обсяг оперативної пам'яті, повний і використаний або доступний обсяг зовнішньої пам'яті, файлова система пристроїв зовнішньої пам'яті, мережеве підключення.
6. Моніторинг програмного забезпечення комп'ютеризованого робочого місця: встановлена операційна система та її версія, кількість встановлених програм, версія встановленої програмної системи, відповідність переліку дозволених до використання програмних засобів.
7. Збереження результатів моніторингу з міткою часу в центральному сховищі, а у разі відсутності зв'язку з ним, на пристрої комп'ютеризованого робочого місця.
8. Постійний доступ фахівців ІТ-відділу до результатів моніторингу апаратного і програмного забезпечення кожного комп'ютеризованого робочого місця.
9. Формування звітів про моніторинг апаратного і програмного забезпечення кожного комп'ютеризованого робочого місця.
10. Формування звітів про застарілі версії програмного забезпечення комп'ютеризованих робочих місць на поточну дату і необхідність його оновлення.

11. Формування повідомлень про зміни апаратного та/або програмного забезпечення комп'ютеризованого робочого місця.

2.2 Архітектура інформаційної підсистеми

Програмна реалізація проєктованої інформаційної системи оцінювання працездатності комп'ютеризованих робочих місць складається з трьох структурних частин:

1. База даних.
2. Застосунок збору даних про комп'ютеризоване робоче місце.
3. Веб-застосунок.

Зв'язок між структурними частинами програмної реалізації інформаційної системи оцінювання працездатності комп'ютеризованих робочих місць наведений на рисунку 2.1.

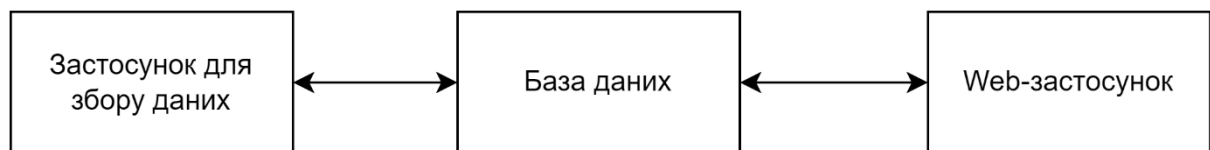


Рисунок 2.1 - Зв'язок між структурними частинами інформаційної системи оцінювання працездатності комп'ютеризованих робочих місць

Джерело: Запропоновано автором

База даних використовується як централізоване сховище для зберігання даних про комп'ютеризовані робочі місця, оскільки вона є зручним засобом зберігання і маніпулювання даними і дозволяє відокремити програмний код від даних. Окрім того, база даних дозволяє і віддалений доступ до даних, що необхідно для зберігання даних про всі комп'ютеризовані робочі місця та вкрай важливо для забезпечення неперервної роботи фахівця на комп'ютеризованому робочому місці з одночасним моніторингом системи цього місця фахівцями IT-відділу.

Застосунок збору даних про комп'ютеризоване робоче місце встановлюється на кожне комп'ютеризоване робоче місце, здійснює

регулярний збір даних про нього та здійснює відповідні записи в базу даних. Застосунок порівнює апаратне забезпечення комп'ютеризованого робочого місця з даними в про місце в базі даних і у разі виникнення розходжень, здійснює запис в базу даних про виявлений інцидент. Аналогічне порівняння відбувається і з встановленим програмним забезпеченням. Якщо з деяких причин база даних недоступна, наприклад, через проблеми з мережевим підключенням або проблеми з веб-сервером, на якому вона розміщена, доцільно дані зберігати на пристроях комп'ютеризованого робочого місця в json-форматі з можливістю запису збережених даних в базу після відновлення підключення.

Веб-застосунок забезпечує доступ до бази даних, в т.ч. і віддалений, та формування необхідних звітів про працездатність комп'ютеризованих робочих місць. Однією з основних переваг використання веб-застосунків є те, що його скрипти розміщуються на веб-сервері і немає необхідності розміщувати програмні файли на кожному робочому місці. Доступ до функціональних можливостей веб-застосунку відбувається, як правило, через браузер.

2.3 Вибір технологій та інструментів розробки

2.3.1. Вибір СУБД

Бази даних є зручним засобом зберігання і маніпулювання даними. Водночас, для виконання цих функцій потрібне відповідне програмне забезпечення. Взаємодія програмних застосунків з базою даних відбувається здебільшого через сервер баз даних. Управління структурою бази даних, доступ і маніпулювання даними, здійснюється через системи управління базами даних (СУБД) [10].

Інструментарій СУБД, в свою чергу, забезпечує:

- створення і редагування відповідних структур для зберігання даних;

- контроль цілісності даних;
- розмежування прав користувачів;
- підтримка багатокористувацького інтерфейсу з метою забезпечення спільної роботи з даними;
- захист організаційної структури даних і, безпосередньо, самих даних;
- створення бекапів організаційних структур і даних [10].

Сучасний ринок СУБД пропонує широкий спектр рішень, здатних задовольнити найвибагливішого розробника.

На рисунку 2.2 наведено десять перших за популярністю СУБД з рейтингу австрійської консалтингової ІТ-компанії Solid-IT DB-Engines (дані на травень 2024 р.). Всього в рейтингу наявні 420 СУБД [11].

420 systems in ranking, May 2024

Rank			DBMS	Database Model	Score		
May 2024	Apr 2024	May 2023			May 2024	Apr 2024	May 2023
1.	1.	1.	Oracle +	Relational, Multi-model	1236.29	+2.02	+3.66
2.	2.	2.	MySQL +	Relational, Multi-model	1083.74	-3.99	-88.72
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model	824.29	-5.50	-95.80
4.	4.	4.	PostgreSQL +	Relational, Multi-model	645.54	+0.49	+27.64
5.	5.	5.	MongoDB +	Document, Multi-model	421.65	-2.31	-14.96
6.	6.	6.	Redis +	Key-value, Multi-model	157.80	+1.36	-10.33
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	135.35	+0.57	-6.28
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model	128.46	+0.97	-14.56
9.	9.	↑ 11.	Snowflake +	Relational	121.33	-1.87	+9.61
10.	10.	↓ 9.	SQLite +	Relational	114.32	-1.69	-19.54

Рисунок 2.2 - Лідери рейтингу СУБД від Solid-IT DB-Engines у травні 2024 р.

Джерело: [11]

Схожі показники дає щомісячний рейтинг СУБД за версією компанії Statista (дані на початок 2024 р.), що відображено на рисунку 2.3.

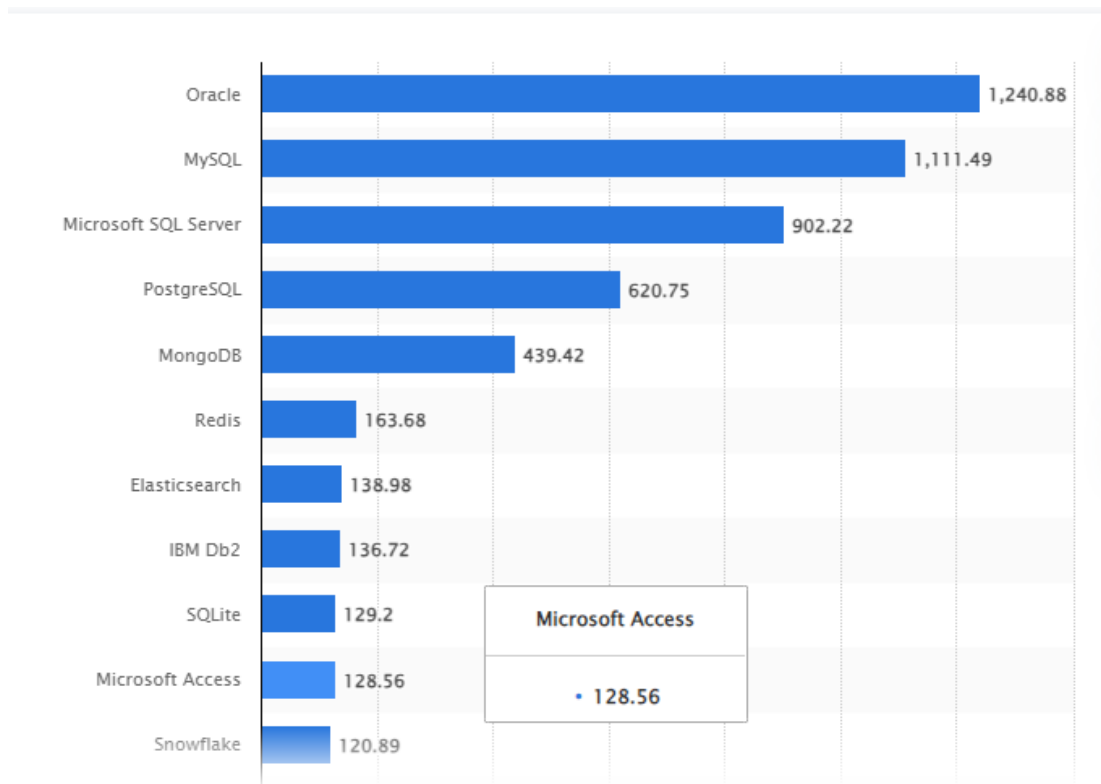


Рисунок 2.3 - Лідери рейтинг СУБД за версією компанії Statista на початок 2024 р.

Джерело: [12]

Для багатьох розробників вибір СУБД, на підставі наведених рейтингів, є очевидним – MySQL, яка на сьогоднішній день фактично стала промисловим стандартом.

Основні переваги СУБД MySQL:

- дві моделі ліцензування - безкоштовний сервер спільноти та корпоративний сервер;
- відкритий вихідний код;
- висока продуктивність;
- простота встановлення і адміністрування БД;
- гнучкі засоби використання і адміністрування БД;
- налаштування вихідного коду під потреби користувача;
- підтримка операцій як зі структурованими даними (SQL), так і напівструктурованими (JSON);

- забезпечення високого рівня безпеки [13].

Окрім того, СУБД MySQL дозволяє здійснювати всі операції з базами даних через веб-застосунок phpMyAdmin. Оскільки веб-застосунок доступний під ліцензією GNU General Public License, то він інтегрується розробниками веб-серверів у свої продукти [14].

Інтерфейс веб-застосунку phpMyAdmin наведений на рисунку 2.4.

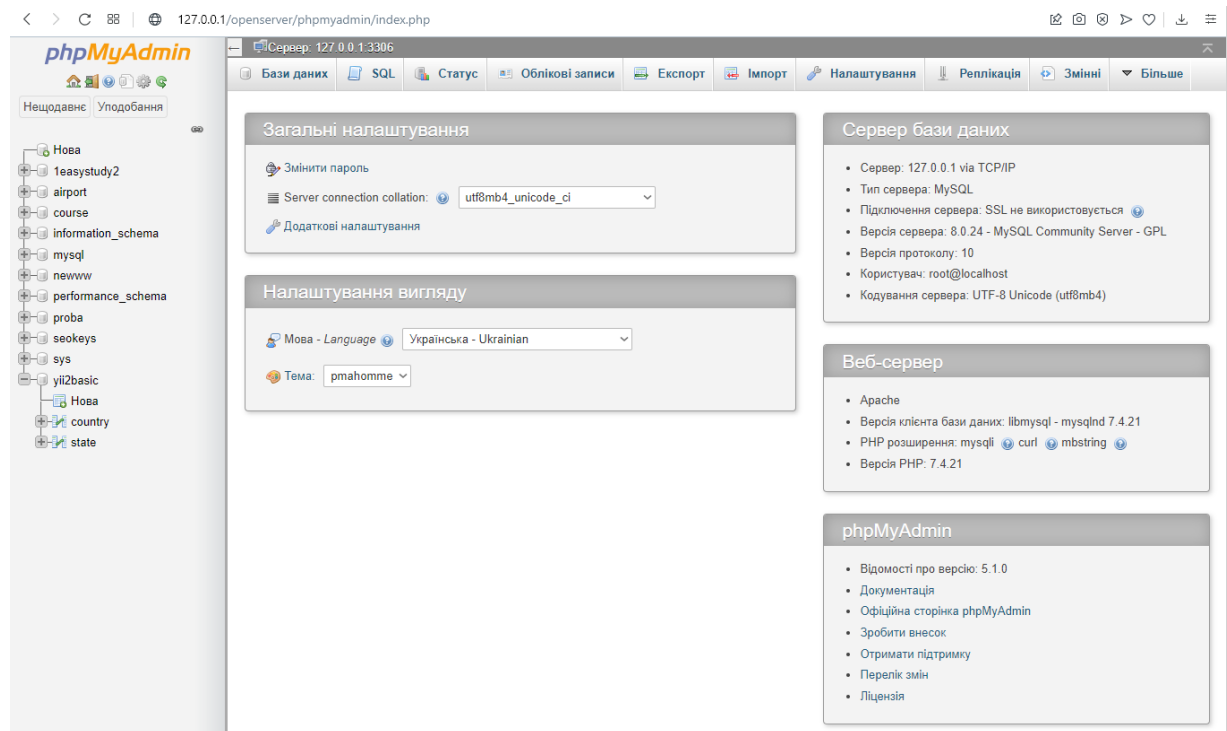


Рисунок 2.4 - Інтерфейс веб-додатку phpMyAdmin

Джерело: Запропоновано автором

Таким чином, для розробки проєкту була обрана СУБД MySQL.

2.3.2 Вибір інструментальних засобів розробки

Вибір мови програмування та відповідного інструментарію програмної реалізації застосунків, не дивлячись на умовну «спеціалізацію» мов, здійснюється з урахуванням особистих уподобань, наявних ресурсів, знань та навичок команди розробників, оскільки будь-яка задача, за деяким виключенням, може бути вирішена з використанням будь-якої сучасної мови програмування. Розробнику пропонується широкий спектр мов

програмування, кожна з яких має, як вже зазначалося, спеціалізацію, не позбавлена недоліків та наявних переваг.

При виборі мови програмування, поміж іншим, враховується багато факторів. Перш за все, це спеціалізація мови – її спрямування, і, як наслідок, швидкість реалізації функціоналу та інтерфейсу проєкту завдяки наявним в її складі засобам і інструментам, в тому числі і у якості додаткових бібліотек, фреймворків, пакетів тощо. Їх встановлення значно розширює базові можливості мови програмування за рахунок використання вже готового коду для реалізації функціоналу майбутнього застосунку. Іншим важливим фактором виступає наявність спільноти розробників і, головне, її активність, завдяки чому можна отримати вже готові рішення та відповіді на проблемні питання [15, 16].

Також, на вибір інструментарію розробки впливає популярність і рейтинг мов програмування, так як це опосередковано свідчить про їх наявні переваги і недоліки. Популярність мови програмування визначається індексом TIOBE Programming Community, що наведений на рисунку 2.5.

Рейтинг TIOBE Programming Community формується відповідно до кількості кваліфікованих інженерів у всьому світі, курсів і сторонніх постачальників. Для підрахунку рейтингів використовуються такі пошукові системи, як Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube і Baidu.

Dec 2023	Dec 2022	Change	Programming Language	Ratings	Change
1	1		 Python	13.86%	-2.80%
2	2		 C	11.44%	-5.12%
3	3		 C++	10.01%	-1.92%
4	4		 Java	7.99%	-3.83%
5	5		 C#	7.30%	+2.38%
6	7	▲	 JavaScript	2.90%	-0.30%
7	10	▲	 PHP	2.01%	+0.39%
8	6	▼	 Visual Basic	1.82%	-2.12%
9	8	▼	 SQL	1.61%	-0.61%
10	9	▼	 Assembly language	1.11%	-0.76%

Рисунок 2.5 - Індекс TIOBE Programming Community у грудні 2023 р.

Джерело: [17]

В таблиці 2.1 наведена порівняльна характеристика деяких мов програмування з першої десятки рейтингу TIOBE Programming Community.

Таблиця 2.1 - Порівняльна характеристика мов програмування

Ознака	JavaScript	PHP	C#	Java	Python
Спрямування розробки	інтерфейс веб-додатків	веб-додатки	веб-додатки, настільні системи з графічним інтерфейсом, корпоративні рішення, вбудовані системи	веб-додатки, настільні системи з графічним інтерфейсом, корпоративні рішення, вбудовані системи	наукові обчислення, машинне навчання, ігри, розвиток мови
Розповсюдження	відкритий код	відкритий код	ліцензування	ліцензування	відкритий код
Синтаксис	простий	простий	складний	складний	простий
Транслятор	інтерпретатор	інтерпретатор	компілятор	компілятор	інтерпретатор
Типізація змінних	динамічна	динамічна	статична	статична	динамічна

Джерело: Запропоновано автором

Не дивлячись на те, що мова PHP перебуває лише на 7-й позиції рейтингу, вона впевнено продовжує утримувати пальму першості у сфері розробки веб-застосунків. Багато популярних систем управління контентом (CMS), наприклад, WordPress, близько третини всіх сайтів в інтернеті та близько 60% сайтів на CMS написано з використанням PHP. PHP активно використовується і для розробки великих проєктів, зокрема, Facebook [17].

До того ж, PHP не вимагає ліцензування і розповсюджується з відкритим кодом. Він значно простіший, у порівнянні з C# та Java, в синтаксичному плані, а можливість динамічної типізації змінних робить процеси розробки та відладки програм значно швидшими.

Окрім того, прискорити процес розробки додатків можна, використовуючи CMS і фреймворки. Серед найбільш популярних PHP-фреймворків – Symfony, Yii2, Laravel.

Symfony - низькорівневий PHP-фреймворк, що дозволяє налаштовувати під будь-які потреби будь-яку систему. Натомість, така гнучкість у налаштуваннях мало що може дозволити використовувати в якості готових рішень, особливо для типових і дрібних завдань, тому він найменше підходить для швидкої розробки застосунків.

Yii2 та Laravel у своїх реалізаціях використовують компоненти Symfony, тому вони менш гнучкі у налаштуваннях. Ці фреймворки здобули свою популярність завдяки своїй зручності для розробників. Laravel має простий і зрозумілий синтаксис в порівнянні з іншими фреймворками. Натомість Yii2 відзначається своїми віджетами та наявним генератором коду [18, 19]. Обидва фреймворки використовують патерн MVC (Модель-Представлення (Вид)-Контролер).

Не дивлячись на те, що спочатку Laravel створювався як система розробки корпоративного програмного забезпечення, він швидко завоював популярність, оскільки має потужний та широкий набір інструментів і можливостей, які можна опанувати за короткий час. До основних переваг Laravel можна віднести:

- збирач скриптів та css;
- шаблонізатор Blade, що дозволяє використовувати простий PHP-код у представленнях;
- гнучке і зручне формування роутів;
- гнучке написання REST API;
- наявність великої кількості бібліотек Laravel.

До основних недоліків Laravel можна віднести:

- відсутність офіційної документації українською мовою;
- відсутність вбудованих генераторів коду [20, 21].

PHP-фреймворк Yii з'явився на ринку у 2006 році і лише у 2014 році з'явився Yii 2.0. Серед переваг Yii 2.0 можна виділити:

- легкий і простий у вивченні;
- має велику кількість вбудованих рішень, в т.ч. і для інтерфейсів;
- має потужний генератор моделей, контролерів та CRUD;
- наявна офіційна документація українською мовою;
- доступний у basic і advanced версіях.

До основних недоліків Yii 2.0 можна віднести:

- відсутність гнучкого формування роутів;
- відсутні постійні оновлення і вихід нових версій [22].

Після інсталяції Yii2 користувач окрім фреймворку отримує і шаблон проекту, в якому вже наявний деякий функціонал - заготовки кількох сторінок, в т.ч., авторизація користувачів, форма зворотнього зв'язку тощо.

Шаблон Basic фреймворку Yii 2.0 спрямований на розробку невеликих проєктів. Після його встановлення у браузері відобразиться початкова сторінка, наведена на рисунку 2.6.

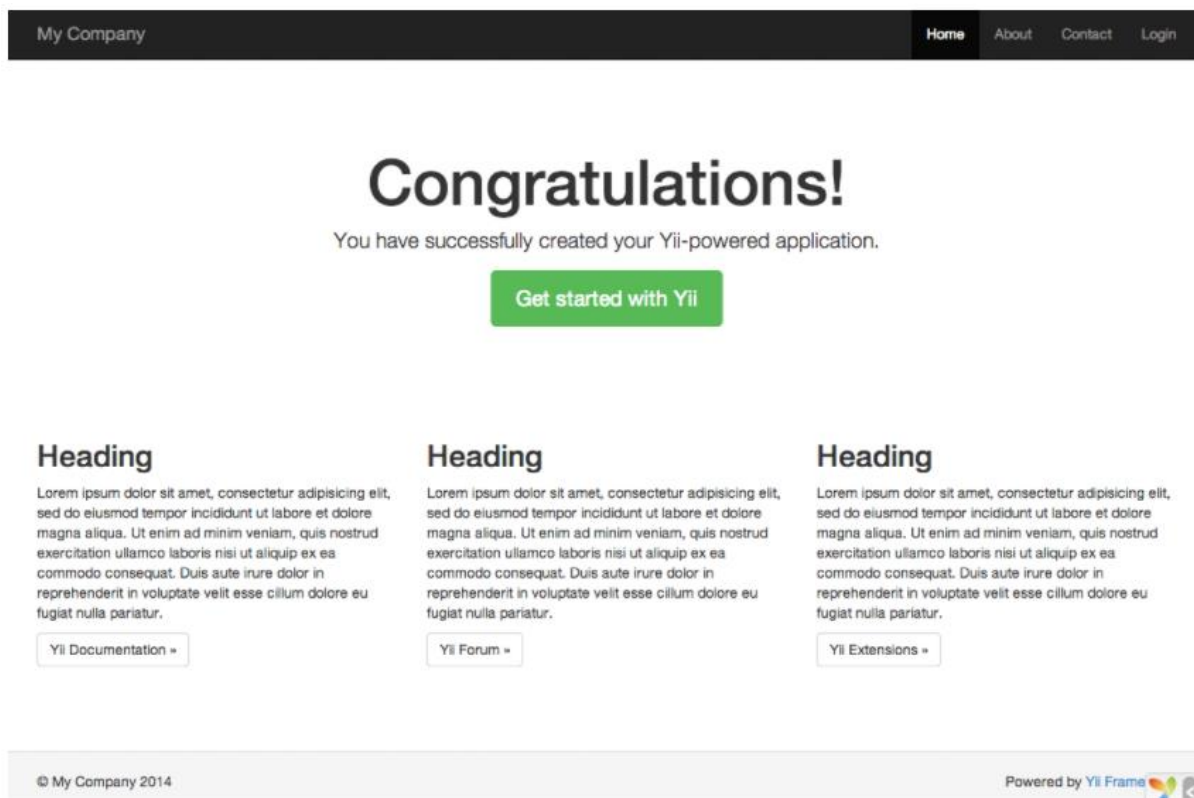


Рисунок 2.6 - Стартова сторінка шаблону Basic фреймворку Yii2

Джерело: [22].

В шаблоні Basic фреймворку Yii2 реалізований інтуїтивно-зрозумілий і доволі простий інтерфейс, що надає суттєві переваги у разі потреби швидкої розробки веб-застосунків, які, здебільшого не мають потреби у високоякісному інтерфейсі, на відміну від сайтів.

Водночас і структура папок та файлів веб-застосунку теж інтуїтивно-зрозуміла і логічна, що суттєво полегшує роботу розробників.

В таблиці 2.2 наведено основні папки і файли шаблону Basic [22, 24].

Таблиця 2.2 - Основні папки і файли шаблону Basic

Файл/папка	Вміст
composer.json	опис застосунку, використовується <i>Composer'ом</i>
config/	файли конфігурації
console.php	конфігурація консольного додатку
web.php	конфігурація веб-застосунку
commands/	файли класів консольних команд
controllers/	файли класів контролерів
models/	файли класів моделей
runtime/	файли, згенеровані <i>Yii</i> під час виконання додатку (журнали, кеш і т. п.)
vendor/	папки і файли фреймворку <i>Yii</i>
views/	файли представлень
web/	файли, доступні користувачу через <i>Web</i>
assets/	файли ресурсів (<i>javascript</i> та <i>css</i>)
index.php	вхідний скрипт застосунку
yii	скрипт виконання консольних команд <i>Yii</i>

Джерело: [22]

Таким чином, для розробки веб-застосунку проєктованої інформаційної системи доцільно обрати шаблон Basic фреймворку *Yii2*.

Іншим питанням, що потребує вирішення – є розробка застосунку збору даних про комп'ютеризовані робочі місця. Не дивлячись на те, що PHP зорієнтований на веб-розробку, він також може використовуватися для написання скриптів CLI (Command Line Interface)-скриптів, які можна запускати на сервері для виконання різних завдань, в т.ч. і збору інформації про систему. Це можливо завдяки розширенню COM, що забезпечує доступ до реєстру Windows, але доступне тільки на Windows.

Відповідно, через специфіку мови PHP, перш за все, через виконання її скриптів виключно веб-сервером, використовувати її для створення застосунку, що збиратиме дані про комп'ютеризоване робоче місце, недоцільно, оскільки, на комп'ютері робочого місця необхідно буде встановити додатково локальний веб-сервер, що збільшить навантаження на всю систему загалом. Для вирішення цієї проблеми можна створити застосунок, що збиратиме інформацію про комп'ютер робочого місця, використавши іншу мову програмування, наприклад, Python.

Python – відносно молода мова програмування. Вона була розроблена в 1990 році співробітником голландського інституту CWI Гвідо ван Росумом і названа на честь англійського колективу коміків "Monty Python's Flying Circus", поціновувачем яких був Гвідо. Швидкий розвиток мова одержала після того, як проектом зацікавилась і почала його просувати на ринку компанія BeOpen, а пізніше – DigitalCreations – автори Zope [23, с. 37].

Python - це інтерпретована, інтерактивна, об'єктно-орієнтована мова програмування, що підтримує кілька парадигм програмування - об'єктно-орієнтоване, процедурне та функціональне програмування. Python поєднує в собі надзвичайну потужність із дуже чітким синтаксисом. Його також можна використовувати як мову розширення для програм, яким потрібен програмований інтерфейс. Нарешті, Python портативний: він працює на багатьох варіантах Unix, включаючи Linux і macOS, а також на Windows [24].

На сьогоднішній день всі сучасні мови програмування мають репозиторії пакетів (модулів), що являють собою набори програм для розв'язку певних задач. Пакети розширюють базові можливості мови програмування, дозволяють уникати повторного написання коду, структурувати код у логічні модулі, що спрощує його читання та підтримку тощо. Python є однією з мов програмування з найбільшою кількістю доступних пакетів через його широке застосування та активність спільноти, що є однією з головних причин його популярності [23].

Серед пакетів, що можна використати в розробці застосунку, можна виділити:

- Platform використовується для отримання інформації про платформу, на якій виконується програма, - інформацію про пристрій, його операційну систему, її реліз та версію тощо. Окрім того, він дозволяє перевірити, чи відповідають характеристики апаратного забезпечення вимогам програми.
- psutil - бібліотека Python, яка використовується для доступу до деталей системи та утиліт обробки. Вона використовується для відстеження використання ресурсів системи та дозволяє контролювати використання таких ресурсів, як процесор, пам'ять, диски, мережа, датчики. Таким чином, ця бібліотека використовується для моніторингу системи, профілювання, обмеження ресурсів процесу та керування запущеними процесами.
- json. Python надає вбудовану бібліотеку json для роботи з об'єктами JSON.
- datetime призначений для роботи з датами та часом. Він надає різноманітні класи для представлення та обробки дат і часу, а також для форматування та аналізу дат і часу в різних форматах.
- mysql.connector - драйвер MySQL, написаний на Python, який не залежить від клієнтських бібліотек MySQL.
- winreg містить засоби, що надають Python API реєстру Windows.
- time – модуль, що забезпечує функції для роботи з часом. [24]

Таким чином, для розробки програмного забезпечення інформаційної підсистеми оцінювання працездатності комп'ютеризованих робочих місць обрано СУБД MySQL, PHP-фреймворк Yii2, мову програмування Python.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ

3.1 Розробка бази даних

Як зазначалося вище, розробка програмної реалізації інформаційної системи має починатись з розробки структури бази даних і, лише тільки після цього, приступають до програмної реалізації додатку, оскільки зміни структур бази даних після початку розробки програмного коду додатку вимагатимуть внесення суттєвих змін в його код.

Основою для проектування бази даних є аналіз предметної області - концептуальна модель, що представляє собою неформальний опис інформаційної структури. На підставі цієї моделі формується логічна модель, що являє собою формалізований опис об'єктів інформаційної системи.

На підставі аналізу досліджуваної предметної області, проведеному в першому розділі роботи, виділені наступні об'єкти:

- wps – комп'ютеризовані робочі місця;
- soft – дозволене програмне забезпечення;
- softtype – тип програмного забезпечення;
- employees – працівники;
- check_results – результати перевірки комп'ютеризованих робочих місць на відповідність попереднім зібраним даним;
- incidents – інциденти;
- system_info – результати перевірки комп'ютеризованих робочих місць.

Завершальним етапом проектування бази даних є створення моделі даних на підставі логічної моделі. У моделі даних представлені всі об'єкти з їх атрибутами та відносини і обмеження у системі. Модель даних подається, як правило, діаграмою відносин сутностей, також відомої як ER-діаграма.

Виходячи з викладеного вище, була розроблена структура бази даних, яка представлена ER-діаграмою на рисунку 3.1.

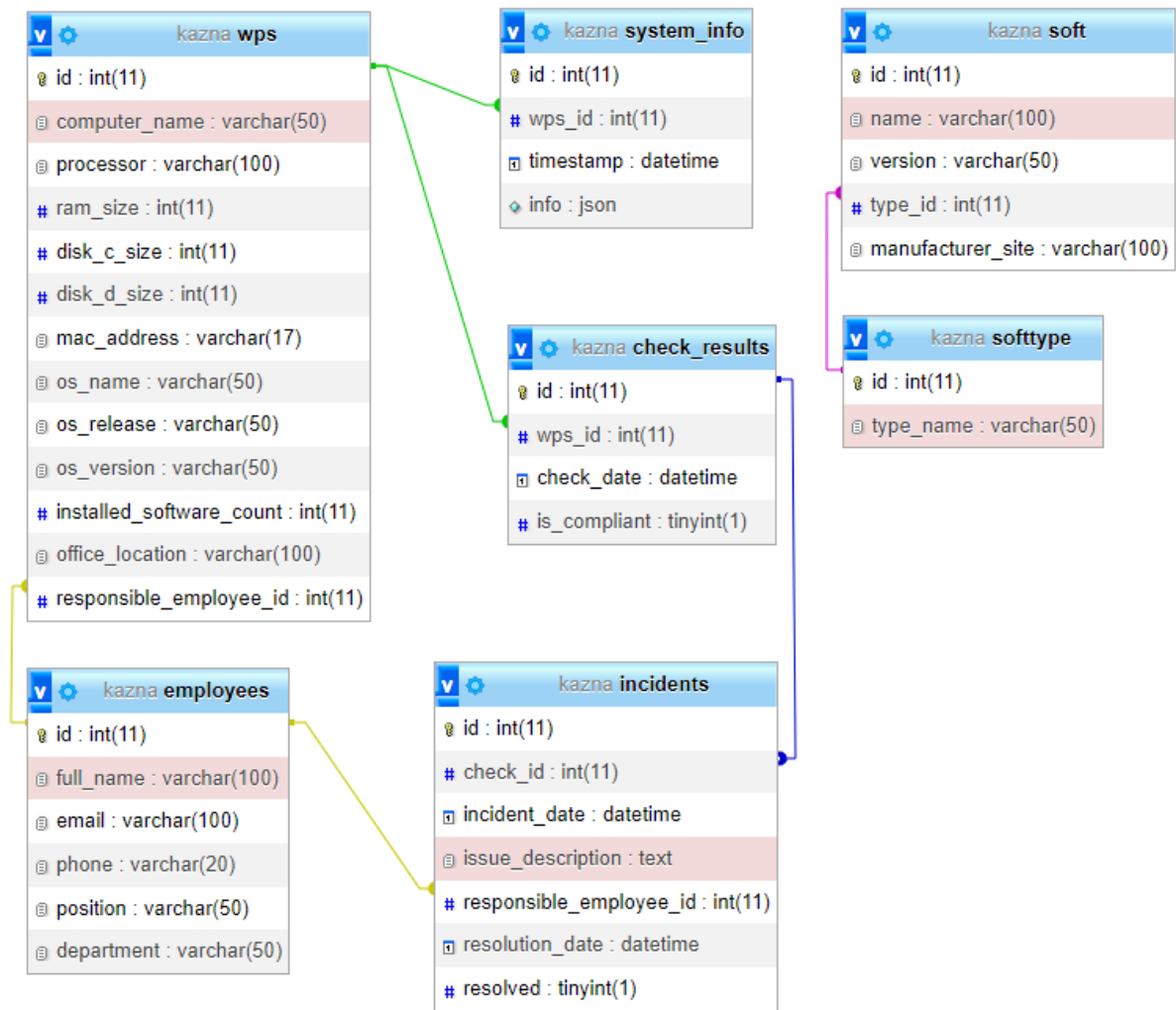


Рисунок 3.1 - ER-діаграма бази даних
Джерело: Запропоновано автором

Структура таблиць бази даних наведена в таблицях 3.2 – 3.9.

Таблиця 3.2 - Структура таблиці check_results

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
wps_id	fk	int(11)	id робочого місця	так
check_date		datetime	Дата збору даних	так
is_compliant		tinyint(1)	Відповідає попереднім даним	так

Джерело: Запропоновано автором

Таблиця 3.3 - Структура таблиці check_results

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
wps_id	fk	int(11)	id робочого місця	так
check_date		datetime	Дата збору даних	так
is_compliant		tinyint(1)	Відповідає попереднім даним	так

Джерело: Запропоновано автором

Таблиця 3.4 - Структура таблиці employees

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
full_name		varchar(100)	ПІБ працівника	ні
email		varchar(100)	Email працівника	так
phone		varchar(20)	Телефон працівника	так
position		varchar(50)	Посада	так
department		varchar(50)	Структурний підрозділ	так

Джерело: Запропоновано автором

Таблиця 3.5 - Структура таблиці incidents

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
check_id	fk	int(11)		так
incident_date		datetime	Час фіксації інциденту	так
issue_description		text	Опис інциденту	так
responsible_employee_id	fk	int(11)		так
resolution_date		datetime	Дата реагування	так
resolved		tinyint(1)	Питання вирішене	так

Джерело: Запропоновано автором

Таблиця 3.6 - Структура таблиці soft

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
name		varchar(100)	Назва ПЗ	ні
version		varchar(50)	Остання версія	так
type_id	fk	int(11)	Тип ПЗ	так
manufacturer_site		varchar(100)	Сайт виробника ПЗ	так

Джерело: Запропоновано автором

Таблиця 3.7 - Структура таблиці softtype

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
type_name		varchar(50)	Тип ПЗ	ні

Джерело: Запропоновано автором

Таблиця 3.8 - Структура таблиці system_info

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
wps_id	fk	int(11)	id робочого місця	так
timestamp		datetime	Час збору даних	так
info		json	Результати збору даних	так

Джерело: Запропоновано автором

Таблиця 3.9 - Структура таблиці wps

Поле	Ключ	Тип	Зміст	Null
id	pk	int(11)		ні
computer_name		varchar(50)	Назва комп'ютера	ні
processor		varchar(100)	Процесор	так
ram_size		int(11)	Обсяг RAM	так
disk_c_size		int(11)	Обсяг диску C	так
disk_d_size		int(11)	Обсяг диску D	так
mac_address		varchar(17)	Мас-адреса	так
os_name		varchar(50)	Операційна система	так
os_release		varchar(50)	Реліз	так
os_version		varchar(50)	Версія	так
installed_software_count		int(11)	Кількість встановлених програм	так
office_location		varchar(100)	Кабінет	так
responsible_employee_id	fk	int(11)	id відповідального працівника	так

Джерело: Запропоновано автором

Дамп структури бази даних наведено у Додатку А.

3.2 Створення програмного коду

Проектована інформаційна система, як зазначалося, складається з двох застосунків – застосунок для збору даних і веб-застосунок.

Застосунок збору даних про комп'ютер, встановлений на робочому місці, має здійснювати крім збору даних, запис на зовнішній носій і в базу даних. Також має здійснювати аналіз наступних даних:

- відповідність обладнання та параметрів системи зазначеним в базі даних для робочого місця;
- відповідність встановлених програм переліку дозволеного програмного забезпечення;
- відповідність версії кожної встановленої програми останній версії цієї програми, що зберігається в базі даних;
- достатній обсяг вільного простору накопичувачів (>10% від загального обсягу).

У разі виявлення невідповідностей при аналізі даних, здійснюється відповідний запис в таблицю incidents бази даних.

Програма реалізована з використанням процедурного програмування, тобто, весь функціонал реалізується окремими процедурами:

1. Функція для збору інформації про систему:

```
def collect_system_info():
    info = {
        "computer_name": os.getenv('COMPUTERNAME', 'unknown'),
        "processor": os.getenv('PROCESSOR_IDENTIFIER', 'unknown'),
        "ram_size": psutil.virtual_memory().total // (1024 * 1024), # В Мбайтах
        "disk_c_size": psutil.disk_usage('C:\\').total // (1024 * 1024), # В Мбайтах
        "disk_c_free": psutil.disk_usage('C:\\').free // (1024 * 1024), # В Мбайтах
        "os_name": os.name,
        "os_release": sys.getwindowsversion().major,
        "os_version": sys.getwindowsversion().build,
        "installed_software": []
    }

    # Отримання MAC-адреси
    for interface, addrs in psutil.net_if_addrs().items():
        for addr in addrs:
            if addr.family == psutil.AF_LINK:
                info["mac_address"] = addr.address

    try:
        disk_d = psutil.disk_usage('D:\\')
        info["disk_d_size"] = disk_d.total // (1024 * 1024) # В Мбайтах
```

```

    info["disk_d_free"] = disk_d.free // (1024 * 1024) # В Мбайтах
except FileNotFoundError:
    info["disk_d_size"] = 0
    info["disk_d_free"] = 0

# Отримання списку встановленого програмного забезпечення
info["installed_software"] = get_installed_software()

return info

```

2. Функція для отримання списку встановленого програмного забезпечення:

```

def get_installed_software():
    software_list = []
    uninstall_key = r"SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall"
    uninstall_key_64 = r"SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall"

    for root_key in [winreg.HKEY_LOCAL_MACHINE, winreg.HKEY_CURRENT_USER]:
        for sub_key in [uninstall_key, uninstall_key_64]:
            try:
                with winreg.OpenKey(root_key, sub_key) as key:
                    for i in range(0, winreg.QueryInfoKey(key)[0]):
                        try:
                            subkey_name = winreg.EnumKey(key, i)
                            with winreg.OpenKey(key, subkey_name) as subkey:
                                software = {}
                                software['name'] = winreg.QueryValueEx(subkey, "DisplayName")[0]
                                try:
                                    software['version'] = winreg.QueryValueEx(subkey, "DisplayVersion")[0]
                                except EnvironmentError:
                                    software['version'] = 'Unknown'
                                software_list.append(software)
                        except EnvironmentError:
                            continue
            except FileNotFoundError:
                continue

    return software_list

```

3. Функція для запису інформації у JSON-файл:

```

def write_to_json(data, filename="system_info.json"):
    try:
        with open(filename, 'r+') as file:
            file_data = json.load(file)
            file_data.append(data)

```

```

        file.seek(0)
        json.dump(file_data, file, indent=4)
    except FileNotFoundError:
        with open(filename, 'w') as file:
            json.dump([data], file, indent=4)

```

4. Функція для перевірки чи є комп'ютер у базі даних:

```

def check_computer_in_db(cursor, computer_name, mac_address):
    query = "SELECT id FROM wps WHERE computer_name = %s OR mac_address = %s"
    cursor.execute(query, (computer_name, mac_address))
    result = cursor.fetchone()
    return result[0] if result else None

```

5. Функція для додавання інциденту:

```

def add_incident(cursor, issue_description):
    query = "INSERT INTO incidents (incident_date, issue_description) VALUES (%s, %s)"
    cursor.execute(query, (datetime.now(), issue_description))

```

Лістинг коду програми наведений в Додатку Б.

Для створення веб-застосунку необхідно попередньо встановити Yii2. Завдяки вбудованому генератору коду, можна швидко розробити моделі – класи сутностей, а також контролери і представлення, що реалізують CRUD-операції з базою даних. Модель наслідує вбудований клас Active Record, який забезпечує об'єктно-орієнтований інтерфейс доступу до даних, що зберігаються у базах даних, та маніпулювання ними. Клас відповідає таблиці бази даних, об'єкт класу - рядку цієї таблиці, а атрибут об'єкта є значенням окремого стовпця рядка. Клас Active Record дозволяє отримувати доступ до атрибутів Active Record та викликати методи Active Record для доступу та маніпулювання даними замість написання SQL-виразів [22]. На рисунку 3.2 наведено вікно генератора коду під час генерації моделі (класу) Wps, що відповідає таблиці wps бази даних, і на рисунку 3.3 наведений генератор коду контролера і представлень, що реалізують CRUD-операції з даними таблиці wps. Після генерації файлів, розробнику залишається тільки вносити зміни в наявні файли задля реалізації функціоналу застосунку.

Model Generator

This generator generates an ActiveRecord class for the specified database table.

Database Connection ID
db

☐ Use Table Prefix
☒ Use Schema Name

Table Name
wps

☐ Standardize Capitals
☐ Singularize

Model Class Name
Wps

Namespace
app\models

Base Class
yii\db\ActiveRecord

Generate Relations
All relations

Generate Junction Relations As
Via Table

☒ Generate Relations from Current Schema
☒ Use '::class'
☒ Generate Labels from DB Comments
☐ Generate ActiveRecord
☐ Enable I18N

Code Template
default (C:\OSPanel\domains\kazna\vendor\yiisoft\yii2-gii\src\generators\model\def...

Preview Generate

Рисунок 3.2 - Створення моделі (класу) Wps в генераторі коду Yii2

Джерело: Запропоновано автором

Моделі доцільно в такий спосіб сформувати для всіх таблиць бази даних. Для генерації коду моделі розробнику достатньо у вікні генератора тільки обрати назву таблиці бази даних, за необхідності встановити додаткові пропонувані опції та натиснути кнопку “Generate”.

Рисунок 3.3 - Генерація коду контролера і представлень, що реалізують CRUD-операції з даними таблиці wps

Джерело: Запропоновано автором

Контролери з представленнями необхідно згенерувати лише для тих таблиць, маніпуляції з даними яких необхідно виконувати користувачу – incidents, softtype, soft.

В таблиці 3.10 наведені назви згенерованих файлів представлень з інформацією про їх призначення.

Таблиця 3.10 - Згенеровані файли представлень

Файл	Призначення
index.php	виведення всіх даних таблиці
create.php	створення нового запису в таблиці
view.php	виведення одного запису таблиці
update.php	внесення змін в запис таблиці
_form.php	форма для введення даних (викликається з create.php та update.php)
_search.php	форма для пошуку даних в таблиці (викликається з index.php)

Джерело: Запропоновано автором

Лістинг коду класу Wps, контролера WpsController та файлів представлень index.php, view.php, create.php, _form.php зі змінами, внесеними при розробці веб-застосунку, наведені в додатках В, Г, Д, Є, Ж, З відповідно.

Після генерації коду моделі, контролера і представлень, можна переглянути вміст таблиці. При цьому, завдяки використанню віджетів, не виникає труднощів з виведенням даних та оформленням відповідної сторінки. На рисунку 3.4 наведено сторінку з фрагментом даних, що містяться в таблиці wps бази даних після «відпрацювання» екшена index контролера WpsController та рендерингу файлу index.php.

Лістинг коду екшена index контролера WpsController:

```
public function actionIndex()
{
    $searchModel = new WpsSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}
```

kazna/web/index.php?r=wps%2Findex

Казна 1.0 Головна Програмне забезпечення Робочі місця Інциденти Повідомлення Вихід (admin)

[Home](#) / Wps

Wps

Create Wps

Showing 1-20 of 20 items.

#	ID	Ім'я комп'ютера	Процесор	RAM	Диск C	
1	1	PC-01	Intel Core i7-8700	16384	256000	View Edit Delete
2	2	PC-02	Intel Core i5-8500	8192	128000	View Edit Delete
3	3	PC-03	Intel Core i7-9700	16384	512000	View Edit Delete
4	4	PC-04	AMD Ryzen 5 3600	16384	256000	View Edit Delete
5	5	PC-05	AMD Ryzen 7 3700X	32768	512000	View Edit Delete

Рисунок 3.4 - Фрагмент даних таблиці wps

Джерело: Запропоновано автором

Останній стовпчик таблиці, наведеної на рисунку 3.4, містить посилання для переходу на сторінку перегляду запису з id=1 (див. рис. 3.5), переходу на сторінку його редагування (див. рис. 3.6) та для його видалення.

Казна 1.0 Головна Програмне забезпечення Робочі місця Інциденти Повідомлення Вихід (admin)	
Home / Wps / 1	
1	
Update Delete	
ID	1
Ім'я комп'ютера	PC-01
Процесор	Intel Core i7-8700
RAM	16384
Диск C	256000
Диск D	512000
Мас-адреса	00:1A:2B:3C:4D:5E
Операційна система	Windows 10
Реліз	10.0.19042
Версія	2009
Installed Software Count	10
Кабінет	Кабінет 101
Відповідальний працівник	1

Рисунок 3.5 - Сторінка перегляду запису з id=1 таблиці wps

Джерело: Запропоновано автором

[Home](#) / [Wps](#) / [1](#) / Update

Update Wps: 1

Ім'я комп'ютера

PC-01

Процесор

Intel Core i7-8700

RAM

16384

Диск C

256000

Диск D

512000

Mac-адреса

00:1A:2B:3C:4D:5E

Операційна система

Windows 10

Рисунок 3.6 - Сторінка редагування запису з id=1 таблиці wps

Джерело: Запропоновано автором

Сторінка створення нового робочого місця наведено на рисунку 3.7.

[Home](#) / [Wps](#) / Create Wps

Create Wps

Ім'я комп'ютера

Процесор

RAM

Диск C

Диск D

Мас-адреса

Операційна система

Реліз

Версія

Кабінет

Відповідальний працівник

Рисунок 3.7 - Вікно створення нового робочого місця

Джерело: Запропоновано автором

При створенні нового і редагуванні існуючого робочого місця, згенерована форма пропонує користувачам вводити дані тільки в текстові поля. Враховуючи те, що поле «Відповідальний працівник» є зовнішнім

ключем поля `id` таблиці `employees`, то вводити потрібно `id` працівника, що незручно користувачу, так як потрібно цей `id` знати. Щоб уникнути цього, необхідно використовувати спадний список з прізвищами працівників, але при цьому в базу даних має записатись `id` працівника. Для цього необхідно у файлі `_form.php` змінити рядок:

```
<?= $form->field($model, 'responsible_employee_id')->textInput() ?>
```

на такий:

```
<?= $form->field($model, 'responsible_employee_id')->dropdownList(
    Employees::find()->select(['full_name', 'id'])->indexBy('id')->column(),
    ['prompt'=>'Оберіть відповідального працівника']); ?>
```

Після заміни, як у вікні редагування, так і у вікні створення нового робочого місця, з'явиться спадний список з прізвищами працівників (див. рис. 3.8), так як вони використовують одну форму.

The screenshot shows a web application interface for creating a new work position. The browser address bar shows 'kazna/web/index.php?r=wps%2Fcreate'. The navigation bar includes 'Казна 1.0' and several menu items: 'Головна', 'Програмне забезпечення', 'Робочі місця', 'Інциденти', 'Повідомлення', and 'Вихід (admin)'. The breadcrumb trail is 'Home / Wps / Create Wps'. The main heading is 'Create Wps'. The form contains the following fields:

- Im'ya kom'pyutera (Computer name)
- Процесор (Processor)
- RAM
- Диск C (Disk C)
- Диск D (Disk D)

Below the 'Диск D' field is a dropdown menu with the prompt 'Оберіть відповідального працівника' (Select responsible employee). The dropdown list contains the following names:

- Іванов Іван Іванович
- Петров Петро Петрович
- Сидоров Сидор Сидорович
- Коваль Олександр Олександрович
- Бондар Олексій Олексійович
- Кравченко Сергій Сергійович
- Литвиненко Андрій Андрійович
- Шеєченко Михайло Михайлович
- Горбачов Віктор Вікторович
- Захаренко Ольга Олександрівна
- Василенко Василь Васильович
- Грищенко Григорій Григорович
- Іванова Олена Олексіївна
- Петренко Петро Петрович
- Семенов Семен Семенович
- Кузнецов Микола Миколайович
- Попов Павло Павлович
- Федоренко Федір Федорович
- Тимошенко Тимофій Тимофійович

At the bottom of the form is a green 'Save' button.

Рисунок 3.8 - Вікно створення нового робочого місця зі спадним списком для вибору відповідального працівника

Джерело: Запропоновано автором

В аналогічний спосіб можна зробити спадний список і для вибору операційної системи. Для формування списку слід відібрати всі записи таблиці `soft` зі значенням у полі `type_id=1`, що відповідає типу програмного забезпечення «Операційна система». При цьому слід врахувати, що в поле має записатися назва операційної системи. Відповідний рядок прийме вигляд:

```
<?= $form->field($model, 'os_name')->dropdownList(Soft::find()->select(['name', 'name'])->where(['type_id'=>1])->indexBy('name')->column(), ['prompt'=>'Оберіть операційну систему']); ?>
```

Однієї з вимог при розробці веб-застосунку є необхідність забезпечення авторизованого доступу до функціоналу веб-застосунку, відповідно, і до даних в базі даних. В Yii2 немає необхідності створювати окремий код для сторінки авторизації при вході в систему, достатньо, використовуючи вбудований клас `AccessControl`, встановити правило для перелічених в масиві сторінок (перерахувати екшени контролера), що вони доступні виключно для авторизованих користувачів. При запиті неавторизованого користувача до будь-якого екшена контролера, буде виводитись сторінка авторизації, наведена на рисунку 3.9.

Лістинг фрагменту коду функції `behaviors()`, що забезпечує авторизований доступ до визначених сторінок:

```
public function behaviors()
{
    return [
        'access' => [
            'class' => AccessControl::class,
            'only' => ['index', 'logout'],
            'rules' => [
                [
                    'actions' => ['index', 'logout'],
                    'allow' => true,
                    'roles' => ['@'],
                ],
            ],
        ],
    ],

    // інший код
}
```

Рисунок 3.9 - Сторінка авторизації

Джерело: Запропоновано автором

Для створення меню в складі Yii2 є віджет Nav, у якому пункти меню представляються у вигляді масиву, що містить назви пунктів, посилання для відкриття відповідних сторінок та додаткові опції, наприклад, відображення пункту в меню за деяких умов. Для налаштування меню необхідно внести зміни в файл `views\layouts\main.php`.

Лістинг коду файлу наведений в додатку I.

Лістинг коду віджета Nav з масивом пунктів меню:

```
echo Nav::widget([
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => [
        ['label' => 'Головна', 'url' => ['/site/index'], 'visible'=>$visible],
        ['label' => 'Програмне забезпечення', 'url' => ['/soft/index'], 'visible'=>$visible],
        ['label' => 'Робочі місця', 'url' => ['/wps/index'], 'visible'=>$visible],
        ['label' => 'Інциденти', 'url' => ['/incidents/index'], 'visible'=>$visible],
        ['label' => 'Повідомлення', 'url' => ['/site/contact'], 'visible'=>$visible ],
        Yii::$app->user->isGuest
        ? ['label' => 'Вхід', 'url' => ['/site/login']]
        : '<li class="nav-item">'
            . Html::beginForm(['/site/logout'])
            . Html::submitButton(
```

```

        'Вихід (' . Yii::$app->user->identity->username . ')',
        ['class' => 'nav-link btn btn-link logout']
    )
    . Html::endForm()
    . '</li>'
]
));
NavBar::end();

```

3.3 Впровадження інформаційної підсистеми в роботу

Розроблена інформаційна система складається з бази даних і двох застосунків – застосунок для збору даних і веб-застосунок. База даних і скрипти веб-застосунку розміщуються на локальному веб-сервері казначейства. За необхідності, вони можуть бути розміщені на веб-хостингу в мережі інтернет, що забезпечить доступ до бази даних з будь-якого комп'ютера мережі незалежно від часу і місця перебування фахівця.

Застосунок для збору даних про комп'ютери розміщується на кожному комп'ютеризованому робочому місці та має запускатися при кожному завантаженні операційної системи. Застосунок працює в фоновому режимі.

Перед використанням веб-застосунку необхідно попередньо ввести наступні дані:

- перелік дозволеного для використання в організації програмного забезпечення з зазначенням його назви, типу, останньої версії та сайту розробника;
- перелік робочих місць організації з зазначенням імені комп'ютера, встановленого процесора, обсягу оперативної пам'яті, обсягу диску C, обсягу диску D, мак-адреси, встановленої операційної системи, її релізу і версії, кабінету розташування робочого місця та відповідального працівника.

Окрім того, для внесення даних, зазначених вище, потрібно також попередньо ввести типи програмного забезпечення і список співробітників.

Для забезпечення авторизованого доступу до функціоналу веб-застосунку, при вході в систему з'являється вікно авторизації, наведене на рисунку 3.9.

Після успішної авторизації, з'являється головна сторінка веб-застосунку, наведена на рисунку 3.10. Для прикладу наведено скріншот сторінки за 25.05.2024 року. Сторінка містить меню, назву веб-застосунку і перелік нових інцидентів, зафіксованих 25.05.2024 р. та попередніх інцидентів, питання з якими ще не вирішене, тобто, в системі відсутня відмітка про виконання завдання.

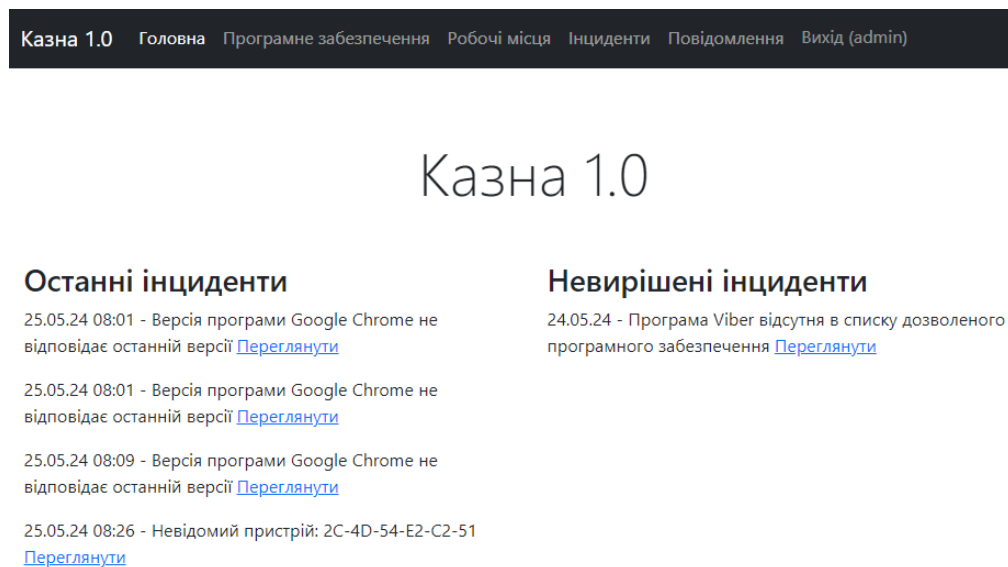


Рисунок 3.10 - Головна сторінка веб-застосунку

Джерело: Запропоновано автором

Аналізуючи виведені дані, можна зробити висновок, що вийшло нове оновлення Google Chrome і оновлення відсутнє на 3-х робочих місцях. Також було встановлено новий пристрій, який ще не додано до бази даних. Окрім того, напередодні на робочому місці було встановлено месенджер Viber, що відсутній в переліку дозволеного програмного забезпечення.

Керівник ІТ-відділу має призначити фахівця відділу відповідальним за розгляд та/або вирішення питань інциденту. Після розгляду інциденту відповідальний фахівець робить відмітку в системі (див. рис. 3.11).

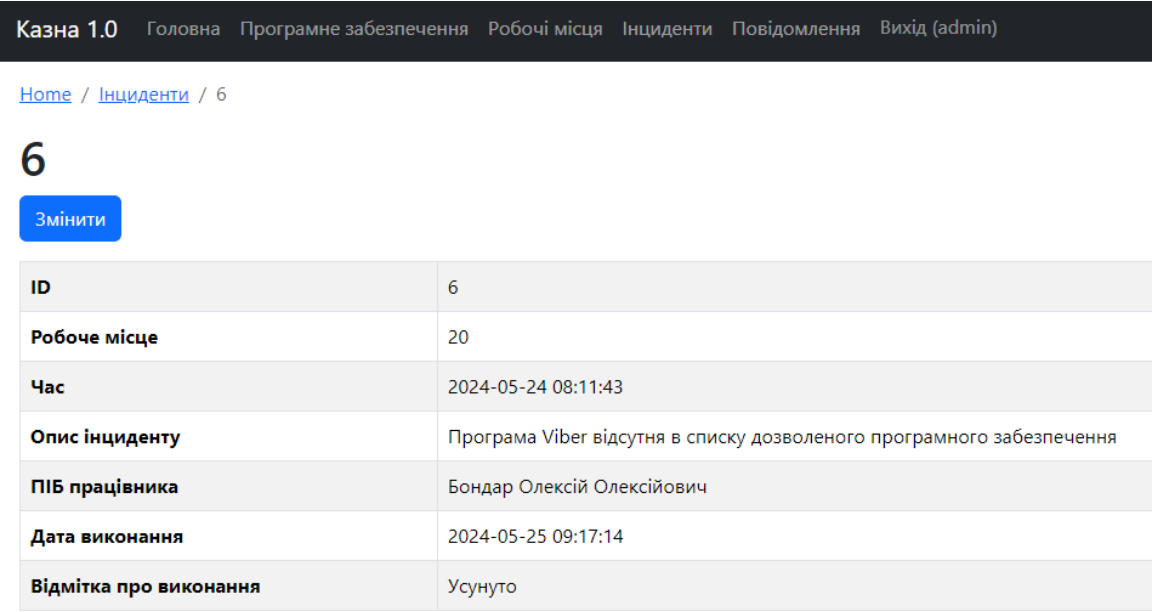


Рисунок 3.11 - Сторінка редагування інциденту

Джерело: Запропоновано автором

При створенні нового робочого місця необхідно за допомогою меню «Робочі місця» відкрити список наявних робочих місць і натиснути кнопку «Нове робоче місце» (див. рис. 3.5), після чого відкриється сторінка, наведена на рисунку 3.7.

Інший функціонал веб-застосунку реалізується аналогічно через меню. Відповідні сторінки мають вигляд аналогічний наведеним вище сторінкам.

ВИСНОВКИ

В кваліфікаційній роботі виконано дослідження предметної області і виявлено фактори для оцінювання працездатності комп'ютеризованих робочих місць; проаналізовано існуючі рішення;

Спроектowana інформаційна система оцінювання працездатності комп'ютеризованих робочих місць, визначено функціональні вимоги до системи та її архітектура; обрано інструментарій програмної реалізації інформаційної системи.

Реалізовано програмне забезпечення інформаційної системи, що складається з застосунку збору даних про комп'ютеризоване робоче місце і веб-застосунку, що забезпечує доступ до бази даних та обробку даних з неї. База даних реалізована в СУБД MySQL, застосунок збору даних – мовою Python, веб-застосунок — мовою PHP з використанням фреймворку Yii2.

Впровадження і використання розробленого програмного забезпечення в роботу Управління Державної казначейської служби України у Середино-Будському районі Сумської області дозволить:

- своєчасно реагувати і попереджувати виникнення проблем з апаратним і програмним забезпеченням комп'ютеризованих робочих місць організації;
- підвищити надійність і стабільність роботи парку комп'ютерної техніки;
- поліпшити безпеку даних;
- зменшити кількість простоїв;
- покращити комфорт працівників і підвищити продуктивність їх роботи;
- підвищити ефективність роботи організації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Микитюк І. ТЕОРЕТИЧНІ ЗАСАДИ ЦИФРОВІЗАЦІЇ КАЗНАЧЕЙСТВА УКРАЇНИ //Економічний вісник університету. – 2022. – №. 54. – С. 19-23.
2. Химич І. Г. Індустрія 5.0: вдосконалення тенденцій індустрії 4.0 //Цифрова економіка і сталий розвиток: новітні тенденції у фінансах, обліку, менеджменті та соціально-поведінкових науках. – ЗУІ ім. Ф. Ракоці II, 2024, 2024.
3. Грицишин В. В. Формування системи управління мотивацією персоналу в умовах інноваційного розвитку підприємства : дис. – Тернопіль, ЗУНУ, 2023.
4. Царик І. М., Черниченко О. І., Черниш Н. М. Критерії і напрямки аналізу ефективності організації робочого місця //Вісник Чернігівського державного технологічного університету. Серія: Економічні науки. – 2013. – №. 1. – С. 38-43.
5. Сапіташ В. С., Іванчук Я. В. Алгоритм надання рекомендацій для підбору комплектуючих апаратного забезпечення комп'ютерних систем : дис. – ВНТУ, 2023.
6. Северина С. В. Інформаційна безпека та методи захисту інформації //Вісник Запорізького національного університету. Економічні науки. – 2016. – №. 1. – С. 155-161.
7. Завгородня, А. С. Технологія моніторингу інцидентів, пов'язаних з встановленням програмного забезпечення, шляхом використання програмного агента : дипломний проект бакалавра : 25 Кібербезпека / Завгородня Анастасія Сергіївна. – Київ, 2022. – 60 с.
8. Волочій Б. Ю. и др. Визначення впливу оновлення програмного забезпечення на показники надійності відмовостійкої багатопроцесорної системи //Східно-Європейський журнал передових технологій. – 2013. – Т. 3. – №. 9 (63). – С. 55-59.

9. Говор Д. С. Проблематика оновлення техніки та програмного забезпечення в Національній поліції України. – Інформаційно-аналітичне забезпечення діяльності органів сектору безпеки і оборони України: матеріали наук.-практ. конф.(м. Львів, 22 груд. 2023 р.).–Львів: ДУВС, 2024.–С. 39-40, 2023.

10. Кудін А. П., Бабич В. О., Бабич О. С. СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ //Sciences of Europe. – 2021. – №. 76-1. – С. 64-68.

11. DB-Engines Ranking [Електронний ресурс] // Consulting company "solid IT" – Режим доступу до ресурсу: <https://db-engines.com/en/ranking>.

12. Ranking of the most popular database management systems worldwide, as of January 2022 [Електронний ресурс] // Statista – Режим доступу до ресурсу: <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>.

13. Md Kamaruzzaman. Top 10 Databases to Use in 2021 [Електронний ресурс] / Md Kamaruzzaman // Towards Data Science. – 2021. – Режим доступу до ресурсу: <https://towardsdatascience.com/top-10-databases-to-use-in-2021-d7e6a85402ba>.

14. Andrew Comeau. MySQL Explained: Your Step-by-Step Guide. - 2017. – 211с.

15. Contieri M. Clean Code Cookbook: Recipes to Improve the Design and Quality of your Code - O'Reilly Media, 2023. - 428 p.

16. Cosker, Glynn (2023), "What Is Information Technology? A Beginner's Guide to the World of IT", Technology Blog, Rasmussen University.

17. Топ 7 проєктів, що написані на PHP. [Електронний ресурс] - URL: <https://levelup.ua/ru/top-7-proektov-kotorye-napisany-na-php>.

18. 17 переваг фреймворку Laravel [Електронний ресурс] - URL: <https://icstudio.online/post/17-perevag-frejmworku-laravel>.

19. The Definitive Guide to Yii 2.0 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>.

20 The PHP Framework for Web Artisans [Електронний ресурс] // Taylor Otwell – Режим доступу до ресурсу: <https://laravel.com/>.

21. What's New in Laravel 10? Build a Secure & Scalable App With Laravel [Електронний ресурс] - URL: <https://webmobtech.com/blog/whats-new-in-laravel-10/>

22. The Definitive Guide to Yii 2.0 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>.

23. Яковенко А. В. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко. – Київ: КПІ ім. Ігоря Сікорського, 2018. – 195 с.

24. General Python FAQ [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/faq/general.html>

ДОДАТКИ

Додаток А

Дамп структури бази даних

```

CREATE DATABASE kazna;
USE kazna;

-- Таблиця для типів програмного забезпечення
CREATE TABLE softtype (
  id INT AUTO_INCREMENT PRIMARY KEY,
  type_name VARCHAR(50) NOT NULL
);

-- Таблиця для дозволеного програмного забезпечення
CREATE TABLE soft (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  version VARCHAR(50),
  type_id INT,
  manufacturer_site VARCHAR(100),
  FOREIGN KEY (type_id) REFERENCES softtype(id)
);

-- Таблиця для працівників
CREATE TABLE employees (
  id INT AUTO_INCREMENT PRIMARY KEY,
  full_name VARCHAR(100) NOT NULL,
  email VARCHAR(100),
  phone VARCHAR(20),
  position VARCHAR(50),
  department VARCHAR(50)
);

-- Таблиця для комп'ютеризованих робочих місць
CREATE TABLE wps (
  id INT AUTO_INCREMENT PRIMARY KEY,
  computer_name VARCHAR(50) NOT NULL,
  processor VARCHAR(100),
  ram_size INT, -- В мегабайтах
  disk_c_size INT, -- В мегабайтах
  disk_d_size INT, -- В мегабайтах
  mac_address VARCHAR(17),
  os_name VARCHAR(50),
  os_release VARCHAR(50),
  os_version VARCHAR(50),
  installed_software_count INT,
  office_location VARCHAR(100),
  responsible_employee_id INT,
  FOREIGN KEY (responsible_employee_id) REFERENCES employees(id)
);

-- Таблиця для результатів перевірки системної інформації
CREATE TABLE check_results (
  id INT AUTO_INCREMENT PRIMARY KEY,
  wps_id INT,
  check_date DATETIME,
  is_compliant BOOLEAN,

```

```
FOREIGN KEY (wps_id) REFERENCES wps(id)
);

-- Таблиця для інцидентів
CREATE TABLE incidents (
  id INT AUTO_INCREMENT PRIMARY KEY,
  check_id INT,
  issue_description TEXT,
  responsible_employee_id INT,
  resolution_date DATETIME,
  resolved BOOLEAN,
  FOREIGN KEY (check_id) REFERENCES check_results(id),
  FOREIGN KEY (responsible_employee_id) REFERENCES employees(id)
);
```

Додаток Б

Лістинг програми збору даних про комп'ютеризоване робоче місце

```

import mysql.connector
import json
import psutil
import platform
from datetime import datetime
import time

# Налаштування підключення до бази даних
db_config = {
    'user': 'root',
    'password': '',
    'host': 'localhost',
    'database': 'kazna'
}

# Функція для збору інформації про систему
def collect_system_info():
    info = {
        "computer_name": platform.node(),
        "processor": platform.processor(),
        "ram_size": psutil.virtual_memory().total,
        "disk_c_size": psutil.disk_usage('/').total,
        "disk_c_free": psutil.disk_usage('/').free,
        "os_name": platform.system(),
        "os_release": platform.release(),
        "os_version": platform.version(),
        "installed_software": []
    }

    # Отримання MAC-адреси
    for interface, addrs in psutil.net_if_addrs().items():
        for addr in addrs:
            if addr.family == psutil.AF_LINK:
                info["mac_address"] = addr.address

    try:
        disk_d = psutil.disk_usage('D:\\')
        info["disk_d_size"] = disk_d.total
        info["disk_d_free"] = disk_d.free
    except FileNotFoundError:
        info["disk_d_size"] = 0
        info["disk_d_free"] = 0

    return info

# Функція для запису інформації у JSON-файл
def write_to_json(data, filename="system_info.json"):
    try:
        with open(filename, 'r+') as file:
            file_data = json.load(file)
            file_data.append(data)
            file.seek(0)
            json.dump(file_data, file, indent=4)
    except FileNotFoundError:
        with open(filename, 'w') as file:

```

```

    json.dump([data], file, indent=4)

# Функція для перевірки чи є комп'ютер у базі даних
def check_computer_in_db(cursor, computer_name, mac_address):
    query = "SELECT id FROM wps WHERE computer_name = %s OR mac_address = %s"
    cursor.execute(query, (computer_name, mac_address))
    result = cursor.fetchone()
    return result[0] if result else None

# Функція для додавання інциденту
def add_incident(cursor, issue_description):
    query = "INSERT INTO incidents (incident_date, issue_description) VALUES (%s, %s)"
    cursor.execute(query, (datetime.now(), issue_description))

# Основна функція програми
def main():
    while True:
        system_info = collect_system_info()
        timestamp = datetime.now()
        system_info["timestamp"] = str(timestamp)

        # Запис інформації у JSON-файл
        write_to_json(system_info)

        try:
            connection = mysql.connector.connect(**db_config)
            cursor = connection.cursor()

            # Перевірка чи є комп'ютер у таблиці wps
            computer_id = check_computer_in_db(cursor, system_info["computer_name"],
            system_info["mac_address"])

            if not computer_id:
                add_incident(cursor, f"Невідомий пристрій: {system_info['mac_address']}")

            # Збереження результатів перевірки у таблицю check_results
            check_query = "INSERT INTO check_results (wps_id, check_date, is_compliant) VALUES (%s, %s, %s)"
            cursor.execute(check_query, (computer_id, timestamp, True))

            # Перевірка відповідності програмного забезпечення
            cursor.execute("SELECT name, version FROM soft")
            allowed_software = cursor.fetchall()

            for software in system_info["installed_software"]:
                if software["name"] not in [item[0] for item in allowed_software]:
                    add_incident(cursor, f"Програма {software['name']} відсутня в списку дозволеного програмного
                    забезпечення")
                elif software["version"] not in [item[1] for item in allowed_software if item[0] == software["name"]]:
                    add_incident(cursor, f"Версія програми {software['name']} не відповідає останній версії")

            # Фіксування всіх змін у базі даних
            connection.commit()
        except mysql.connector.Error as err:
            print(f"Error: {err}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()

```

```
# Чекати 1 годину до наступного збору інформації  
time.sleep(3600)  
  
if __name__ == "__main__":  
    main()
```

Додаток В

Лістинг коду класу Wps

```
<?php

namespace app\models;

use Yii;

/**
 * This is the model class for table "wps".
 *
 * @property int $id
 * @property string $computer_name Ім'я комп'ютера
 * @property string|null $processor Процесор
 * @property int|null $ram_size RAM
 * @property int|null $disk_c_size Диск C
 * @property int|null $disk_d_size Диск D
 * @property string|null $mac_address Mac-адреса
 * @property string|null $os_name Операційна система
 * @property string|null $os_release Реліз
 * @property string|null $os_version Версія
 * @property int|null $installed_software_count
 * @property string|null $office_location Кабінет
 * @property int|null $responsible_employee_id Відповідальний працівник
 *
 * @property CheckResults[] $checkResults
 * @property Employees $responsibleEmployee
 * @property SystemInfo[] $systemInfos
 */
class Wps extends \yii\db\ActiveRecord
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'wps';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['computer_name'], 'required'],
            [['ram_size', 'disk_c_size', 'disk_d_size', 'installed_software_count', 'responsible_employee_id'], 'integer'],
            [['computer_name', 'os_name', 'os_release', 'os_version'], 'string', 'max' => 50],
            [['processor', 'office_location'], 'string', 'max' => 100],
            [['mac_address'], 'string', 'max' => 17],
            [['responsible_employee_id'], 'exist', 'skipOnError' => true, 'targetClass' => Employees::class,
'targetAttribute' => ['responsible_employee_id' => 'id']],
        ];
    }

    /**
     * {@inheritdoc}
     */
}
```

```

*/
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'computer_name' => 'Ім'я комп'ютера',
        'processor' => 'Процесор',
        'ram_size' => 'RAM',
        'disk_c_size' => 'Диск C',
        'disk_d_size' => 'Диск D',
        'mac_address' => 'Mac-адреса',
        'os_name' => 'Операційна система',
        'os_release' => 'Реліз',
        'os_version' => 'Версія',
        'installed_software_count' => 'Installed Software Count',
        'office_location' => 'Кабінет',
        'responsible_employee_id' => 'Відповідальний працівник',
    ];
}

/**
 * Gets query for [[CheckResults]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getCheckResults()
{
    return $this->hasMany(CheckResults::class, ['wps_id' => 'id']);
}

/**
 * Gets query for [[ResponsibleEmployee]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getResponsibleEmployee()
{
    return $this->hasOne(Employees::class, ['id' => 'responsible_employee_id']);
}

/**
 * Gets query for [[SystemInfos]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getSystemInfos()
{
    return $this->hasMany(SystemInfo::class, ['wps_id' => 'id']);
}
}

```

Додаток Г

Лістинг коду контролера WpsController

```

<?php

namespace app\controllers;

use app\models\Wps;
use app\models\WpsSearch;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

class WpsController extends Controller
{
    /**
     * @inheritdoc
     */
    public function behaviors()
    {
        return array_merge(
            parent::behaviors(),
            [
                'verbs' => [
                    'class' => VerbFilter::className(),
                    'actions' => [
                        'delete' => ['POST'],
                    ],
                ],
            ]
        );
    }

    /**
     * Lists all Wps models.
     *
     * @return string
     */
    public function actionIndex()
    {
        $searchModel = new WpsSearch();
        $dataProvider = $searchModel->search($this->request->queryParams);

        return $this->render('index', [
            'searchModel' => $searchModel,
            'dataProvider' => $dataProvider,
        ]);
    }

    /**
     * Displays a single Wps model.
     * @param int $id ID

```

```

* @return string
* @throws NotFoundException if the model cannot be found
*/
public function actionView($id)
{
    return $this->render('view', [
        'model' => $this->findModel($id),
    ]);
}

/**
 * Creates a new Wps model.
 * If creation is successful, the browser will be redirected to the 'view' page.
 * @return string|\yii\web\Response
 */
public function actionCreate()
{
    $model = new Wps();

    if ($this->request->isPost) {
        if ($model->load($this->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }
    } else {
        $model->loadDefaultValues();
    }

    return $this->render('create', [
        'model' => $model,
    ]);
}

/**
 * Updates an existing Wps model.
 * If update is successful, the browser will be redirected to the 'view' page.
 * @param int $id ID
 * @return string|\yii\web\Response
 * @throws NotFoundException if the model cannot be found
 */
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($this->request->isPost && $model->load($this->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    }

    return $this->render('update', [
        'model' => $model,
    ]);
}

/**

```

```

* Deletes an existing Wps model.
* If deletion is successful, the browser will be redirected to the 'index' page.
* @param int $id ID
* @return \yii\web\Response
* @throws NotFoundHttpException if the model cannot be found
*/
public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}

/**
* Finds the Wps model based on its primary key value.
* If the model is not found, a 404 HTTP exception will be thrown.
* @param int $id ID
* @return Wps the loaded model
* @throws NotFoundHttpException if the model cannot be found
*/
protected function findModel($id)
{
    if (($model = Wps::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundHttpException('The requested page does not exist.');
```

Додаток Д

Лістинг коду файлу представлення views\wps\index.php

```
<?php
use app\models\Wps;
use yii\helpers\Html;
use yii\helpers\Url;
use yii\grid\ActionColumn;
use yii\grid\GridView;
$this->title = 'Робочі місця';
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="wps-index">
    <h1><?= Html::encode($this->title) ?></h1>
    <p>
        <?= Html::a('Нове робоче місце', ['create'], ['class' => 'btn btn-success']) ?>
    </p>
    <?php //echo $this->render('_search', ['model' => $searchModel]); ?>

    <?= GridView::widget([
        'dataProvider' => $dataProvider,
        'filterModel' => $searchModel,
        'columns' => [
            ['class' => 'yii\grid\SerialColumn'],
            'computer_name',
            'processor',
            'ram_size',
            'disk_c_size',
            'mac_address',
            'os_name',
            'os_release',
            'os_version',
            'office_location',
            [
                'attribute' => 'responsibleEmployee.full_name',
            ],
            [
                'class' => ActionColumn::className(),
                'urlCreator' => function ($action, Wps $model, $key, $index, $column) {
                    return Url::toRoute([$action, 'id' => $model->id]);
                }
            ],
        ],
    ]); ?>

</div>
```

Додаток Є

Лістинг коду файлу представлення views\wps\view.php

```
<?php

use yii\helpers\Html;
use yii\widgets\DetailView;

$this->title = $model->id;
$this->params['breadcrumbs'][] = ['label' => 'Робочі місця', 'url' => ['index']];
$this->params['breadcrumbs'][] = $this->title;
\yii\web\YiiAsset::register($this);
?>
<div class="wps-view">
    <h1><?= Html::encode($this->title) ?></h1>
    <p>
        <?= Html::a('Змінити', ['update', 'id' => $model->id], ['class' => 'btn btn-primary']) ?>
        <?= Html::a('Видалити', ['delete', 'id' => $model->id], [
            'class' => 'btn btn-danger',
            'data' => [
                'confirm' => 'Ви впевнені, що хочете видалити дані?',
                'method' => 'post',
            ],
        ]) ?>
    </p>

    <?= DetailView::widget([
        'model' => $model,
        'attributes' => [
            'id',
            'computer_name',
            'processor',
            'ram_size',
            'disk_c_size',
            'disk_d_size',
            'mac_address',
            'os_name',
            'os_release',
            'os_version',
            'office_location',
            //'responsible_employee_id',
            [
                'attribute' => 'responsibleEmployee.full_name',
                'label' => 'Відповідальний працівник',
            ],
        ],
    ]) ?>
</div>
```

Додаток Ж

Лістинг коду файлу представлення views\wps\create.php

```
<?php

use yii\helpers\Html;

$this->title = 'Нове робоче місце';
$this->params['breadcrumbs'][] = ['label' => 'Робочі місця', 'url' => ['index']];
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="wps-create">

    <h1><?= Html::encode($this->title) ?></h1>

    <?= $this->render('_form', ['model' => $model,]) ?>

</div>
```

Додаток 3

Лістинг коду файлу представлення views\wps_form.php

```
<?php

use yii\helpers\Html;
use yii\widgets\ActiveForm;
use app\models\Employees;
use app\models\Soft;
?>

<div class="wps-form">

    <?php $form = ActiveForm::begin(); ?>

    <?= $form->field($model, 'computer_name')->textInput(['maxlength' => true]) ?>

    <?= $form->field($model, 'processor')->textInput(['maxlength' => true]) ?>

    <?= $form->field($model, 'ram_size')->textInput() ?>

    <?= $form->field($model, 'disk_c_size')->textInput() ?>

    <?= $form->field($model, 'disk_d_size')->textInput() ?>

    <?= $form->field($model, 'mac_address')->textInput(['maxlength' => true]) ?>

    <?= $form->field($model, 'os_name')->dropdownList(Soft::find()->select(['name', 'name'])-
>where(['type_id'=>1])->indexBy('name')->column(), ['prompt'=>'Оберіть операційну систему']); ?>

    <?= $form->field($model, 'os_release')->textInput(['maxlength' => true]) ?>

    <?= $form->field($model, 'os_version')->textInput(['maxlength' => true]) ?>

    <?= $form->field($model, 'office_location')->textInput(['maxlength' => true]) ?>

    <?php //echo $form->field($model, 'responsible_employee_id')->textInput() ?>

    <?= $form->field($model, 'responsible_employee_id')->dropdownList(Employees::find()->select(['full_name',
'id'])->indexBy('id')->column(), ['prompt'=>'Оберіть відповідального працівника']); ?>

    <div class="form-group">
        <?= Html::submitButton('Зберегти', ['class' => 'btn btn-success']) ?>
    </div>

    <?php ActiveForm::end(); ?>

</div>
```

Додаток І

Лістинг коду файлу views\layouts\main.php

```

<?php
use app\assets\AppAsset;
use app\widgets\Alert;
use yii\bootstrap5\Breadcrumbs;
use yii\bootstrap5\Html;
use yii\bootstrap5\Nav;
use yii\bootstrap5\NavBar;

AppAsset::register($this);

$this->registerCsrfMetaTags();
$this->registerMetaTag(['charset' => Yii::$app->charset], 'charset');
$this->registerMetaTag(['name' => 'viewport', 'content' => 'width=device-width, initial-scale=1, shrink-to-fit=no']);
$this->registerMetaTag(['name' => 'description', 'content' => $this->params['meta_description'] ?? '']);
$this->registerMetaTag(['name' => 'keywords', 'content' => $this->params['meta_keywords'] ?? '']);
$this->registerLinkTag(['rel' => 'icon', 'type' => 'image/x-icon', 'href' => Yii::getAlias('@web/favicon.ico')]);

$visible = false;
if(!Yii::$app->user->isGuest)
{
    $visible = true;
}

?>
<?php $this->beginPage() ?>
<!DOCTYPE html>
<html lang="<?= Yii::$app->language ?>" class="h-100">
<head>
    <title><?= Html::encode($this->title) ?></title>
    <?php $this->head() ?>
</head>
<body class="d-flex flex-column h-100">
<?php $this->beginBody() ?>

<header id="header">
    <?php
    NavBar::begin([
        'brandLabel' => Yii::$app->name,
        'brandUrl' => Yii::$app->homeUrl,
        'options' => ['class' => 'navbar-expand-md navbar-dark bg-dark fixed-top']
    ]);
    echo Nav::widget([
        'options' => ['class' => 'navbar-nav navbar-right'],
        'items' => [
            ['label' => 'Головна', 'url' => ['/site/index'], 'visible'=>$visible],
            ['label' => 'Програмне забезпечення', 'url' => ['/soft/index'], 'visible'=>$visible],
            ['label' => 'Робочі місця', 'url' => ['/wps/index'], 'visible'=>$visible],
            ['label' => 'Інциденти', 'url' => ['/incidents/index'], 'visible'=>$visible],
            ['label' => 'Повідомлення', 'url' => ['/site/contact'], 'visible'=>$visible ],
            Yii::$app->user->isGuest
            ? ['label' => 'Вхід', 'url' => ['/site/login']]
            : '<li class="nav-item">'
                . Html::beginForm(['/site/logout'])
                . Html::submitButton(

```

```

        'Вихід (' . Yii::$app->user->identity->username . ')',
        ['class' => 'nav-link btn btn-link logout']
    )
    . Html::endForm()
    . '</li>'
]
]);
NavBar::end();
?>
</header>

<main id="main" class="flex-shrink-0" role="main">
    <div class="container">
        <?php if (!empty($this->params['breadcrumbs'])): ?>
            <?= Breadcrumbs::widget(['links' => $this->params['breadcrumbs']]) ?>
        <?php endif ?>
        <?= Alert::widget() ?>
        <?= $content ?>
    </div>
</main>

<footer id="footer" class="mt-auto py-3 bg-light">
    <div class="container">
        <div class="row text-muted">
            <div class="col-md-6 text-center text-md-start">&copy; УДКC <?= date('Y') ?></div>
            <div class="col-md-6 text-center text-md-end"><?php //echo Yii::powered() ?></div>
        </div>
    </div>
</footer>

<?php $this->endBody() ?>
</body>
</html>
<?php $this->endPage() ?>

```