

МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

РОЗРОБКА ДОДАТКУ ЕЛЕКТРОННОГО НАВЧАННЯ

Здобувача вищої освіти першого (бакалаврського) рівня,
4 курсу, групи КБ-19-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»


(підпис)

Павленка Михайла Дмитровича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Татушняк Тетяна Володимирівна, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

✓ Керівник освітньо-професійної
програми


(підпис)

Федотова-Півень Ірина Миколаївна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри

протокол від «26» травня 2023 № 13

Завідувач
кафедри


(підпис)

Філоненко Михайло Миколайович, к.фіз.-мат.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2023

Реєстрація № 26 06.06.2023
(дата)



(підпис відповідальної особи.)

Результат перевірки

до захисту
(до захисту, на доопрацювання)

29.05.23

(дата)



(підпис)

Результат захисту

4 (добре)
(оцінка)

15.06.2023

(дата)



(підпис)

Голова комісії:



(підпис)

В.М. Франчук

(ініціали, прізвище)

Члени комісії:



(підпис)

М.М. Філоненко

(ініціали, прізвище)



(підпис)

О.В. Редич

(ініціали, прізвище)



(підпис)

С.В. Котух

(ініціали, прізвище)

БЛАНК ОЦІНЮВАННЯ ДИПЛОМНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	24		15.06.2023 Франчук В.М. <i>В.М.</i> Філоненко М.М. <i>М.М.</i>
2. Дотримання вимог щодо оформлення	від 0 до 10	6		Редич О.В. <i>О.В.</i>
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	3		Котух Є.В. <i>Є.В.</i> Суховерхівка Л.П. <i>Л.П.</i>
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	37		

Оцінювання кваліфікаційної роботи науковим керівником - 38 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100- бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підписи)
75	C	4 (добре)	15.06.2023	Франчук В.М. <i>В.М.</i> Філоненко М.М. <i>М.М.</i> Редич О.В. <i>О.В.</i> Котух Є.В. <i>Є.В.</i> Суховерхівка Л.П. <i>Л.П.</i>

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

факультет фінансів та цифрових технологій
 кафедра комп'ютерних та інформаційних технологій і систем
 спеціальність 122 «Комп'ютерні науки»
 світня програма «Інформаційні управляючі системи і технології (за галузями)»

Затверджую

Завідувач кафедри

Філоненко М.М.

« 16 » 11 2022

**Завдання на випускню кваліфікаційну (дипломну) роботу
 здобувачу вищої освіти**

Павленку Михайлу Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної (дипломної) роботи

Розробка додатку електронного навчання

Затверджена наказом ректора від «15» листопада 2022 № 1205 (зі змінами
 наказ від «04» квітня 2023 № 361)

2. Строк здачі закінченої роботи «26» травня 2023

3. Цільова установка застосування набутих знань, умінь в розробці веб-
 додатку для електронного навчання

Мета роботи розробка та реалізація веб-додатку для електронного навчання
 на мові програмування C++.

Об'єкт дослідження: системи дистанційної освіти в Україні та технології
 розробки веб-додатків

Предмет дослідження процес розробки веб-додатку для електронного
 навчання з функціоналом календаря

База дослідження ТОВ «ЕПАМ ДІДЖИТАЛ»

4. Вихідні дані до роботи інформаційні дані розкладу занять університету

5. Перелік графічного матеріалу: створення моделі за допомогою Gii,
 структура Bootstrap, приклад відображення завдань на календарі
 у режимі "Відобразити місяць", Приклад відображення завдань на календарі
 у режимі «Відобразити тиждень».

6. Зміст випускної кваліфікаційної (дипломної) роботи

Вступ

Розділ 1. Аналіз предметної області

1.1 Основні відомості про веб-проекування

1.2. Інформаційне освітнє середовище. Функції системи дистанційного навчання в інформаційному освітньому середовищі

1.3. Порівняльна характеристика засобів розробки систем дистанційного навчання

Розділ 2. Технологія проєктування додатку

2.1. Технічне завдання на проєктування та розробку додатку

2.2. Інструменти розробки додатку

Розділ 3. Програмна реалізація додатку

3.1. Функціонал додатку електронного навчання «Органайзер»

3.2. Інструкція для користувача додатком «Органайзер»

Висновки

Список використаних джерел

Додаток А. Код програми

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	За фактом
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.11.2022	15.11.2022
2	I розділ (провести аналіз предметної області та аналіз обраних інструментів)	01.02.2023	01.02.2023
3	II розділ (методика реалізації дослідження)	03.04.2023	04.04.2023
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проєктні рішення)	10.05.2023	09.05.2023
5	Висновки	15.05.2022	15.05.2022
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	26.05.2023	26.05.2023
7	Отримання відгуку керівника	06.06.2023	06.06.2023
8	Отримання зовнішньої рецензії	08.06.2023	08.06.2023

8. Дата видачі завдання «16» листопада 2022

9. Керівник випускної кваліфікаційної (дипломної) роботи


(підпис)

Ратушняк Т.В.
(прізвище, ініціали)

10. Керівник освітньо-професійної програми


(підпис)

Федотова-Півень І.М.
(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти


(підпис)

Павленко М.Д.
(прізвище, ініціали)

АНОТАЦІЯ

Розробка додатку електронного навчання

Мета роботи полягає в розробці та реалізації веб-додатку для електронного навчання на мові програмування C++.

Завданнями роботи є: проведення аналізу предметної області, розробка архітектури та дизайну додатку, проєктування бази даних, реалізація додатку та тестування, розгортання додатку.

Предметом дослідження дослідження процес розробки веб-додатку для електронного навчання з функціоналом календаря.

Об'єктом дослідження є системи дистанційної освіти в Україні та технології розробки веб-додатків.

Базою дослідження виступає ТОВ «ЕПАМ ДІДЖИТАЛ».

За результатами дослідження сформульовані етапи створення та впровадження веб-додатка.

Одержані результати можуть бути використані учасниками навчального процесу як інструмент планування.

Кваліфікаційна робота містить 50 сторінок, 4 рисунки, список використаних джерел з 17 найменувань, 1 додаток.

Ключові слова: дистанційне навчання, веб-додаток, додаток для електронного навчання, середовище програмування Visual Studio, мова програмування C++, електронний календар.

Рік виконання кваліфікаційної роботи 2023

Рік захисту роботи 2023

ABSTRACT

Development of an e-learning application

The purpose of the work is to develop and implement a web application for e-learning in the C++ programming language.

The tasks of the work are: conducting an analysis of the subject area, developing the architecture and design of the application, designing the database, implementing the application and testing, deploying the application.

The subject of the study is the process of developing a web application for e-learning with calendar functionality.

The object of the study is distance education systems in Ukraine and web application development technologies.

EPAM DIGITAL LLC serves as the basis of the research.

Based on the results of the research, the stages of creating and implementing a web application are formulated.

The obtained results can be used by the participants of the educational process as a planning tool.

The qualification work contains 50 pages, 4 figures, a list of used sources from 17 titles, 1 appendix.

Keywords: distance learning, web application, e-learning application, Visual Studio programming environment, C++ programming language, electronic calendar.

The year of performance of qualification work is 2023

The year of job protection is 2023

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1. Основні відомості про веб-проєктування	11
1.2. Інформаційне освітнє середовище. Функції системи дистанційного навчання в інформаційному освітньому середовищі	12
1.3. Порівняльна характеристика засобів розробки систем дистанційного навчання	16
РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ДОДАТКУ	20
2.1. Технічне завдання на проєктування та розробку додатку	20
2.2. Інструменти розробки додатку	20
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ	33
3.1. Функціонал додатку електронного навчання «ОРГАНАЙЗЕР»	33
3.2. Інструкція для користувача додатком «ОРГАНАЙЗЕР»	37
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	41
ДОДАТОК А. Код програми	44

ВСТУП

У кваліфікаційній роботі розглядається створення програми на мові C++ в середовищі Visual Studio [1-2], яка буде призначена для студентів з метою полегшення навчання. Програма буде зосереджена на ефективному управлінні часом та нагадуванні про заплановані завдання.

Безліч програмних продуктів в епоху цифрових технологій розробляється для веб-ресурсів та мобільних телефонів. Більшість останніх передових моделей електронних пристроїв не поступаються за функціоналом та характеристиками громіздким мобільним телефонам та комп'ютерам.

Зараз у світі практично не залишилося людей, які б не користувалися комп'ютером чи ноутбуком. Причинами їх популярності можна назвати:

- універсальність: Комп'ютери та ноутбуки можуть виконувати різноманітні завдання;
- зручність: Ноутбуки є портативними й можуть бути використані в будь-якому зручному для користувача місці;
- з'єднання з Інтернетом: Комп'ютери та ноутбуки надають доступ до Інтернету, що відкриває безліч можливостей для комунікації, отримання інформації, онлайн-розваг та електронної торгівлі;
- робочі можливості: Використання комп'ютерів та ноутбуків є необхідним для багатьох професій;
- розваги: Комп'ютери та ноутбуки також широко використовуються для розваг. Вони дозволяють грати в комп'ютерні ігри, дивитися фільми, слухати музику, переглядати фотографії та виконувати багато інших дій, які забезпечують релаксацію та розвагу.

На сьогоднішній день основну класифікацію веб-браузерів можна розподілити на такі: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Opera

та інші. Веб-браузер Google Chrome є одним з найпоширеніших та надійних варіантів, він пропонує широкі можливості та простий інтерфейс, що робить його популярним серед користувачів.

Завдяки тому, що велика кількість студентів користуються комп'ютерами та телефонами з доступом до мережі інтернет, це викликає попит на веб-додатки з функціоналом для електронного навчання.

Мета роботи є розробка та реалізація веб-додатку для електронного навчання на мові програмування C++.

Об'єктом дослідження є системи дистанційної освіти в Україні та технології розробки веб-додатків.

Завданнями роботи є: проведення аналізу предметної області, розробка архітектури та дизайну додатку, проєктування бази даних, реалізація додатку та тестування, розгортання додатку.

За результатами дослідження сформульовані етапи створення та впровадження веб-додатка.

Одержані результати можуть бути використані учасниками навчального процесу як інструмент планування.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Основні відомості про веб-проєктування

Веб-додаток – це програмне забезпечення, яке розробляється для використання у веб-середовищі та доступу через веб-браузер. Він призначений для надання функціональності та послуг користувачам без необхідності встановлення додаткового програмного забезпечення на їх пристрої.

Процес створення веб-додатку включає наступні етапи:

- Розробка функціональності та інтерфейсу: Веб-розробники створюють код, що визначає поведінку та зовнішній вигляд додатка. Вони використовують різноманітні мови програмування, такі як HTML, CSS та JavaScript, для створення веб-сторінок та взаємодії з користувачем.
- Тестування та налагодження: Розробники перевіряють роботу додатка, виявляють та виправляють помилки, оптимізують його продуктивність та забезпечують його сумісність з різними веб-браузерами.
- Розгортання та хостинг: Після завершення розробки додаток розгортається на веб-сервері і стає доступним для користувачів через Інтернет.

При розробці веб-додатків необхідно враховувати деякі особливості: вони повинні працювати у різних веб-браузерах, таких як Google Chrome, Mozilla Firefox, Safari тощо, на різних операційних системах, таких як Windows, macOS або Linux. Також важливо забезпечити, щоб додаток був адаптивним та мав гарний зовнішній вигляд на різних пристроях з різними розмірами екранів. Веб-додатки можуть використовувати різні технології та бібліотеки, щоб забезпечити розширену функціональність, такі як робота з базами даних, взаємодія зі сторонніми сервісами або використання AJAX для асинхронного оновлення вмісту без перезавантаження сторінки.

Веб-додатки можуть бути доступні для користувачів через URL-адреси та не вимагати встановлення додаткового програмного забезпечення. Це робить їх

зручними для використання на різних пристроях, таких як персональні комп'ютери, ноутбуки, планшети, смарт-телевізори та інші, які мають доступ до Інтернету та веб-браузерів.

Сучасні організації мають різноманітні засоби для розповсюдження та організації інформації серед населення. Один з найбільш ефективних способів – веб-додаток, який дозволяє доступатися до інформації через веб-сайт. В порівнянні з іншими методами, такими як додатки для персональних комп'ютерів або мобільні додатки, веб-додаток має свої переваги. Використання веб-додатку дозволяє уникнути проблеми з обмеженим дисковим простором на електронних пристроях користувачів. Оскільки веб-додаток розміщений на сервері, користувачі можуть отримати доступ до інформації без необхідності завантажувати та встановлювати програму на свої пристрої. Це особливо зручно для користувачів, які хочуть швидко отримати доступ до інформації без додаткових зусиль.

Отже, використання веб-додатків є перевагою в порівнянні з іншими методами розповсюдження інформації, такими як додатки для персональних комп'ютерів або мобільні додатки. Веб-додатки забезпечують швидкий та зручний доступ до інформації без необхідності встановлення програми на пристрої користувача, що дозволяє економити час та сприяє зручності використання.

1.2. Інформаційне освітнє середовище. Функції системи дистанційного навчання в інформаційному освітньому середовищі

Розглянемо, як розуміють термін «дистанційну освіту» наші вчені. «Дистанційна освіта - комплекс масових освітніх послуг, що надаються спеціальним інформаційним середовищем за допомогою засобів передачі навчально-методичної інформації на великі відстані (телефон, радіо, інтернет, телебачення, супутниковий зв'язок та ін.)» [3-4]. Проте, насамперед важливо

визначитися, яких концептуальних педагогічних положеннях доцільно будувати сучасний курс дистанційного навчання. Коротко їх можна звести до таких:

1. У центрі процесу навчання знаходиться самостійна пізнавальна діяльність учня (навчання, а не викладання). Вчення, самостійне придбання та застосування знань стало потребою сучасної людини протягом усього її свідомого життя в умовах постіндустріального, інформатизованого суспільства.

2. Звідси, з одного боку, необхідна гнучкіша система освіти, що дозволяє здобувати знання там і тоді, де і коли це зручно навчається. А з іншого, важливо, щоб учень не тільки опанував певну суму знань, але, що представляється набагато важливіше, щоб він навчився самостійно набувати знання, працювати з інформацією, опанував способи пізнавальної діяльності, які він міг би застосовувати в подальшому за необхідності підвищувати кваліфікацію, міняти професійну орієнтацію тощо.

3. Організація самостійної (індивідуальної чи груповий) діяльності учнів у мережі передбачає не меншою мірою, ніж у очному навчанні, використання новітніх педагогічних технологій, стимулюючих розкриття внутрішніх резервів кожного учня і водночас сприяють формуванню соціальних якостей особистості (уміння працювати у колективі, виконуючи різні соціальні ролі, допомагаючи один одному у спільній діяльності, вирішуючи спільними зусиллями складні пізнавальні завдання). Насамперед йдеться про широке застосування методу проектів, навчання у співпраці, дослідницьких, проблемних методів.

4. Самостійне придбання знань має носити пасивний характер, навпаки, що навчається від початку має бути залучений в активну пізнавальну діяльність, не обмежується оволодінням знаннями, але неодмінно передбачає їх застосування вирішення різноманітних проблем навколишньої дійсності. У ході такого навчання учні (будь-якого віку та соціального статусу) повинні, перш за все, навчитися набувати та застосовувати знання, шукати та знаходити потрібні для

них засоби навчання та джерела інформації, вміти працювати з цією інформацією.

5. Система контролю за засвоєнням знань та способами пізнавальної діяльності, здатністю, вмінням застосовувати отримані знання у різних проблемних ситуаціях повинна мати систематичний характер, будуватися як на основі оперативного зворотного зв'язку (закладеного як у текст матеріалу, так і можливості оперативного звернення до викладача чи консультанта курсу) , і відстроченого контролю (наприклад, при тестуванні).

6. Дистанційне навчання, індивідуалізоване за своєю суттю, не повинно разом з тим виключати можливостей комунікації не тільки з викладачем, а й з іншими партнерами, співробітництва у процесі різного роду пізнавальної та творчої діяльності. Проблеми соціалізації дуже актуальні під час дистанційного навчання.

У системі дистанційного навчання використовуються як традиційні, так і специфічні методи, засоби та форми навчання, що базуються на комп'ютерних та телекомунікаційних технологіях. Найважливішою складовою освітнього процесу при ДО є цілеспрямована та контрольована інтенсивна самотійна робота учня.

Аналіз вітчизняної та зарубіжної теорії та практики [Tytenko] дозволив відзначити характерні особливості ДО. Серед них:

- Модульність. В основу програм ДО закладається модульний принцип, що дозволяє з набору незалежних дисциплін формувати навчальний план, що відповідає індивідуальним та груповим потребам учнів.

- Гнучкість. Учні займаються у зручний для себе час, у зручному місці та темпі.

- Дальнодія. Відстань від місця знаходження, що навчається до навчального закладу (за умови якісної роботи зв'язку) не впливає на якість навчального процесу.

- Паралельність. Навчання може проводитись «без відриву від

виробництва».

- Асинхронність. У процесі навчання навчальний та навчальний працюють за зручним для кожного розкладом.
- Охоплення, чи масовість. Кількість учнів у системі ДО перестав бути критичним параметром.
- Рентабельність. ДО економічно ефективніше традиційних форм здобуття освіти.
- Викладач. У системі ДН викладач виконує нові ролі та функції.
- Той, хто навчається. Вимоги до учня істотно відрізняються від традиційних.
- Нові інформаційні технології (НДТ). У системі ДО використовуються всі види інформаційних технологій, але переважно нові інформаційні технології, засобами яких є комп'ютери, комп'ютерні мережі, мультимедіа системи тощо.
- Соціальність. ДО певною мірою знімає соціальну напруженість, забезпечуючи рівну можливість здобуття освіти.
- Інтернаціональність. За допомогою ДО здійснюється експорт та імпорт освітніх послуг.

Перелічені особливості визначають переваги ДО над іншими формами здобуття освіти. Однак, пред'являючи специфічні вимоги до викладача та слухача, ДО не знижує, а підвищує їх трудовитрати.

Канонічна дидактична система, в якій протікає традиційний освітній процес, складається з семи елементів: мета, зміст, методи, засоби, форми навчання, які навчають та навчаються.

Дамо характеристику елементів педагогічної системи ДО, зазначаючи особливості.

Мета навчання складає систему знань, умінь та навичок, що формуються відповідно до державних освітніх стандартів. Ціль має ієрархічну структуру. Мета

навчальної дисципліни постає як один із елементів мети підготовки спеціаліста. Мета вивчення теми є елементом системи цілей навчальної дисципліни тощо. Ціль – це початок організації навчального процесу, і вона може трактуватися як засвоєння змісту на необхідному рівні. Таке загальнопедагогічне розуміння мети інваріантно до форми здобуття освіти і, отже, застосовується до ДО.

Ті, хто навчається при ДО, виявляються в абсолютно нових умовах не тільки тому, що можуть перебувати на великій відстані, бути зайнятими виробничими справами тощо, але й тому, що їм надано «свободу» у навчанні. Це і вільний графік, і гнучкий вибір дисциплін тощо. Однак "свободою" треба вміти користуватися. Високі вимоги пред'являються особистісним якостям слухачів ДО: цілеспрямованості, наполегливості, чесності та інших. Вони мають основами методики і техніки самостійної роботи.

Навчання в системі ДН вимагає певної готовності до навчання (початкового набору знань, умінь, навичок) та матеріально-технічного забезпечення робочого місця. Противники ДО часто вказують цей факт, як у можливість створення додаткового «майнового» нерівності у здобутті освіти. Уникнути цього допоможе використання кореспондентської моделі навчання, розробка комплектів навчальних матеріалів у розрахунку на контингент учнів, який не має доступу до засобів НІТ тощо. З іншого боку, можна бути впевненими, що прогрес мікроелектроніки дозволить вже в найближчому майбутньому вирішити проблему доступності засобів НІТ.

Використання технологій дистанційного навчання дозволяє:

1. Знизити витрати на проведення навчання (не потрібні витрати на оренду приміщень, поїздки до місця навчання, як учнів, так і викладачів тощо);
2. Проводити навчання великої кількості осіб;
3. Підвищити якість навчання за рахунок застосування сучасних засобів, об'ємних електронних бібліотек тощо;
4. Створити єдине освітнє середовище (особливо актуально для корпоративного навчання).

1.3. Порівняльна характеристика засобів розробки систем дистанційного навчання

Аналіз цільової аудиторії. На цьому етапі клієнт повинен вирішити, чому він планує використовувати додаток, яка є кінцевою метою розробки інструменту мобільного зв'язку з аудиторією.

Розробка узгодженого рішення. Перед початком розробки необхідно отримати технічне завдання (ТЗ) у замовника або надати короткий виклад для завершення та подальшої роботи над цим документом. Після отримання заповненого брифінгу та/або ТЗ ви можете приступити до створення прототипу і складання профілів для оцінки можливостей кінцевого продукту. Грунтуючись на баченні дизайнера, оцінці бізнесу та узгодженні деталей технічного завдання, ви можете розпочати процес розробки. Прототипи розроблені дизайнером та можуть бути як статичними, так і інтерактивними.

Статичні прототипи та інтерактивні макети повинні створюватися з урахуванням технічної та програмної бази, яку ви плануєте використовувати для створення програми. Написання коду та впровадження технологій. З готовим дизайном програма переходить до розробників: їм доводиться створювати веб додаток на основі мов програмування, фреймворків та різних технологій.

Тестування обов'язкове на різних етапах розробки додатків як на симуляторах, так і на реальних пристроях. Метою тестування є впевненість у тому, що взаємодія програми з апаратною та програмною платформою настільних комп'ютерів та ноутбуків буде точно такою, як передбачалося на етапі створення прототипу. Створити попередню версію. В результаті серії тестів та покращень програми має бути отримана робоча версія програми.

Саме ця версія буде додана на веб ресурс. Публікація програми на веб ресурсі - заключний етап роботи. Необов'язковий крок: подальша технічна підтримка та маркетингове просування додатку. За бажання клієнт може надати додаткові послуги, такі як технічна підтримка програми, подальший випуск нових версій для оновлення, а також маркетингове просування.

Наведемо огляд найпопулярніших систем дистанційного навчання [3-4]:

- Infotechno - система, що реалізується на сайті www.infotechno, розроблялася на основі аналізу найкращих систем навчання через інтернет, представлених на ринку (Прометей, Гіперметод, Moodle та ін.), і, отже, є найсучаснішою. При розробці системи були проведені консультації з викладачами та співробітниками провідних українських вишів з метою максимального обліку всіх вимог, які пред'являються освітніми установами до віртуального навчання. Система infotechno є першою в Україні між вузівською системою навчання через Інтернет.

- WebTutor – інформаційна система, призначена для автоматизації бізнес-процесів, пов'язаних із управлінням персоналом.

Інформаційна система WebTutor складається з модулів, кожен із яких забезпечує необхідний функціонал. У тому числі до складу модулів входить модуль “Дистанційне навчання”, що забезпечує планування, проведення та аналіз результатів навчання користувачів (співробітників компанії, клієнтів, студентів) за допомогою дистанційних курсів.

- Competentum Magister для навчальних закладів дозволяє створювати навчальні курси, планувати траєкторію навчання, використовувати компетентнісний підхід та персоналізувати навчальну програму учнів.

- eLearning Server – коробковий продукт eLearning Server дозволяє створити власний навчальний центр в Інтернет/Інтранет та організувати повний цикл дистанційного навчання.

Завдяки модульній архітектурі, відкритому коду та вбудованим засобам налаштування та інтеграції функціональність eLearning Server легко розширюється.

Має рекомендації Міністерства Освіти України та сертифікат Асоціації Дистанційного навчання України.

- Moodle – це середовище дистанційного навчання, призначене для створення якісних дистанційних курсів. Цей програмний продукт використовується більш ніж у 100 країнах світу університетами, школами, компаніями та незалежними викладачами. За своїми можливостями Moodle витримує порівняння з відомими комерційними системами управління навчальним процесом, водночас вигідно відрізняється від них тим, що поширюється у відкритих вихідних кодах - це дає можливість "заточити" її під особливості кожного освітнього проекту, доповнити новими сервісами.

- Adobe Connect — це спеціалізоване програмне забезпечення на базі технології Adobe Flash, розроблене для дистанційного навчання, проведення інтерактивних конференцій, нарад, онлайн підтримки клієнтів, обміну інформацією та інших не менш корисних функцій.

РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ДОДАТКУ

2.1. Технічне завдання на проєктування та розробку додатку

1. Цілі та завдання Веб-додатка. Розробка програми для підвищення ступеня лояльності студентів, співробітників та абітурієнтів, використовуючи зручний та простий спосіб доступу до інформації університету. Додаток повинен включати в себе інформацію про розклад занять групи та викладача, новини, дані співробітників університету та однокласників, успішність та посилання на офіційні електронні ресурси ВНЗ.

2. Мова реалізації. Програма буде реалізована українською мовою. Надалі можливе додавання англійської мови та німецької мов для іноземних студентів, співробітників та абітурієнтів університету.

3. Особисті нагадування. Повинен бути реалізований доступ студенту для додавання особистих нагадувань про роботи які необхідно виконати.

4. Стартовий екран. При завантаженні на екрані з'явиться інструкція з ознайомленням функціоналу програми або міститиме стрічку новин, завантажувану з бази університету.

5. Мова програмування. При написанні коду програми повинна використовуватися така мова як C++. Запити до БД будуть написані мовою PHP [10-11].

6. Розміщення програми на Веб-ресурсі. Програма повинна бути розміщена в мережі інтернет.

7. Дизайн. Ескізи, колір та стиль оформлення повинні відповідати корпоративному стилю університету.

2.2. Інструменти розробки додатку

Інтерфейс користувача повинен забезпечувати інтерактивну взаємодію за допомогою меню, що надає вибір будь-якої з реалізованих операцій у будь-якій

послідовності. Список операцій, що реалізуються, такий:

Введення даних;

Свята, що випали на вихідні;

Кількість свят, що збігається з вихідними;

Очистити екран;

Вивести усі дані;

Записати дані у файл;

Рахувати дані з файлу;

Про програму;

При реалізації необхідно передбачити перевірку на адекватність всіх даних, що вводяться - неприпустимі значення при введенні року, неприпустимі значення при зчитуванні даних з файлу.

Також для зручності необхідно реалізувати операцію очищення екрану, з наступним виведенням меню та останніх даних, введених користувачем.

Виконання операцій має супроводжуватися підказками та попередженнями. Як, наприклад, при виборі операцій «Введення даних», «Свята, які випали на вихідні», «Кількість свят, що збігається з вихідними», «Вивести всі дані», «Записати дані до файлу», «Рахувати дані з файлу», то за відсутності даних у програмі виводити відповідні повідомлення про помилку.

При реалізації необхідно передбачити перевірку на адекватність всіх даних, що вводяться – значення, що не відповідає допустимому діапазону при введенні року і виборі пункту меню.

Операції, що реалізуються в програмі, для зручності повинні виконуватися в окремому вікні, тому необхідно передбачити очищення екрана, при виборі деяких операцій з наступним виведенням меню на екран.

Буде корисним додати додатковий пункт у меню – «Про програму», прочитавши який користувач може ознайомитися з можливостями програми.

Вихід із програми здійснюється за допомогою вибору відповідного пункту меню «Вихід».

Для виконання курсового проекту було обрано середовище розробки програм для мов програмування C/C++ Visual Studio 2015, оскільки воно містить дуже зручний редактор, адаптований під:

- Windows 8.1
- Windows 8
- Windows 7 з пакетом оновлень 1 (SP1)
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 з пакетом оновлень 1 (SP1)

та має стильний, зручний інтерфейс російською мовою.

stdafx.h - служить для генерації файлу заголовків; до нього включено більшість стандартних і використовуваних у кожному додатку файлів, що включаються.

Для реалізації поставлених завдань були використані такі стандартні бібліотеки мови Cі:

- `stdio.h` – містить константи та оголошення функцій та типів, що використовуються для різних операцій стандартного введення та виведення, таких як:

- `fopen` – відкриття файлу для певних типів читання та запису;
- `fclose` - закриття файлу;
- `printf` – використовується для виведення стандартний потік виведення;
- `scanf` – використовується для введення із стандартного потоку введення;

- `fprintf` - використовується для виведення у файл;
- `fscanf` – використовується для введення файлу;

- `iostream` – заголовний файл із класами, функціями та змінними для організації введення-виведення в мові програмування C++.

Для досягнення поставленої мети було розроблено необхідний набір функцій:

Функція `null` варта введення початкових значень масиву. Опис функції:

```
void null (int x[])
```

Тип значення, що повертається – `void`, оскільки функція нічого не повертає. Формальні параметри: `int x[]` – передається одновимірний масив, що з елементів кожен із яких визначає день тижня певного свята. Повна реалізація функції представлена у додатку (листинг Би., рядки 7-16).

2) Функція `dayWeek` визначає день тижня для кожного зі свят на рік, введений користувачем. Опис функції:

```
void dayWeek (int g, int x[], int r)
```

Тип значення, що повертається – `void`, оскільки функція нічого не повертає. Формальні параметри: `int g` – рік, введений користувачем, `int x[]` – передається одновимірний масив, кожен із елементів якого задає день тижня певного свята, `int r` – змінна, яка вважає кількість високосних років, від 1970 до введеного користувачем.

3) Функція `weekend` призначена для виведення свят, які випали на вихідні, та виведення дат, на які вони були перенесені. Опис функції:

```
void weekend (int y[], char c[7] [33], char p[7] [9], int x[])
```

Тип значення, що повертається – `void`, оскільки функція нічого не повертає. Формальні параметри: `int y[]` – передається одновимірний масив, що складається з елементів, кожен із яких задає дату певного свята, `int x[]` – передається одновимірний масив, кожен із елементів якого задає день тижня певного свята,

`char c[7] [33]` – передається двовимірний масив, кожен із елементів якого має своє текстове значення, що складається з назви свята та місяця, в який він

святкується, `char p[7] [9]` – передається двовимірний масив, кожен із елементів якого має своє текстове значення, що складається із місяця певного свята.

4) Функція `numberHolidays` призначена для підрахунку кількості свят, які збігаються з вихідними. Опис функції:

```
void numberHolidays (int x[])
```

Тип значення, що повертається – `void`, оскільки функція нічого не повертає. Формальні параметри: `int x[]` – передається одновимірний масив, кожен із елементів якого задає день тижня певного свята.

5) Функція `input` призначена для введення користувачем року в межах від 2005 до 2099. Опис функції:

```
int input()
```

Тип значення, що повертається - `int`, оскільки за допомогою цієї функції в головну програму повертається номер року, введений користувачем.

Функція повертає значення `g`, яке надалі використовується для функцій: `dayWeek`, `print`, `fpr`, `leapYear`.

6) Функція `menu` призначена для виведення меню програми користувача. Опис функції:

```
void menu()
```

Тип значення, що повертається – `void`, оскільки функція нічого не повертає.

7) Функція `print` призначена висновку, введеного користувачем року. Опис функції:

```
void print (int g)
```

Тип значення, що повертається – `void`, оскільки функція нічого не повертає. Формальні параметри: `int g` – передається рік, введений користувачем.

8) Функція `fpr` призначена для запису даних у файл.

Опис функції:

```
void fpr (int x[], int g)
```

Тип значення, що повертається – `void`, оскільки функція нічого не повертає.
 Формальні параметри: `int g` – передається рік введений користувачем, `int x[]` – передається одновимірний масив, кожен із елементів якого задає день тижня певного свята.

9) Функція `fsc` варта зчитування, раніше використаних програмою, даних із файла. Опис функції:

`int fsc (int x[])`

Тип значення, що повертається – `void`, оскільки функція нічого не повертає.
 Формальні параметри: `int x[]` – передається одновимірний масив, кожен із елементів якого задає день тижня певного свята.

Функція повертає значення `g`, яке використовується для функцій: `dayWeek`, `print`, `fpr`.

10) Функція `leapYear` призначена підрахунку кількості високосних років від 1970 до введеного користувачем. Опис функції:

`int leapYear (int g)`

Тип значення, що повертається – `int`, оскільки за допомогою цієї функції в головну програму повертається кількість високосних років від 1970 до введеного користувачем. Формальні параметри: `int g` – передається рік, введений користувачем

11) Функція `about` призначена для виведення інформації про програму.

Опис функції:

`void about()`

Тип значення, що повертається – `void`, оскільки функція нічого не повертає

Реалізовані функції надалі використовуються в основному тілі програми – у головній функції, глобальні змінні якої:

`int ru` – номер операції з меню програми, що спочатку приймає значення 1.

`int r` – кількість високосних років від 1970 до введеного користувачем.

`int g` – рік, введений користувачем, спочатку приймає значення 0.

`int v` - рік від якого починається відлік, спочатку набуває значення 0.

`int x[7]` – дні тижня кожного зі свят, що спочатку приймає значення 3,1,7,5,6,5,3.

`int y[7]` – дні дат кожного зі свят, що приймає значення 7,23,8,1,9,12,4.

`char c[7] [33]` – містить текстові значення назви свят та місяців їх дат, приймає значення «січня різдво», «лютий водохреща», «березень жіночий день», «квітня день пам'яті культури та історії», «травня день героїв України», «червень день Конституції», «липня день родини».

`char p[7] [9]` – містить текстові значення місяців кожного зі свят, набуває значень «січня», «лютого», «березня», «квітня», «травня», «червня», «липня».

`FILE* file` – покажчик змінної файлового типу;

Вихід із програми здійснюється вибором пункту меню «0».

Детальний опис роботи деяких функцій:

dayWeek

void dayWeek (int g, int x[], int r)

{

int i;

g = g - 1970;

for (i = 0; i < 7; i++)

x[i] = x[i] + g + r;

for (i = 0; i < 7; i++)

if (x[i] > 6)

x[i] = x[i]%7;

}

До кожного з значень днів тижня свят 1970 додаються різниця введеного і 1970, і кількість високосних років за проміжок від 1970 до введеного

користувачем. Далі для знайдених значень (великих 7) знаходиться залишок від розподілу на 7, т.к. значення більше 6 не відповідають значенням днів тижня (1 – пн., 2 – вт., 3 – ср., 4 – чт., 5 – пт., 6 – сб., 0 – нд.).

leapYear

int leapYear (int g)

{

int r = 0, v = 1972;

r = ((g - v) / 4) + 1;

return r;

}

Щоб знайти у високосних років, різницю між введеним користувачем і 1972 роками ділиться на 4, і до неї додається 1, т.к. високосний 1972 не входить у цю різницю.

fpr

void fpr (int x[], int g)

{

int i;

FILE * fp;

fopen_s (&fp, "file.txt", "wb");

if (fp)

{

for (i = 0; i<7; i++)

fwrite (&x[i], sizeof (x[i]), 1, fp);

fwrite (&g, sizeof(g), 1, fp);

}

Web-додаток і двох основних частин. Серверної частини, на якій обробляються всі запити до бази даних та клієнтської частини, що відповідає за

відображення користувачеві структури програми та даних, які він запитує.

Nginx web-сервер – самостійний HTTP-сервер. Цей web-сервер добре підходить для високонавантажених проектів.

Можливості Nginx web-сервера:

- обслуговування статичних запитів, індексних файлів, автоматичне створення списку файлів;
- розподіл навантаження та відмовостійкість;
- модульність, у тому числі стиснення (gzip), докачування (byte-ranges), перетворення зображень;
- підтримка HTTP/2 пріоритезація на основі ваги і залежностей;
- налаштування форматів логів, швидка ротація логів;
- спеціальні сторінки помилок;
- обмеження доступу залежно від адреси клієнта.

При розробці web-програми Календар Nginx використовувався як основний web-сервер. Для прискорення процесу розробки для конфігурації Nginx використовувався Vagrant.

Vagrant - засіб, що дозволяє значно простіше і швидше створити нову віртуальну машину. Vagrant є конфігураційним файлом, який автоматично розгортається у внутрішньому оточенні нової віртуальної машини. Після цього з'являється можливість з мережевого інтерфейсу віртуальної машини перенаправити порти і виділити IP-адресу, а також встановити потрібне програмне забезпечення. Всі налаштування вказуються у файлі Vagrant.

Налаштування Vagrant файлу почалося із вказівки локальних доменних імен, для цього у файлі прописувався наступний код:

```
domains = {  
  backend: 'backend.organizer.dev', frontend: 'organizer.dev',  
}
```

Після цього вказуються посилання файли конфігурацій для віртуальної машини.

```
config = {  
  local: './vagrant/config/vagrant-local.yml',  
  example: './vagrant/config/vagrant-local.example.yml'  
}
```

У файлах конфігурацій були прописані основні налаштування віртуальної машини:

- GitHub Token – унікальний ключ розробника, без наявності якого неможливо вести розробку проекту на платформі Yii2;

- налаштування серверного часу;
- перевірки на поновлення;
- ім'я віртуальної машини;
- ір-адреса віртуальної машини;
- кількість процесорних ядер віртуальної машини;
- оперативної пам'яті віртуальної машини

.github_token: 442b21b09dec39da5794a2b68c5fb3faf9530628 timezone: UTC

box_check_update: false machine_name: organizer ip: 192.168.83.138

cpus: 1

memory: 512

Після запуску команди Vagrant UP буде створено нову віртуальну машину з необхідними параметрами та програмним забезпеченням.

Для роботи з базами даних при розробці web-програми "Календар" на базі фреймворку Yii2, використовувалися міграції. Міграції баз даних дозволяють створювати нові таблиці з налаштуванням всіх необхідних зв'язків, так і змінювати вже існуючі таблиці, додавати нові поля і редагувати зв'язки між

таблицями в базі даних.

Для створення нової міграції використовується команда:

php yii migrate/create <name>

Команда прописується в консолі, після чого створюється файл із міграцією.

Для виконання мігацій використовується команда:

php yii migrate

Інформація про всі виконані міграції записується в окрему таблицю бази даних.

Файл міграцій містить у собі код, що відповідає за створення таблиць, їх видалення або внесення змін до вже існуючої таблиці.

Для створення в базі даних таблиці, що містить дані про події та заходи, що відображаються в календарі, була написана міграція в коді якої міститься створення таблиці із зазначенням найменування полів типу даних і правил заповнення.

```
$this->createTable('{{%event}}', [ 'id' => $this->primaryKey(),
'idUser' => $this->integer()->notNull(), 'title' => $this->string(64)->notNull(),
'allDay' => $this->boolean(),
'start' => $this->dateTime()->notNull(), 'end' => $this->dateTime(),
'description' => $this->text(), 'created_at' => $this->integer(), 'updated_at' =>
$this->integer()
], $ tableOptions);
```

Після чого в коді міграції прописується зв'язок між полем, що містить у собі інформацію про особистий номер користувача, який створив подію та таблицею користувачів, полем, що містить особистий номер користувача.

```
$this->addForeignKey( 'fk-event-idUser', '{{%event}}', 'idUser',
'{{%user}}', 'id'
);
```

Gii представлений у Yii2 як модуль. За допомогою даного модуля були згенеровані моделі, контролери та подання для подальшої роботи з ними. Gii генерує шаблон коду для подальшого редагування. Приклад створення моделі таблиці завдань представлений рисунку 2.1.

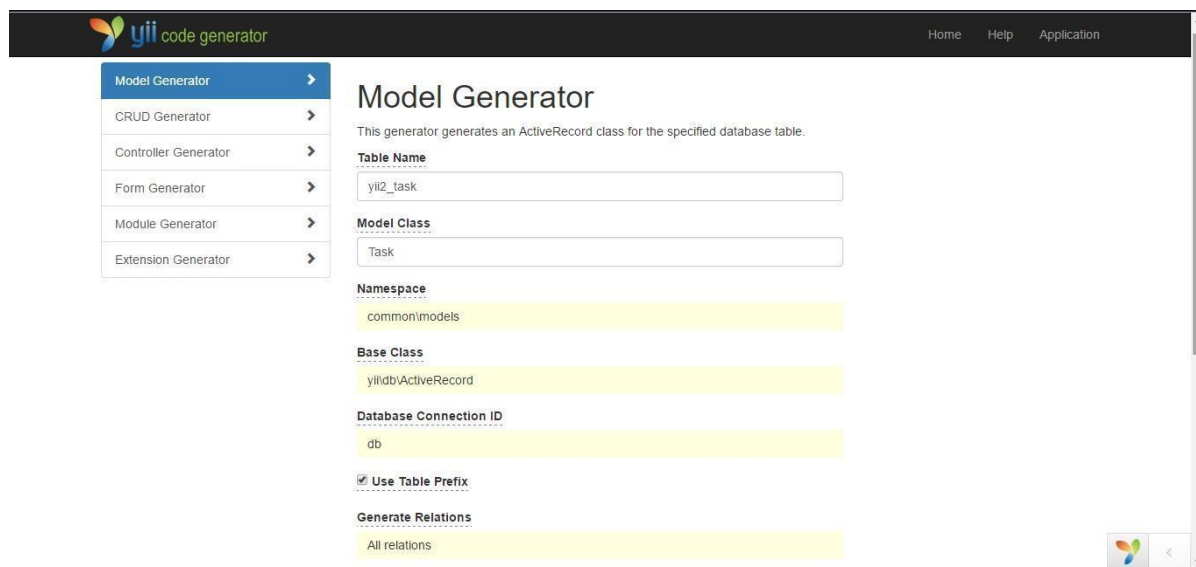


Рисунок 2.1. - Створення моделі за допомогою Gii

Для створення моделі викликається модуль Gii шляхом додавання адресного рядка браузера після доменного імені нашого проекту "\Gii". Вхідні дані необхідні для створення моделі:

- ім'я таблиці;
- ім'я моделі;
- адресу розташування файлу моделі;
- ім'я базового класу функції та інші додаткові настройки.

Bootstrap 3 - Фреймворк, що дозволяє створювати адаптивні, сучасні, крос-браузерні та стандартизовані інтерфейси. Даний Фреймворк відмінно підходить для створення адаптивних шаблонів, починаючи від широкоекранних моніторів до екранів мобільних пристроїв з не великою роздільною здатністю.

У файлах можна побачити наступну структуру та зміст. Всі файли логічно

згруповані за загальними властивостями і містять обидві версії: мінімізовану та компільовану.

На рисунку 2.2 наведено структуру файлів Bootstrap 3.

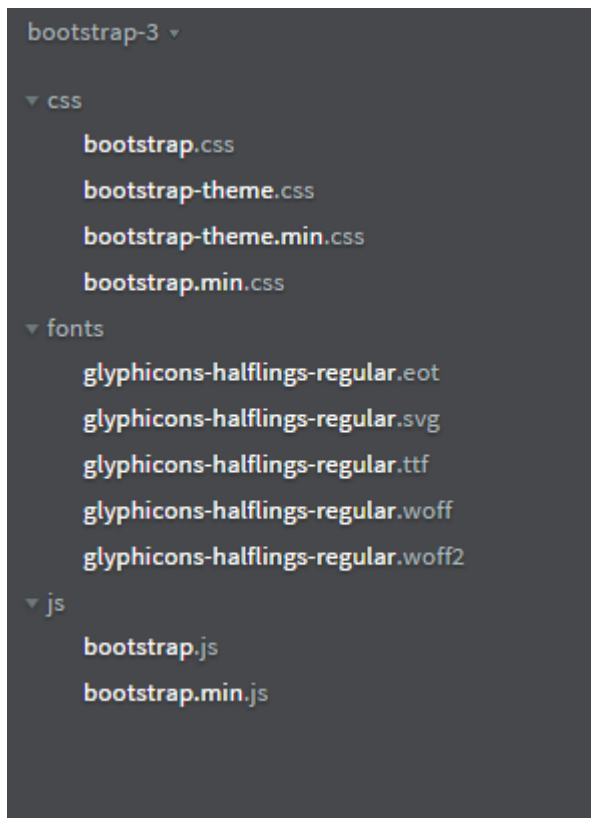


Рисунок 2.2. - Структура Bootstrap

Bootstrap 3 використовувався при створенні всього інтерфейсу web-програми. Для розміщення вікна з полями введення для авторизації на сайті використовувався наступний код.

```
<div class="col-lg-offset-4 col-lg-4 col-md-offset-4 col-md-4" id="login_form  
common">
```

Фрагмент цього коду відповідає за виділення на середніх та великих моніторах 4 колонки з 12, з відступом ліворуч 4 колонки, під вікно авторизації у web-додатку.

Для початку роботи з web-програмою "Календар" користувачеві необхідно

пройти реєстрацію. Для реєстрації в програмі вказуються основні дані користувача, його прізвище ім'я по батькові, номер телефону, адресу електронної пошти та пароль від облікового запису.

```
Yii::$app->response->format = \yii\web\Response::FORMAT_JSON; return  
ActiveForm::validate($model);  
}  
if ($model->load(Yii::$app->request->post())) {  
    $model->validate();  
    if ($user = $model->signup()) {  
        if (Yii::$app->getUser()->login($user)) { return $this->goHome();  
        }}}}  
return $this->render('signup', [ 'model' => $model,  
]);  
}
```

При натисканні на кнопку реєстрації викликається нове уявлення, яке підвантажує форму реєстрації нового користувача, з усіма необхідними полями. Після заповнення даних у поля відбувається автоматична перевірка коректності введених даних, викликається функція валідації форми, яка перевіряє наявність даних у полях зазначених у моделі як "необхідні для заповнення". Після валідації та натискання на кнопку "Зареєструватися" відбувається відправлення даних на контролер, де всі дані обробляються та відправляються до бази даних.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ

3.1. Функціонал додатку електронного навчання «Органайзер»

Функціонал web-додатку "Органайзер" ділиться на три основні напрямки:

- робота із завданнями - створення нових завдань, видалення та зміна завдань, призначення виконавця та термінів виконання завдання;
- робота з подіями - створення нової події у календарі на робочому столі web-додатка, перегляд та видалення події;
- робота зі співробітниками компанії - при реєстрації компанії в web-додатку "Календар", у директора компанії з'являється можливість прив'язувати інші акаунти як працівники, згодом призначати для них завдання. Також є можливість прив'язувати інші акаунти як секретарі.

У додатку передбачено 4 типи облікових записів:

- індивідуальний - обліковий запис для індивідуального використання, на робочому столі є особистий календар та список завдань, у головному меню присутні пункти для реєстрації власної компанії та перегляду пропозицій про співпрацю в інших компаніях як працівник або секретар. Індивідуальний обліковий запис є стартовим, будь-який користувач, який пройшов реєстрацію у web-додатку, отримує обліковий запис зі статусом "індивідуальний";
- директор - статус облікового запису змінюється з "індивідуальний" на "директор" за умови, що користувач, що працює під цим акаунтом, здійснить реєстрацію своєї компанії в web-додатку. У головному меню перестає відображатися пункт "пропозиції" і з'являється новий пункт "компанія", в якому користувач може переглядати список співробітників своєї компанії, при необхідності надсилати пропозиції щодо роботи для нових співробітників або звільняти старих. На робочому столі користувача так само відображається календар з подіями, і список завдань, але тепер є функція призначення виконавця для завдання, якого можна вибрати зі списку співробітників компанії. Користувач

акаунту зі статусом "директор", так само може закріпити за собою особистого секретаря, який матиме доступ до календаря подій директора та мати можливість створювати та видаляти події;

- працівник - акаунту надається статус "працівник" за умови, що користувач акаунту зайдет у вкладку пропозицій про співробітництво та прийме пропозиції від однієї з компаній на посаду працівника. При використанні облікового запису "працівник" у головному меню перестають відображатися пункти меню для реєстрації власної компанії або перегляду пропозицій про співпрацю, замість них з'являється пункт меню "компанія", в якому користувач має можливість закріпити за собою свого особистого секретаря, надіславши запит на роботу для якої -або облікового запису. На робочому столі користувача відображається календар із подіями, список складених завдань та список завдань, для яких директор компанії призначив його виконавцем;

- секретар - статус облікового запису змінюється на "секретар" за умови, коли користувач зайдет у вкладку пропозицій про співробітництво та прийме пропозицію від однієї з компаній на посаду секретаря. В обліковому записі зі статусом "секретар" у головному меню зникають пункти, які дозволяють зареєструвати в додатку власну компанію або прийняти пропозицію про співпрацю від інших компаній. На робочому столі секретаря відображається календар безпосереднього начальника, тобто користувача, за яким закріплений даний секретар. Користувач має доступ до редагування подій, створення нових подій та їх видалення.

- Для реєстрації власної компанії користувач вказує назву компанії, короткий опис компанії та контактний телефон. Після реєстрації з'являється пункт меню "компанія", в якому директор компанії може переглядати список співробітників компанії. У цьому розділі користувач може надсилати пропозиції про роботу або звільняти старих співробітників. Подання вікна

реєстрації, як і інші уявлення, побудовані на основі фреймворку bootstrap 3.

Після заповнення всіх полів у вікні реєстрації компанії дані відправляються на контролер. Для обробки даних на контролері було написано таку функцію.

```
public function actionCreate()
{
$model = new Company();
$modelWork = новий Worker();
$user = \Yii::$app->user->identity;
if ($model->load(Yii::$app->request->post())) {
$model->user_created = $user->id; if ($model->save()){
$modelWork->user_id = $user->id;
$modelWork->company_id = $model->id;
$modelWork->user_role = 'director';
$modelWork->save();
$user->role = $this->nextRoleUser; if ($user->save()) {
return $this->redirect('/site/index');
}
}
} else {
return $this->render('create', [ 'model' => $model,
```

Після створення моделей відбувається передача отриманих даних із подання у модель компанії, після чого модель компанії зберігається до бази даних. Наступним кроком відбувається запис усіх необхідних даних у модель працівника, таких як:

- ID працівника - особистий номер співробітника, у разі дорівнює ID користувача під аккаунтом якого відбувалася реєстрація компанії;

- ID компанії - ідентифікатор компанії, у яку влаштувався працівник;
- Посада співробітника - у разі посаду присвоюється " директор " .

Після запису всіх необхідних даних відбувається збереження бази даних. Таким чином, при створенні компанії, у таблиці "компанії" з'являється запис про зареєстровану компанію, а в таблиці "працівники" створюється запис про першого працівника, на посаду якого "директор".

Додавання нових користувачів у компанію відбувається на запрошення, директор компанії вводить дані людини, яку він хоче запросити на роботу до своєї компанії, у базі даних створюється запис із пропозицією про роботу та даними про співробітника. Після цього, користувач якому було надіслано запрошення на роботу може або прийняти його або видалити. Якщо користувач приймає запрошення, його статус змінюється на той, який йому запропонували в запрошенні і в базу даних заносяться дані про нового працівника компанії. Для реалізації обробки даних про нового користувача компанії в контролері був прописаний наступний код.

```
public function actionTakeOffer(){
    $user = \Yii::$app->user->identity;
    if($offer_id = \Yii::$app->request->post('offer_id')){
        $offer = RequestCompanyUser::find()->where(['id' => $offer_id])->one();
        if($offer){
            $worker = новий Worker();
            $worker->company_id = $offer->company_id;
            $worker->user_id = $user->id;
            $worker->user_role = $offer->role_improver; if($worker->save()){
                $userChild = новий UserChild();
                $userChild->user_parent_id = $offer->user_request_id;
                $userChild->user_child_id = $user->id; if($userChild->save()){
```

```

$user->role = $offer->role_improver; if($user->save()){
if($offer->delete()) return $this->redirect('index');
if($offer->delete()) return $this->redirect('index');
}}}}
return 0;}

```

Вище представлений код, який реалізує додавання до бази даних нового запису до таблиці "працівники", після чого створюється новий запис у таблиці "начальники та працівники" та змінюється статус облікового запису користувача.

3.2. Інструкція для користувача додатком «Органайзер»

Список завдань - дозволяє переглядати список складених задач, присутні функції детального перегляду задач, видалення чи зміна завдань. Для облікового запису, що має статус "директор", з'являється функція призначення виконавця для кожного із завдань, виконавця можна вибрати зі списку співробітників компанії. Для облікового запису має "працівник" з'являється додатковий список, у яких директор компанії призначив його виконавцем.

Для реалізації функціоналу додавання нового завдання було написано наступний код.

```

public function actionCreate()
{
$model = new Task();
$user = \Yii::$app->user->identity; if (!\Yii::$app->request->isAjax){
$model->load(\Yii::$app->request->post());
$model->user_create_id = $user->id; if($model->save()) {
return $this->redirect(['/event/index', 'id' => $model->id]);
}}}

```

Представлений вище код реалізує отримання даних з форми створення нового завдання та створює запис про нове завдання у базі даних.

Форма створення нового завдання підвантажується в модальне вікно, модальне вікно написано за допомогою фреймворкових засобів bootstrap 3.

Приклад відображення форми створення нового завдання представлено рисунку 2.1.

Для реалізації функціоналу, який дозволяє відображати календар і позначені на ньому події, використовувався готовий віджет "FullCalendar".

Форма додавання нової події на календар так само, як і при створенні нового завдання, відображається в модальному вікні.

Календар має два режими відображення подій "відображати місяць" та "відображати тиждень".

Приклад відображення завдань на календарі в режимі "Відображати місяць" наведено на рисунку 3.1.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Рис. 3.1. - Приклад відображення завдань на календарі у режимі "Відображати місяць"

Приклад відображення завдань на календарі в режимі відображати тиждень представлений на рисунку 13.

	Sun 6/11	Mon 6/12	Tue 6/13	Wed 6/14	Thu 6/15	Fri 6/16	Sat 6/17
all-day							
1pm							

Рис. 3.2. - Приклад відображення завдань на календарі

в режимі «Відображати тиждень»

Основне розбиття можливостей користувачів відбувається у головному меню. Залежно від типу облікового запису в головному меню відображатимуться лише пункти меню, доступні для цього типу облікового запису. Для реалізації головного меню з перевіркою типу облікового запису було написано наступний код.

```
<?php
if (!\Yii::$app->user->isGuest){
    $user = \Yii::$app->user->identity;
    $user_privilege = $user->getRole(

if($user_privilege === 'director'){
    $label_company = ['label' => 'Компанія', 'url' => ['/company/index']];
    $label_request_company_user = '';
    $label_desktop = ['label' => 'Робочий стіл', 'url' => ['/event/index']];
    }
if($user_privilege === 'individual'){
    $label_company = ['label' => 'Створити компанію', 'url' =>
['/company/create']];
    $label_request_company_user = ['label' => 'Пропозиції', 'url' =>
['/requestcompanyuser/index']];
    $label_desktop = ['label' => 'Робочий стіл', 'url' => ['/event/index']];
    }
if($user_privilege === 'secretary'){
    $label_company = '';
    $label_request_company_user = '';
```

```
$label_desktop = ['label' => 'Робочий стіл', 'url' => ['/event/index']];  
}  
if($user_privilege === 'worker'){  
$label_company = ['label' => 'Компанія', 'url' => ['/company/index']];  
$label_request_company_user = '';  
$label_desktop = ['label' => 'Робочий стіл', 'url' => ['/event/index']];  
}}else{  
  
$label_company = '';  
$label_desktop = '';  
$label_request_company_user = '';
```

ВИСНОВКИ

Застосування розробленого програмного комплексу на вирішення завдань оперативного планування робочого дня і розподілу завдань дозволило оптимізувати роботу як окремого користувача, і цілої компанії, створивши єдиний загальний оперативний простір.

Розроблено проста та зручна програма-Календар. Яка допоможе скласти список справ на день/тиждень/місяць, позначити список завдань, розподілити завдання між співробітниками.

Програма реалізована у вигляді мережевого додатку, отже підвищується мобільність користувачів та доступність використання.

Мережеве рішення, призначене для забезпечення обміну даними та використання загальних ресурсів компанії, має бути відкритим – повинно підтримувати безліч різноманітних апаратних засобів, мережевих протоколів. Запорукою ефективності його роботи є орієнтація застосування відкритих, кросс-платформних технологій.

Основною перевагою цього додатка в порівнянні з аналогами є відсутність довгих і трудомістких налаштувань перед початком роботи, абсолютно безкоштовне використання, не містить надмірного функціоналу. Отримані результати можуть бути використані для оперативного прийняття рішення про виконання завдань, що надійшли.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вступ до програмування мовою C++. Організація даних / Т. О. Карнаух, Ю. В. Коваль, М. В. Потієнко, А. Б. Ставровський. К.: ВПЦ "Київський університет", 2015.
2. Вступ до програмування мовою C++. Структури даних / Р.А. Веклич, Т.О. Карнаух, А. Б. Ставровський. К.: ВПЦ "Київський університет", 2018.
3. Гладченко О., Ратушняк Т., Беспарточна О. Сучасні інформаційно-комунікаційні технології для забезпечення якісного дистанційного навчання // Теорія і практика управління соціальними системами. Розділ: Інформаційні технології і засоби навчання в професійній підготовці майбутніх фахівців. – 2022. – № 2. – С.76–91. doi: 10.20998/2078-7782.2022.2.06
4. Голощук Р. О. Особливості впровадження технологій та методів дистанційного навчання в освітній процес // Вісн. Нац. ун-ту "Львів. політехніка". - 2015. - № 829.
5. Документація Docker Hub [Електронний ресурс] — Режим доступу: <https://docs.docker.com/docker-hub/>
6. Ратушняк Т. В. Програмування мовою JAVA: практикум: навчальний посібник. Державна фіскальна служба України, Університет державної фіскальної служби України. – Ірпінь, 2017. – 212 с.
7. Теоретико-методологічні основи комп'ютерних баз знань в економіці: монографія / Ріппа С.П., Погореловська І.Д., Редич О.В. та ін. ; за ред. С.П. Ріппи. – Ірпінь: УДФСУ, 2021. – 170 с.
8. Хорець В.О. Мобільний додаток “Органайзер студента”/ Хорець В.О., Шушура О.М. // II Всеукраїнська науково-практична інтернет-конференція

молодих вчених та здобувачів вищої освіти «Сучасна молодь в світі інформаційних технологій» — Херсон, 2021.

9. Analytical information on trends in the organization of distance learning in institutions of professional higher and higher education during quarantine in the 2020/2021 academic year (according to the results of online surveys of participants in the educational process). available at : https://ru.osvita.ua/doc/files/news/825/82561/1_ANALITICHNA_DOVIDKA_O_pituvannya_FPO_ZVO_DS.pdf (дата звернення: 11.11.2021)
10. Beckman, Brian (1994), Theory of Spectral Graph Layout, Tech. Report MSR-TR-94- 04, Microsoft Research
11. Go vs C++ Compared and Contrasted. (2019). Режим доступу:: <https://career karma.com/blog/go-vs-c-plusplus/#go-vs-c++-speed-and-readability>
- 12.9. Pakhomova, T. O., Komova, O. S., Belia, V. V., Yivzhenko, Y. V., Demidko, E. V. (2021). “Transformation of the pedagogical process in higher education during the quarantine”. Linguistics and Culture Review. no. 5, pp. 215–230. available at : <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85116857650&doi=10.37028%2flingcure.v5nS2.1341&partnerID=40&md5=174aa4c124d3cd2704056e461efc4bf2> DOI: 10.37028/lingcure.v5nS2.1341
13. PHP Manual. PHP: веб-сайт. URL: <https://www.php.net/manual/en/langref.php#langref> 2. PHP. Вільний шлях.
14. PHP. Вільний шлях : веб-сайт. URL; <http://iflista.github.io/php-the-right-way/>
3. Вивчення php з нуля | Уроки php для початківців. PHP : веб-сайт. URL: <https://phpbuilder.ru/ua/learn>
15. Pro Git 2nd Edition [Електронний ресурс] / Scott Chacon, Ben Straub — 2014 — Режим доступу: <https://git-scm.com/book/en/v2>

16. Top 10 Most Popular Programming Languages (2020). Режим доступа:
<https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/>
17. Tytenko S. V. INTERACTIVE CONCEPT MAPS IN ONTOLOGY-ORIENTED INFORMATION AND LEARNING WEB-SYSTEMS. KPI Science News, no. 2, pp. 24– 36, 2019. doi:10.20535/kpi-sn.2019.2.167515

ДОДАТОК А. Код програми

```
#include "stdafx.h"
#include <iostream>
void null (int x[])
{
    x[0] = 3;
    x[1] = 1;
    x[2] = 7;
    x[3] = 5;
    x[4] = 6;
    x[5] = 5;
    x[6] = 3;
}
void dayWeek (int g, int x[], int r)
{
    int i;
    g = g - 1970;
    for (i = 0; i < 7; i++)
        x[i] = x[i] + g + r;
    for (i = 0; i < 7; i++)
        if (x[i] > 6)
            x[i] = x[i] % 7;
}
void weekend (int y[], char c[7] [33], char p[7] [9], int x[])
{
    int i;
    for (i = 0; i < 7; i++)
```

```

{
if (i == 2)
{
if (x[i] == 6)
printf («Випав на вихідний -%d %s, перенесений на понеділок %d %s і
на вівторок % d % s\n», y[i], c[i], y[i] + 2, p[i], y[i] + 3, p[i]);
if (x[i] == 0)
printf («Випав на вихідний -%d %s, перенесений на понеділок %d %s і
на вівторок % d % s\n», y[i], c[i], y[i] + 1, p[i], y[i] + 2, p[i]);
}
else
{
if (x[i] == 6)
printf («Випав на вихідний -%d %s, перенесений на понеділок %d
%s\n», y[i], c[i], y[i] + 2, p[i]);
if (x[i] == 0)
printf («Випав на вихідний -%d %s, перенесений на понеділок %d
%s\n», y[i], c[i], y[i] + 1, p[i]);
}
}
}

void numberHolidays (int x[])
{
int kol, i;
for (kol = 0, i = 0; i<7; i++)
if ((x[i] == 6) || (x[i] == 0))
kol++;

```

```

printf («Кількість свят, що збігається з вихідними =%d\n»,
kol);
}

void menu()
{
printf («\nВиберіть один з пунктів меню:\n1.введення даних\n2.свята,
які
випали на вихідні\n3. Кількість свят, що збігається з
вихідними\n4.очистити екран\n5.вивести всі дані\n6.записати дані в
файл\n7.рахувати дані з файлу\n8.про програму\n0.вихід\n»);
}

void print (int g)
{
system («cls»);
menu();
if (g! = 0)
printf ("%d\n", g);
}

int input()
{
int g, k = 1;
while(k)
{
printf («Введіть рік з 2015 по 2099»);
scanf_s (%d, &g);
if ((g > 2004) && (g < 2100))

```

```

{
k = 0;
print(g);
}
else
print
;
}
return g;
}
void fpr (int x[], int g)
{
int i;
FILE * fp;
fopen_s (&fp, "file.txt", "wb");
if (fp)
{
for (i = 0; i<7; i++)
fwrite (&x[i], sizeof (x[i]), 1, fp);
fwrite (&g, sizeof(g), 1, fp);
}
fclose(fp);
}
int fsc (int x[])
{
int i = 0, g = 0;
FILE * fp;

```

```
fopen_s (&fp, "file.txt", "rb");
if (fp)
{
while (i! = 7)
{
if ((fread (&x[i], sizeof (x[i]), 1, fp)) == 1)
i++;
else
{
printf (« n»);
return 0;
}
if ((i == 7) && ((fread (&g, sizeof(g), 1, fp))!= 1))
return 0;
}
}
}
fclose(fp);
printf («Зчитування даних із файлу відбулося успішно.\n»);
return g;
}
int leapYear (int g)
{
int r = 0, v = 1972;
r = ((g - v) / 4) + 1;
return r;
}
```

```

void about()
{
printf («Електронний додаток виробничий календар, який
зазначеному році визначає кількість збігів основних свят»
«з вихідними днями виводить дати цих днів, а також дати на які будуть
перенесено вихідні.\nРобота була виконана: Бормісов Д.В., група 606–41,
2016
рік.\n»);
}
int _tmain (int argc, _TCHAR* argv[])
{
setlocale (LC_ALL, "rus");
int pu=1, r, g=0, v=0, x[7] = {3,1,7,5,6,5,3}, y[7] = {7,23,8,1, 9,12,4};
char c[7] [33] = {«січня різдво», «лютий день захисника
вітчизни», «березня жіночий день», «травня весни і праці», «травня
перемоги», «червень день Росії», «листопад день народної єдності»};
char p [7] [9] = {"січня", "лютого", "березня", "травня", "травня", "червня",
"листопада"};
menu();
while(pu)
{
scanf_s (%d, &pu);
switch (pu)
{
case 1:
print(g);
null(x);

```

```

g = input();
r = leapYear(g);
dayWeek (g, x, r); break;
case 2:
if ((g > 2004) && (g < 2100))
weekend (y, c, p, x);
else
; break;
case 3:
if ((g > 2014) && (g < 2100))
numberHolidays(x);
else
break;
case 4:
print(g); break;
case 5:
if ((g > 2014) && (g < 2100))
{
printf («Пик -%d\n», g);
weekend (y, c, p, x);
numberHolidays(x);
}
else
break;
case 6:
if ((g > 2014) && (g < 2100))
{

```

```
fpr(x, g);  
printf («Запис даних у файл відбувся успішно.\n»);  
}  
else  
printf ; break;  
case 7:  
g = fsc(x);  
r = leapYear(g);  
break;  
case 8:  
about();  
system («pause»);  
print(g);  
break;  
}  
}  
return 0;  
}
```