

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

Автоматизація процесу проектування меблів для виробничого
підприємства «ВіЯр» м. Київ

Студента 4 курсу, 10 групи,
напрям підготовки 6.050103
спеціалізації «Програмна
інженерія»

Нескород Аркадій
Олегович

—
підпис студента

Науковий керівник
кандидат технічних наук
доцент кафедри програмної
інженерії та кібербезпеки

Рзаєва Світлана
Леонідівна

—
підпис керівника

Гарант освітньої програми
кандидат технічних наук
доцент кафедри програмної
інженерії та кібербезпеки

Цензура Микола
Олександрович

—
підпис керівника

Київський національний торговельно-економічний університет

Факультет _____ ФОАІС _____ Кафедра _____
програмної інженерії та кібербезпеки

Освітній ступінь _____ бакалавр

Спеціальність _____ 6.050103 програмна інженерія

Спеціалізація / освітня програма _____

Затверджую

Зав. кафедри _____

"__" "__" 20__ р.

Завдання на випускний кваліфікаційний проект студентів

Нескорода Аркадія Олеговича

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту Автоматизація процесу проектування меблів для виробничого підприємства «ВіЯр» м. Київ

Затверджена наказом ректора від "__" "__" 20__ р. № _____

2. Строк здачі студентом закінченої проекту _____

3. Цільова установка та вихідні дані до проекту

Мета проекту розробка моделей даних для зберігання інформації про мебельні конструкції, нормалізація цих моделей, побудова правильних зв'язків між таблицями, які забезпечуть правильне заповнення таблиць бази даних без логічних помилок, а також розробка основного функціоналу web-додатку, який реалізуватиме можливість конструювання простих мебельних конструкцій для виробничого об'єднання «ВіЯр».

Об'єкт дослідження процес проектування меблів

Предмет дослідження автоматизація процесу проектування меблів

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

Вступ

РОЗДІЛ 1 ХАРАКТЕРИСТИКА КОМПАНІЇ ТМ «ВІЯР»

1.1. Опис структури компанії ТМ «ВіЯр»

1.2. Розробка технічного завдання

1.3 Висновки до розділу 1

РОЗДІЛ 2 ХАРАКТЕРИСТИКА ПРОГРАМНО-ТЕХНІЧНОЇ БАЗИ

2.1. Аналіз технічної бази компанії «ВіЯр»

2.2. Розробка БД для компанії ТМ «ВіЯр»

2.3. Висновки до розділу 2

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ ПРОЦЕСУ

ПРОЕКТУВАННЯ МЕБЛІВ ДЛЯ ВИРОБНИЧОГО ПІДПРИЄМСТВА
«ВІЯР»

3.1 Визначення цілі проекту

3.2. Розробка API доступу до БД

3.3 Побудова графічного інтерфейсу клієнтського режиму

3.3.1. Опис технології розробки графічного інтерфейсу

3.3.2. Реалізації графічного інтерфейсу

3.4. Розробка логіки тривимірної сцени

3.4.1. Опис технології розробки

3.4.2. Реалізація тривимірної сцени

3.5. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

Додаток 1: Лістинг конфігурація для webpack 4

Додаток 2: Лістинг правил для webpack 4

Додаток 3: Лістинг збору модулів THREE.js

Додаток 4: Лістинг модулю побудови сцени THREE.js

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4

7. Дата видачі завдання « ____ » _____ 20__ р.

8. Науковий керівник випускного кваліфікаційного проекту

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

(*nidnuc*, *dama*)

$$\frac{(nidnuc, \partial ama)}{(PIB, nidnuc, \partial ama)}$$

Випускний кваліфікаційний проект студента

Гарант освітньої програми _____

(прізвище, ініціали, підпис)

(підпис, прізвище, ініціали)

« » 20 p.

Зміст	
ВСТУП.....	3
РОЗДІЛ 1 ХАРАКТЕРИСТИКА КОМПАНІЇ ТМ «ВІЯР».....	5
1.1. Опис структури компанії ТМ «ВіЯр».....	5
1.2. Розробка технічного завдання.....	6
1.3 Висновки до розділу 1.....	9
РОЗДІЛ 2 ХАРАКТЕРИСТИКА ПРОГРАМНО-ТЕХНІЧНОЇ БАЗИ.....	11
2.1. Аналіз технічної бази компанії «ВіЯр».....	11
2.2. Розробка БД для компанії ТМ «ВіЯр»	12
2.3. Висновки до розділу 2.....	13
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ ПРОЦЕСУ ПРОЕКТУВАННЯ МЕБЛІВ ДЛЯ ВИРОБНИЧОГО ПІДПРИЄМСТВА «ВІЯР»	15
3.1 Визначення цілі проекту.....	15
3.2. Розробка АРІ доступу до БД	17
3.3 Побудова графічного інтерфейсу клієнтського режиму.....	19
3.3.1. Опис технології розробки графічного інтерфейсу.....	19
3.3.2. Реалізації графічного інтерфейсу.....	20
3.4. Розробка логіки тривимірної сцени.....	22
3.4.1. Опис технології розробки.....	22
3.4.2. Реалізація тривимірної сцени.....	23
3.5. Висновок до розділу 3	24
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	25
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	26
ДОДАТКИ.....	27
Додаток 1: Лістинг конфігурація для webpack 4	27
Додаток 2: Лістинг правил для webpack 4.....	29
Додаток 3: Лістинг збору модулів THREE.js.....	30
Додаток 4: Лістинг модулю побудови сцени THREE.js	32

					<i>КНТЕУ 6.050103 10-09.БР</i>				
					<i>Автоматизація процесу проектування меблів для виробничого підприємства «ВіЯр» м. Київ</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		Зміст	2	26	
.Зав. каф.		Криворучко О.В.				<i>Зміст</i>	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
.Керівник		Рзасва С. Л.							
Гарант		Цензура М. О.							
Розроб		Нескород А. О.							

ВСТУП

Метою цього випускного кваліфікаційного проекту, є розробка моделей даних для зберігання інформації про мебельні конструкції, нормалізація цих моделей, побудова правильних зв'язків між таблицями, які забезпечуть правильне заповнення таблиць бази даних без логічних помилок, а також розробка основного функціоналу web-додатку, який реалізуватиме можливість конструювання простих мебельних конструкцій для виробничого об'єднання «ВіЯр».

Проект являється *актуальним* як для компанії так і для суспільства, так як замовити готові матеріали для збірки якої-небудь меблів набагато дешевше ніж готову збірку. Цей проект допоможе звичайним людям, вигідно купувати якісну мебель без великих зусиль, так як даний проект направлений на середньостатистичну людину, не маючи особливих пізнань у тонкостях мебельного ремесла. В свою чергу компанія розширить кількість клієнтів та підвищить свою конкурентоспроможність на ринку.

Отже, *об'єкт дослідження* – процес проектування меблів.

Предмет дослідження – автоматизація процесу проектування меблів.

Задачі дослідження:

- аналіз актуальності та перспектив розробки проекту;
- розробка технічного завдання;
- створення гнучкого web-додатку для конструювання простих конструкцій.

					<i>КНТЕУ 6.050103 10-09.БР</i>			
					<i>Автоматизація процесу проектування меблів для виробничого підприємства «ВіЯр» м. Київ</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>В</i>	<i>3</i>	<i>26</i>
.Зав. каф.	Криворучко О.В.							
.Керівник	Рзасва С. Л.							
Гарант	Цензура М. О.							
Розроб	Нескород А. О.				<i>Вступ</i>	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		

- реалізація інформаційної системи проектування та представлення меблів у віртуальній середі;
- впровадження розробленої інформаційної системи у діяльність компанії «ВіАр».

					КНТЕУ 6.050103 10-09.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		4

РОЗДІЛ 1

ХАРАКТЕРИСТИКА КОМПАНІЇ ТМ «ВІЯР»

1.1. Опис структури компанії ТМ «ВіЯр»

ТМ «ВіЯр» - все для меблевого виробництва. Компанія «ВіЯр» була створена в 2002 році. Успішно працює вже більше 17 років на ринку меблів і меблевих комплектуючих.

Клієнти компанії – це спеціалізовані торгові компанії, різні виробничі підприємства, а також приватні майстри, що виробляють меблі. Саме для цього кола споживачів підприємство успішно постачає і виводить на ринок найсучасніші матеріали і якісні комплектуючі для меблів.

Торгова марка «ВіЯр» – це зразок високої якості продукції, різноманітності нових ідей і постійної уваги до своїх клієнтів. ТМ пропонує як індивідуальні, так і комплексні рішення. Комплектуючі для виробництва меблів підібрані з урахуванням багатьох факторів і вимог, вони покликані підвищити функціональність, зручність, цінність меблів для кінцевого споживача.

Замовлення на комплектуючі можна створити у виставочному центрі за допомогою менеджера, або за допомогою безкоштовного конструктору замовлень на сайті компанії, все що необхідно зробити для початку роботи – зареєструватися.

За допомогою онлайн додатку можна створити замовлення на наступні послуги:

- порізка та додаткова обробка ДСП, ДВП, МДФ: крайкування

					КНТЕУ 6.050103 10-09.БР			
Зм.	Аркуш	№ докум.	Підпис	Дата	Автоматизація процесу проектування меблів для виробничого підприємства «ВіЯр» м. Київ	Стадія	Аркуш	Аркушів
.Зав. каф.		Криворучко О.В.				Р1	5	26
.Керівник		Рзасва С. Л.						
Гарант		Цензура М. О.						
Розроб		Нескород А. О.						
					Характеристика компанії ТМ «ВіЯр»	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		

(EVA-клеєм, PUR-клеєм (водостійким) або Лазерне (безшовне)), зріз кутів, свердління, нанесення пазів фрезою або пилою, вибірка чверті, кутові вирізи «радіус», «кут» або «заріз», вирізи по спеціальним шаблонам під ручки, «Посмішка», Дуга, тощо, підрізання у розмір, зрощення деталей;

- порізка та додаткова обробка стільниці: крайкування (EVA-клеєм, PUR-клеєм (водостійким) або Лазерне (безшовне)), зріз заводської крайки та закруглення, зріз під кутом, кутові вирізи «радіус», «кут» або «заріз», спеціальні шаблони, в тому числі «Замок»;

- порізка крашених фасадів МДФ та обклеювання плівкових фасадів МДФ з додатковими обробками: внутрішнє фрезування, обробка торців, створення радіусного(гнутого) фасаду, виріз під вітрину, отвори під петлі;

- зборка розсувних, розпашних та складних систем;

- обробка скла та дзеркала.

Майже всі перераховані послуги виконуються на ЧПУ (автоматизованому станку) програми для яких формуються автоматично конструктором замовлень, інші специфічні обробки виконуються на ручному станку Altendorf.

Також можливо створити онлайн специфічне замовлення, яке не підтримує конструктор. Для цього треба оформити порізку та можливі операції, і при оформленні замовлення прикласти креслення. Такі замовлення менеджери оброблюють окремо та зв'язуються з замовником для уточнення вартості, термінів, тощо.

1.2. Розробка технічного завдання

Технічне завдання на створення інформаційної системи управління проектами “ ВіярПро типові конструкції” на основі розробки і впровадження апаратних та програмних комплексів, а також розробки та впровадження

					КНТЕУ 6.050103 10-09.БР	Аркуш
						6
Зм.	Аркуш	№ докум	Підпис	Дата		

бізнес-процесів проектного управління та взаємодії зацікавлених сторін виробництва меблевих комплектуючих та деталей, які фінансуються за кошти підприємств ТМ «ВіЯр».

Повне найменування системи:

Інформаційна система управління проектами “ВіЯрПро типові конструкції”.

Скорочене найменування системи:

ІСУП “ВПТК”.

Головний бенефіціар та потенційні користувачі системи:

Головним бенефіціаром та прямими користувачами ІСУП ВПТК є ТМ “ВіЯр” та клієнти компанії.

Призначення системи:

Проект має на меті спростити створення проектів типових конструкцій та їх замовлення через існуючу облікову систему компанії. Для цього система має бути спроможна виконувати такі задачі:

- спрощення введення даних;
- забезпечення захищеного протоколу автентифікації для входу в систему;
- запобігання помилкам з боку користувача під час введення даних;
- виконання перехресної перевірки інформації з використанням бази даних облікової системи компанії та бази даних системи;
- створення API для забезпечення доступу до даних та їх завантаження в різних форматах.

Мета створення системи:

ІСУП ВПТК створюється с метою:

- забезпечення збору і первинної обробки вихідної інформації, необхідної для підготовки проекту до замовлення;

					КНТЕУ 6.050103 10-09.БР	Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата		

- створення системи для створення бази даних типових конструкцій включаючи в себе дані про її складові частини;
- підвищення якості (точності зборки після виконання замовлення, уникнення проблем з підбором фурнітури) результату;
- скорочення часу на створення замовлення;
- зменшення навантаження на відділ технічної підтримки.

В результаті створення ІСУП ВПТК повинні бути поліпшені значення наступних показників:

- зменшення порогу для входження створення замовлення власноруч (без допомоги менеджера);
- зменшення кількості рекламаций виробничих замовлень;
- зменшення часу, що витрачається на створення та обслуговування замовлення;
- збільшення продаж фурнітури.

Вимоги до структури та функціонування системи, перелік підсистем:

Система ІСУП ВПТК повинна бути централізованою, тобто всі дані повинні розташовуватися в центральному сховищі.

В системі необхідно виділити наступні функціональні підсистеми:

- підсистема збору, обробки та завантаження даних, яка призначена для реалізації процесів збору даних з облікової системи компанії, приведення зазначених даних до виду, необхідного для наповнення підсистеми зберігання даних;
- підсистема зберігання даних, яка призначена для зберігання даних в структурах та роздачу даних через API;
- підсистема формування проектів типових конструкцій, включаючи в себе фільтри та правила формування конструкції, конвертація введеної інформації в зручний вигляд та передача в підсистему зберігання через API;

					КНТЕУ 6.050103 10-09.БР	Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

- підсистема створення спільного проекту, включаючого програму для ЧПУ та прив'язаний рахунок у обліковій системі.

Вимоги до способів і засобів інформаційного обміну між компонентами системи.

Як протокол взаємодії між компонентами Системи на транспортно-мережевому рівні необхідно використовувати протокол HTTP/HTTPS. Та для доступу до облікової системи протокол SOAP.

Вимоги до режимів управління системою

В залежності від типу користувача система ІСУП ВПТК повинна працювати в трьох режимах: «Адміністрування БД», «Проектування конструкції» та «Створення замовлення».

Режим «Адміністрування БД» використовується тільки Замовником. В цьому режимі можна налаштовувати імпорту даних, редагувати даних та переходити в режим «Проектування конструкції».

Режим «Проектування конструкції» використовується тільки Замовником. В цьому режимі можна створювати або редагувати вже існуючі конструкції в БД. Режим представляє собою повноцінний інструмент для створення конструкції із набору деталей та фурнітури, включає в себе трьох вимірне графічне представлення результату введених даних.

Режим «Створення замовлення» спілкується с БД через API і представляє клієнту список доступних конструкцій. Клієнт може вибрати, відредагувати та відправити в систему на обробку та оформлення замовлення.

1.3 Висновки до розділу 1

ТМ «ВіЯр» для покращення сервісу і розширення ринку потребує нове програмне забезпечення, яке автоматизує процес представлення готових виробів клієнтам компанії та допоможе створювати замовлення в «один клік»

					КНТЕУ 6.050103 10-09.БР	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

без необхідності додаткового зв'язку між клієнтом та співробітником компанії.

В результаті аналізу предметної області, безпосередньо компанії та споживчого ринку було створено технічне завдання. Технічне завдання описує всі вимоги до нової системи, які забезпечать виконання поставлених цілей та задач.

					КНТЕУ 6.050103 10-09.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

РОЗДІЛ 2

ХАРАКТЕРИСТИКА ПРОГРАМНО-ТЕХНІЧНОЇ БАЗИ

2.1. Аналіз технічної бази компанії «ВіЯр»

Створювана система призначена для окремого підприємства, тому при проектуванні і виборі технології треба відштовхуватися від того, що на даний момент компанія використовує у своїх робочих проектах.

Підприємство має декілька windows-серверів, але вони виділені спеціально для віддаленої роботи працівників або функціонування облікової системи, тому доступ до них закрит.

Також є сім лінукс серверів виокремлених спеціально для web-додатків:

- 1) сервер для сайту міста Київ;
- 2) сервер для сайту міста Харків;
- 3) сервер для сайту міста Дніпро;
- 4) сервер для сайту міста Мінськ (Білорусія);
- 5) сервер для конструктора замовлень ViyarPro;
- 6) сервер для корпоративного порталу;
- 7) сервер для тестування розробок.

Крім того, у ТЗ замовник зазначив, що виділить окремі сервери для нової системи з аналогічною конфігурацією, а саме два сервери з ОС CentOS 7 і предустановленими сервісами php 7, apache, mysql. Також було дозволено використовувати nodejs, як компоновщик проекту.

					<i>КНТЕУ 6.050103 10-09.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Автоматизація процесу проектування меблів для виробничого підприємства «ВіЯр» м. Київ</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
.Зав. каф.		Криворучко О.В.				P2	11	26
.Керівник		Рзасва С. Л.						
Гарант		Цензура М. О.						
Розроб		Нескород А. О.			<i>Характеристика програмно-технічної бази</i>	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		

Отже, для системи обрана наступна структура:

- один виділений сервер для проекту буде працювати у вигляді API для доступу до БД. На ньому буде працювати MS MySQL, в котрій буде зберігатися необхідна інформація для проекту. Сама API написана мовою PHP і слугуватиме зв'язком між БД та другим сервером;
- другий виділений сервер буде працювати безпосередньо за клієнтськими запитами. Через нього клієнт буде отримувати клієнтську сторінку для створення замовлення та безпосередньо відправляти заяву на створення замовлення в обліковій системі.

2.2. Розробка БД для компанії ТМ «ВіЯр»

База даних необхідна для задання основної конфігурації до можливих конструкцій, а також зберігання списку користувачів та їх замовлень. Для реалізації фізичної структури бази даних обрана вільна реляційна система управління базами даних MySQL версії 5,6.

MySQL – компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

					КНТЕУ 6.050103 10-09.БР	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

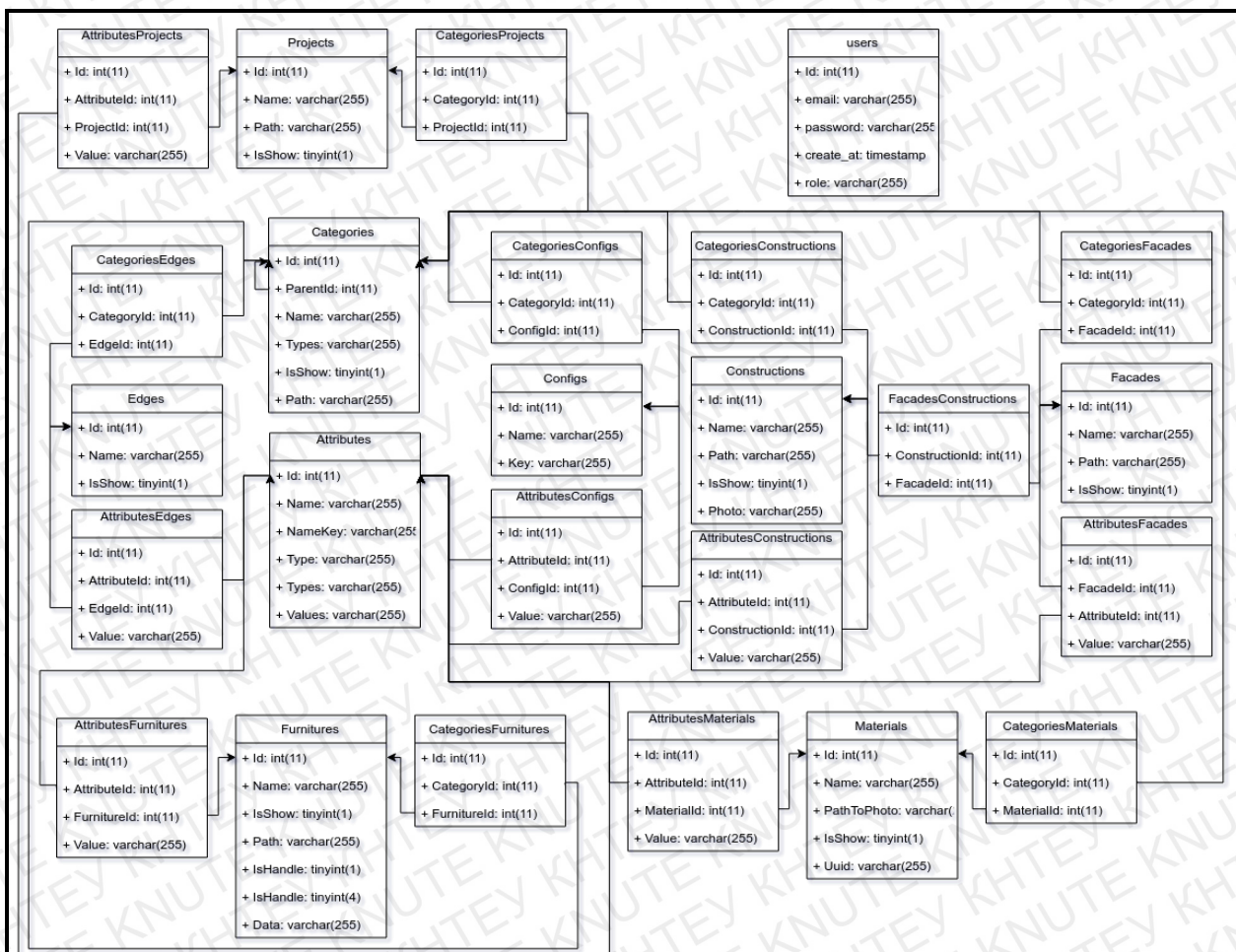


Рис. 2.1. Фізична модель бази даних

2.3. Висновки до розділу 2

Глобально, будь-яка програма складається з трьох частин: вхідні дані, логіка, вихідні дані. Саме за коректність поданих вхідних даних в програму відповідає база даних. При правильній її побудові відпадає необхідність багатьох перевірок на цілісність вхідних даних на етапі виконання логіки, тому що правильно спроектована БД за всіма принципами побудови БД сама по собі не дає ввести у структуру некоректні дані. Саме тому цей етап розробки системи є дуже важливим.

Умовою успішного створення БД, була можливість із графічного інтерфейсу додавати поля до існуючих об'єктів. Дана умова успішно виконана методом атрибутів. Усі поля зберігаються в окремій таблиці і по

					Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 6.050103 10-09.БР
					13

ключам зв'язуються безпосередньо з самим об'єктом, таким чином можна кожному об'єкту створити нове поле і при цьому не треба створювати нову таблицю.

					<i>КНТЕУ 6.050103 10-09.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ ПРОЦЕСУ
ПРОЕКТУВАННЯ МЕБЛІВ ДЛЯ ВИРОБНИЧОГО
ПІДПРИЄМСТВА «ВІЯР»

3.1 Визначення цілі проекту

ІСУП ВПТК призначена для підвищення швидкості та зручності зручності створення типових замовлень клієнтами.

Основним призначенням ІСУП ВПТК є автоматизація створення типових проектів та представлення клієнтам компанії можливості редагувати основні параметри проекту і подальшого оформлення замовлення.

В рамках проекту автоматизуються процес оформлення замовлення на підприємстві:

- можливість створення єдиного проекту для декількох конструкцій;
- можливість швидкого редагування базових параметрів проекту;
- включення в рахунок фурнітури;
- автоматична розстановка додаткових обробок для фурнітури;
- виключення помилок клієнтами при проектуванні меблів у специфічному сервісі компанії.

Система повинна підтримувати такі режими функціонування:

- основний режим, в якому підсистеми ІСУП ВПТК виконують всі свої основні функції;
- профілактичний режим, в якому одна або всі підсистеми ІСУП ВПТК не виконують своїх функцій.

						<i>КНТЕУ 6.050103 10-09.БР</i>			
						<i>Автоматизація процесу проектування меблів для виробничого підприємства «ВіЯр» м. Київ</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>					
Зав. каф.		Криворучко О.В.			РЗ		15	26	
Керівник		Рзасва С. Л.							
Гарант		Цензура М. О.							
Розроб		Нескород А. О.				Розробка програмного модулю процесу проектування меблів для виробничого підприємства «ВіЯр»	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		

В основному режим функціонування Система ІСУП ВПТК повинна забезпечувати:

- роботу користувачів, клієнтське програмне забезпечення і технічні засоби забезпечують можливість функціонування в режимі 24 години на день, 7 днів на тиждень (24x7);
- серверне програмне забезпечення та технічні засоби серверів забезпечують можливість цілодобового функціонування за перервами на обслуговування;
- справне функціонування системного, базового і прикладного програмного забезпечення системи;
- виконання своїх функцій – збір, обробка та завантаження даних; зберігання даних; передача даних через API.

Для забезпечення основного режиму функціонування системи необхідно виконувати вимоги і витримувати умови експлуатації програмного забезпечення і комплексу технічних засобів системи, зазначені у відповідних документах (технічна документація, інструкції з експлуатації і т.д.).

У профілактичному режимі Система ІСУП ВПТК повинна забезпечувати можливість проведення наступних робіт:

- технічне обслуговування;
- модернізацію апаратно-програмного комплексу;
- усунення аварійних ситуацій (аварійний режим функціонування).

Загальний час проведення профілактичних робіт не повинен перевищувати X% від загального часу роботи системи в основному режимі (Y годин в місяць).

Вказуються вимоги по діагностуванню системи (які кошти будуть використовуватися або створюватися, щоб забезпечити діагностику системи).

					КНТЕУ 6.050103 10-09.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

Для забезпечення високої надійності функціонування Системи як системи в цілому, так і її окремих компонентів повинно забезпечуватися виконання вимог з діагностування її стану.

В залежності від типу користувача система ІСУП ВПТК повинна працювати в трьох режимах: «Адміністрування БД», «Проектування конструкції» та «Створення замовлення».

Режим «Адміністрування БД» використовується тільки Замовником. В цьому режимі можна налаштовувати імпорт даних, редагувати даних та переходити в режим «Проектування конструкції».

Режим «Проектування конструкції» використовується тільки Замовником. В цьому режимі можна створювати або редагувати вже існуючі конструкції в БД. Режим представляє собою повноцінний інструмент для створення конструкції із набору деталей та фурнітури, включає в себе трьох вимірне графічне представлення результату введених даних.

Режим «Створення замовлення» спілкується с БД через API і представляє клієнту список доступних конструкцій. Клієнт може вибрати, відредагувати та відправити в систему на обробку та оформлення замовлення.

3.2. Розробка API доступу до БД

Сервер бази даних необхідно розробити повністю як окремий сервіс. В такому випадку при розширенні проекту не виникне проблем з сумісністю підсистем.

Для зберігання даних була обрана СУБД MySQL. Для управління СУБД необхідно розробити базову структуру та інтерфейс для наповнення бази даних даними.

Для розробки API обрано використовувати Slim Framework. Slim - це мікроструктура PHP, яка допомагає швидко писати прості, але потужні веб-

					КНТЕУ 6.050103 10-09.БР	Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата		

додатки і API. По суті, Slim - диспетчер, який отримує HTTP-запит, викликає відповідну процедуру зворотного виклику і повертає HTTP-відповідь.

Додаток Slim містить маршрути, що відповідають певним HTTP-запитами. Кожен маршрут викликає зворотний виклик і повертає HTTP-відповідь. Щоб почати роботу, спочатку створюється екземпляр і налаштовується додаток Slim.. Нарешті, запускається додаток Slim.

Графічний інтерфейс побудован за допомогою Vue.js. Фреймворк надає формат шаблонізації «.vue» для створення реактивного GUI. Шалон складається з трьох частин: template – html-каркас, script – js логіка, style – css правила.

Графічний інтерфейс надає можливість створювати нові об'єкти за заданими моделями даних, а також модифікувати існуючі моделі (додавати нові атрибути).

Рис 3.1 Графічний інтерфейс адміністрування БД

3.3 Побудова графічного інтерфейсу клієнтського режиму

3.3.1. Опис технології розробки графічного інтерфейсу

Актуальним підходом до побудови WEB GUI є використання фреймворків для реактивного рендерингу html-сторінок. Vue.js один з найкращих подібних фреймворків для реалізації реактивних html-сторінок.

Фреймворк підтримує три варіанти створення сторінок:

- рендер з використанням шаблонів;
- рендер з використанням рендер-функції та JSX;
- серверний рендер.

Для даного проекту доцільніше всього використовувати рендер за шаблонами. Він найпростіший і найзручніший. При використанні разом з webpack та vue-loader шаблони конвертуються в рендер-функції, тому по продуктивності не уступає другому варіанту. Серверний рендеренг дуже важки та дорогий метод, він більше підходить для особливих систем, які потребують більшої швидкості та надійності, наприклад банківські системи.

Шаблон vue.js представляється компонентою. В загальному випадку це файл з розширенням «.vue». Вміст такого файлу представляється трьома тегами: `template` – html-каркас компоненти зі вставками, які будуть підставлятися із скрипту під час рендеренгу; `script` – JavaScript або TypeScript логіка компоненти, в цьому тезі створюється об'єкт, який буде відповідати за те, які дані зберігає компонента, які дані отримує та які функції виконую; `style` – css/sass правила відображення компоненти, ці правила можуть бути закріпленні до конкретної компоненти, або мати вплив на всю html-розмітку.

Webpack – інструмент працюючий на основі node.js, призначений для збирання проекту в готовий дистрибутив. Крім того даний інструмент має достатньо утиліт для зручного зневадження та тестування додатку. Для webpack існує спеціальний плагін vue-loader, за допомогою якого цей інструмент отримує можливість конвертації компонентів vue.

					КНТЕУ 6.050103 10-09.БР	Аркуш
						19
Зм.	Аркуш	№ докум	Підпис	Дата		

3.3.2. Реалізації графічного інтерфейсу

Налаштування webpack:

Перед усім для роботи з проектом необхідно налаштувати збірку проекту. Webpack для специфічних збірок використовує додаткові модулі, які встановлюються за допомогою менеджера пакетів NPM і зберігаються в папці node_modules, а сам перелік модулів зберігається списком в файлі package.json. Для проекту необхідні такі модулі:

- @babel/core@^7.4.4, @babel/preset-env@^7.4.4 – окремі частини модулю babel, призначені для роботи з новими стандартами ECMAScript, а саме парсингу та конвертації до зрозумілого більшості браузерів формату;
- babel-loader@^8.0.6 – модуль спеціально розроблений для webpack, він допомагає webpack загрузити та обробляти JS-файли;
- css-loader@^2.1.1 – модуль призначений для загрузки стилів css;
- file-loader@^3.0.1 – модуль призначений для загрузки простих файлів;
- html-webpack-plugin@^3.2.0 – модуль являється плагіном для webpack.

З його допомогою webpack може створювати html-сторінки;

- vue-loader@^15.7.0, vue-style-loader@^4.1.2, vue-template-compiler@^2.6.10 – модулі призначені для загрузки та обробки компонент Vue.js (в загальному випадку файли за розширенням «.vue»);

- webpack@^4.31.0, webpack-cli@^3.3.2 – модулі самого webpack 4;
- webpack-dev-server@^3.3.1 – модуль призначений для розробки додатків для webpack. Представляє собою локальний сервер, який здатен слідкувати за змінами в файлах і автоматично їх заново обробляти замість того щоб збирати все спочатку.

Для завершення налаштування webpack залишається тільки створити файл webpack.config.js і прописати в ньому всі правила, реалізацію конфігурації прикладено до додатку 1.

Файлова структура проекту:

					КНТЕУ 6.050103 10-09.БР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

Вихідний код розбито за типом файлів, а також режиму в якому використання. Причина того, що три режими із чотирьох лежать в одній структурі в том, що більшість моделей в них спільні і відрізняються лиш налаштуваннями.

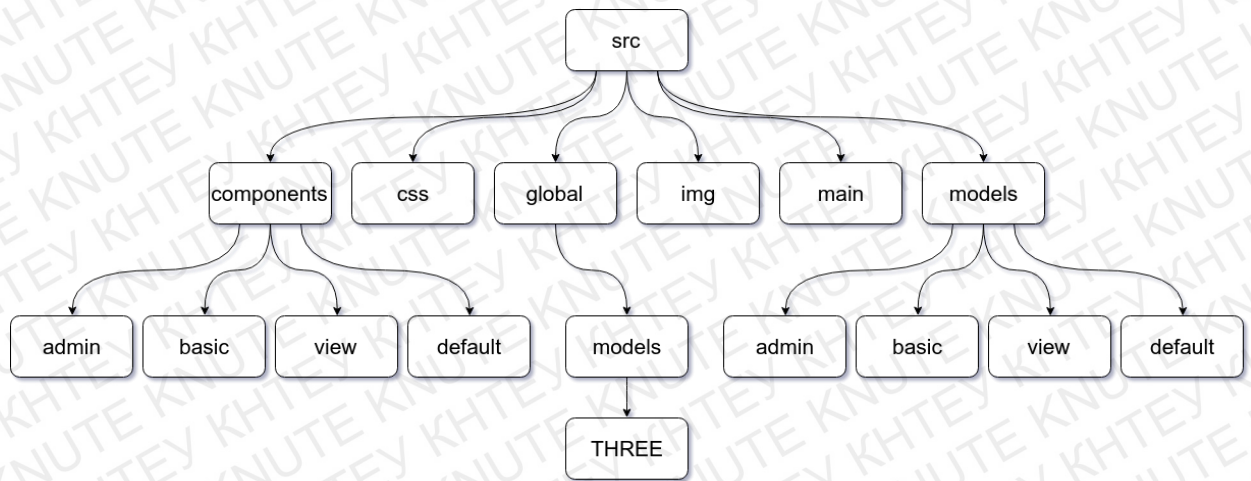


Рис. 3.2. Файлова структура

- *components* – директорія з файлами компонентами «.vue». Директорія розбита на окремо кожний режим та default для спільних компонент;
- *css* – директорія з файлами стилів CSS;
- *global* – директорія з файлами, в яких прописані глобальні константи та дані. В ній знаходиться папка THREE, в якій знаходяться всі кас томні модулі для THREE та файл збірки пакету THREE.js із конкретно необхідних модулів для проекту. THREE.js підключається глобально;
- *img* – директорія з графічними зображеннями;
- *main* – директорія з файлами запуску конкретного режиму додатку;
- *models* – директорія в якій зберігаються моделі даних та контролери. Директорія розбита на окремо кожний режим та default для спільних моделей. Більшість моделей мають основний опис в файлі за директорією default і розширюються методом наслідування в окремих директоріях для конкретного режиму.

Основний підхід до реалізації:

Всі моделі даних реалізуються з передбаченням можливості їх використання з попереднім розширенням в будь-якому із режимів, використовуючи наслідування.

Компоненти виділені в окрему директорію і моделі ні як з ними не взаємодіють. Вся необхідність спілкуватись між GUI та моделями даних передбачена глобальним зберіганням моделі проекту та реактивністю Vue.js.

3.4. Розробка логіки тривимірної сцени

3.4.1. Опис технології розробки

WebGL — це стандарт на базі OpenGL ES 2.0, що дозволяє розробникам веб-контенту вбудовувати в веб-оглядачі, які підтримують HTML5, повноцінну 3D-графіку, не вдаючись до посередництва плагінів. В намірах розробників поширити стандарт WebGL не тільки в браузерах персональних комп'ютерів, а й у мобільних інтернет-пристроях. До робочої групи WebGL входять Khronos Group, представники провідних розробників інтернет-браузерів, таких як Apple Safari, Google Chrome, Mozilla Firefox і Opera, а також фахівці AMD і Nvidia.

Ця технологія дозволяє упроваджувати апаратно-прискорену 3D графіку у веб-сторінки без необхідності використовувати спеціальні плагіни веб-браузера на будь-якій платформі, що підтримує OpenGL або OpenGL ES. Технічно це буде прив'язкою скриптів JavaScript до функцій, визначених в бібліотеках OpenGL ES 2.0, реалізовану на рівні браузера.

WebGL є подальшим розвитком експерименту Canvas 3D в Mozilla і вже представлена у збірках розробників Mozilla Firefox і WebKit, а також в попередніх релізах Google Chrome 4.

THREE.js - легка кросбраузерна бібліотека JavaScript, що використовується для створення та відображення анімованої комп'ютерної

					<i>КНТЕУ 6.050103 10-09.БР</i>	Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата		

3D графіки при розробці веб-додатків. Three.js скрипти можуть використовуватися спільно з елементом HTML5 CANVAS, SVG або WebGL. Вихідний код розташований в репозиторії GitHub.

Особливості:

- тип рендеру: Canvas, SVG або WebGL;
- додавання і видалення об'єктів в режимі реального часу;
- доступні перспективна та ортографічна камери;
- каркаси, пряма кінематика, інверсна кінематика, покадрова анімація;
- джерела світла: зовнішній, спрямований, точковий; тіні: кинуті і отримані;
- повний доступ до всіх OpenGL-шейдерів (GLSL);
- багата кількість заготовлених об'єктів, серед них частинки, спрайт, лінії, скелетна анімація, тощо;
- різноманітна геометрія: площину, куб, сфера, тор, 3D текст, тощо;
- модифікатори: тканина, видавлювання;
- великий набір плагінів для імпорту та експорту;
- наявність повної документації;
- наявність прикладів (більше 150 на офіційному сайті).

3.4.2. Реалізація тривимірної сцени

Тривимірна сцена будується засобами бібліотек THREE.js, яка підключається глобально. В проект включаються тільки необхідні модулі, реалізація збору модуля надано в додатку 3.

Для створення безпосередньо сцени був розроблений спеціальний модуль DefaultScene. В ньому прописана логіка по побудові сцени та ініціалізації рендеренгу, а також додаткові метод для динамічної модифікації сцени (лістинг модулю наданий в додатку 4). Модель даного модулю

					КНТЕУ 6.050103 10-09.БР	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

створюється в компоненті Scene.vue і включається в глобальну область для можливості із будь-якої точки додатку працювати з графікою.

Для кімнати та деталей конструкцій розроблені окремі модулі, які являються розширеними класами до класу Mesh (Mesh – повноцінний 3д об'єкт з геометрією, текстурою та додатковими атрибутами) або Group (Group – 3д об'єкт, який не має явної форми і ні як не відображається, але має математичну позицію і може включати та модифікувати дочірні 3д об'єкти).

3.5. Висновок до розділу 3

Проект досить об'ємний, в ньому багато логіки реалізації, а також зовнішніх модулів реалізованих іншими розробниками, але завдяки webpack, наявна можливість зберігати все в будь-якій структурі і в різних файлах. Також завдяки файловому менеджеру NPM усі зовнішні модулі підключаються автоматично.

Інструменти для розробки спрощують зневадження та пошук помилок, а також пришвидшують швидкість написання коду, завдяки livereload та livebuilding.

Реалізацією апробації проекту стало впровадження в компанію ТМ «ВіЯр», що засвідчує акт впровадження.

					КНТЕУ 6.050103 10-09.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		24

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В результаті було розроблену складну систему із чотирьох режимів, які призначення для створення виробів, адміністрування та продажу виробів клієнтам. Також були розроблені додаткові інтерфейси обміну даних з існуючими системами компанії.

Для реалізації системи проаналізовано компанію та ринок, створено технічне завдання та виділено список завдань та цілей. Всі поставлені задачі успішно виконані та реалізовані.

Результатом реалізації випускного кваліфікаційного проекту стало впровадження системи, що засвідчує акт впровадження.

Надалі проект можна розширити створенням режиму для замовлення не окремих виробів, а цілих груп. Наприклад, замовлення кухонь, на сцені представляти кімнати, в яку можна ставити вироби по визначеним правилам і в результаті підраховувати розкрій та фурнітуру.

Також проект можна розширити створенням зв'язків з іншими проектами. Наприклад, компанія планує запуск порталу «ВіЯр Базар», на якому клієнти зможуть знаходити собі меблярів і домовлятися про збірку виробу або групи виробів. Можна приєднати до цього новий проект, щоб клієнт міг самостійно вибрати, який виріб і в якому кольорі йому потрібен, а мебляр по результатах просто його збере.

					КНТЕУ 6.050103 10-09.БР			
Зм.	Аркуш	№ докум.	Підпис	Дата	Автоматизація процесу проектування меблів для виробничого підприємства «ВіЯр» м. Київ	Стадія	Аркуш	Аркушів
.Зав. каф.		Криворучко О.В.				ВП	25	26
.Керівник		Рзасва С. Л.						
Гарант		Цензура М. О.						
Розроб		Нескород А. О.			Висновки та пропозиції	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Задорожна Н. Т. Менеджмент документообігу в інформаційних системах освіти (для ВНЗ і ППО): Навч.-метод. посіб. // Н.Т. Задорожна, К.М. Лавріщева – К.: КП Видавництво «Педагогічна думка», 2007. – 227 с.
2. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах»: від 05.07.1994 №80/94-ВР // [Електронний ресурс] – Режим доступу : <http://zakon2.rada.gov.ua/laws/show/80/94-вр>
3. Покришень Д. А. Науково-методичні та технологічні вимоги до інформаційних систем галузі освіти / Інформаційні технології і засоби навчання. 2012. №4 (30) [Електронний ресурс]. – Режим доступу : <http://www.journal.iitta.gov.ua>.
4. Покришень Д.А. Методико-технологічні вимоги до інформаційного ресурсу закладу післядипломної педагогічної освіти / Д.А. Покришень // Інформаційні технології і засоби навчання: електронне наукове фахове видання [Електронний ресурс] / Гол. ред.: В.Ю. Биков; Ін-т інформ. технологій і засобів навчання АПН України, Ун-т менеджменту освіти АПН України. – 2014. – № 1(39). – Режим доступу <http://www.journal.iitta.gov.ua> – Заголовок з екрана.
5. Покришень Д.А. Технологічні принципи проектування інформаційних систем / Д.А. Покришень // Сборник научных трудов SWorld. – Выпуск 1. Том 17. – Иваново: МАРКОВА АД, 2014. – С. 37-44
6. Что такое UX и UI дизайн – особенности и отличия [Електронний ресурс]. – Режим доступу : <http://www.kasper.by/help/chto-takoe-ux-i-ui-dizain/>

					<i>КНТЕУ 6.050103 10-09.БР</i>				
					<i>Автоматизація процесу проектування меблів для виробничого підприємства «ВіАр» м. Київ</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		СД	26	26	
.Зав. каф.		Криворучко О.В.				<i>Список використаних джерел</i>	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
.Керівник		Рзасва С. Л.							
Гарант		Цензура М. О.							
Розроб		Нескород А. О.							

ДОДАТКИ

Додаток 1: Лістинг конфігурація для webpack 4

```
const path = require('path')
const webpack = require('webpack')
const HtmlWebpackPlugin = require('html-webpack-plugin');
const VueLoaderPlugin = require('vue-loader/lib/plugin')

const isServ = require('os').hostname() === 'viyar.pro'
const outputPath = isServ ? path.resolve('./ext_www/kitchens.viyar.pro/') : path.join(__dirname,
'../dist')

const config = env => {
  const res = {
    mode: env,
    // Entry viyar-modules/src/
    entry: path.join(__dirname, 'main/basic.js'),

    // Output
    output: {
      path: outputPath,
      publicPath: './',
      filename: 'js/view.js'
    },

    // Loaders
    module: {
      rules: require('./rules')
    },

    // Plugins
    plugins: [
      new VueLoaderPlugin(),
      new HtmlWebpackPlugin({
```



```
    hash: true,
    filename: 'index.html',
    template: './viyar-modules/template/template.html',
    inject: 'body'
  }),
  // Define global
  new webpack.DefinePlugin({
    offline: env !== 'production',
    MODE: "'basic'"
  }),
  // Require global
  new webpack.ProvidePlugin(
    require('./global/basic')
  )
],

devServer: {
  publicPath: '/'
}
}

if(env === 'production') {
  res.optimization = { minimize: true }
} else {
  // Reload On File Change
  res.watch = true
  // Development Tools (Map Errors To Source File)
  res.devtool = 'source-map'
}

return res
}

module.exports = config
```

Додаток 2: Лістинг правил для webpack 4

```
module.exports = [  
  // JS  
  {  
    test: /\.js$/,  
    exclude: file => /node_modules/.test(file) && !file.includes('/node_modules/three/'),  
    use: ['babel-loader']  
  },  
  // CSS  
  {  
    test: /\.css$/,  
    use: [  
      'vue-style-loader',  
      'css-loader'  
    ]  
  },  
  // VUE  
  {  
    test: /\.vue$/,  
    loader: 'vue-loader'  
  },  
  // FILES  
  {  
    test: /\.(png|jpg|gif|svg|obj)$/,  
    loader: 'file-loader',  
    options: {  
      name: '[name].[ext]?[hash]'  
    }  
  }  
]
```

Додаток 3: Лістинг збору модулів THREE.js

```
const THREE = {}  
  
// core  
THREE.EventDispatcher = require('three/src/core/EventDispatcher').EventDispatcher  
THREE.Object3D = require('three/src/core/Object3D').Object3D  
THREE.Geometry = require('three/src/core/Geometry').Geometry  
THREE.BufferGeometry = require('three/src/core/BufferGeometry').BufferGeometry  
THREE.Face3 = require('three/src/core/Face3').Face3  
THREE.Raycaster = require('three/src/core/Raycaster').Raycaster  
THREE.Shape = require('three/src/extras/core/Shape').Shape  
THREE.Path = require('three/src/extras/core/Path').Path  
  
// render  
THREE.WebGLRenderer = require('three/src/renderers/WebGLRenderer').WebGLRenderer  
  
// scenes  
THREE.Scene = require('three/src/scenes/Scene').Scene  
  
// cameras  
THREE.PerspectiveCamera =  
require('three/src/cameras/PerspectiveCamera').PerspectiveCamera  
THREE.OrthographicCamera =  
require('three/src/cameras/OrthographicCamera').OrthographicCamera  
  
// lights  
THREE.AmbientLight = require('three/src/lights/AmbientLight').AmbientLight  
THREE.PointLight = require('three/src/lights/PointLight').PointLight  
  
// helpers  
THREE.AxesHelper = require('three/src/helpers/AxesHelper').AxesHelper  
THREE.Box3Helper = require('three/src/helpers/Box3Helper').Box3Helper  
  
// loaders  
THREE.TextureLoader = require('three/src/loaders/TextureLoader').TextureLoader  
THREE.FileLoader = require('three/src/loaders/FileLoader').FileLoader  
  
// objects  
THREE.Mesh = require('three/src/objects/Mesh').Mesh  
THREE.Group = require('three/src/objects/Group').Group  
THREE.Line = require('three/src/objects/Line').Line  
THREE.LineSegments = require('three/src/objects/LineSegments').LineSegments
```



```
THREE.Sprite = require('three/src/objects/Sprite').Sprite
// math
THREE.Vector2 = require('three/src/math/Vector2').Vector2
THREE.Vector3 = require('three/src/math/Vector3').Vector3
THREE.Quaternion = require('three/src/math/Quaternion').Quaternion
THREE.Spherical = require('three/src/math/Spherical').Spherical
THREE.Box3 = require('three/src/math/Box3').Box3
THREE.Color = require('three/src/math/Color').Color
THREE.Plane = require('three/src/math/Plane').Plane
// textures
THREE.CanvasTexture = require('three/src/textures/CanvasTexture').CanvasTexture
// exclude
THREE.Stats = require('stats-js')
// other
const geometries = require('three/src/geometries/Geometries')
const materials = require('three/src/materials/Materials.js')
const constants = require('three/src/constants')
const buffattrs = require('three/src/core/BufferAttribute')

for(let geom in geometries) THREE[geom] = geometries[geom]
for(let mat in materials) THREE[mat] = materials[mat]
for(let con in constants) THREE[con] = constants[con]
for(let buf in buffattrs) THREE[buf] = buffattrs[buf]

// custom and other modules
require('./modules/list').forEach(name => THREE[name] =
require(`./modules/${name}`)[name])

// custom methods
require('./methods').append(THREE)

// export
// module.exports = THREE
export { THREE }
```

Додаток 4: Лістинг модулю побудови сцени THREE.js

```
const Stats = require('stats-js')
import { Scene } from 'three/src/scenes/Scene'
import { AmbientLight } from 'three/src/lights/AmbientLight'
import { PointLight } from 'three/src/lights/PointLight'
import { WebGLRenderer } from 'three/src/renderers/WebGLRenderer'
import { PerspectiveCamera } from 'three/src/cameras/PerspectiveCamera'
import { OrthographicCamera } from 'three/src/cameras/OrthographicCamera'
import { OrbitControls } from './OrbitControls'
import * as dat from 'dat.gui'

const cam_data = [
  [180, 0], // 0 спереди
  [-90, 0], // 1 слева
  [180, 90], // 2 сверху
  [90, 0], // 3 справа
  [0, -90], // 4 снизу
  [0, 0] // 5 сзади
]

function polarToVector3(lat, lng, radius) {

  lng = Math.max(-89.99, Math.min(89.99, lng))

  const theta = (lat - 90) * (Math.PI / 180)
  const phi = (90 - lng) * (Math.PI / 180)

  const x = radius * Math.sin(phi) * Math.cos(theta)
  const y = radius * Math.cos(phi)
  const z = radius * Math.sin(phi) * Math.sin(theta)

  return {x, y, z}
}

function DefaultScene() {
```

```
// init stats
this.stats = new Stats()
this.stats.setMode(0);
this.stats.domElement.style.position = 'absolute';
this.stats.domElement.style.top = 'auto';
this.stats.domElement.style.left = 'auto';
this.stats.domElement.style.right = '0px';
this.stats.domElement.style.bottom = '0px';
document.body.appendChild(this.stats.domElement)

// create scene
this.scene = new Scene()

// create lights
const pointLight = new PointLight(0xFFFFFF, 0.3, 0, 2)
const ambientLight = new AmbientLight(0xcccccc, 0.2)
pointLight.position.y = 1500

this.scene.add(pointLight)
this.scene.add(ambientLight)

// create renderer
this.renderer = new WebGLRenderer({ antialias: true, alpha: true })
this.renderer.setClearColor(0xEEEEEE, 0.0)
window.addEventListener('resize', this.updateRendererSize.bind(this), false)

// create camera
this.camera;

// create controls
this.controls;

// create dat.GUI
if(offline) {
  const gui = new dat.GUI()
```



```

gui.add(pointLight, 'intensity', 0, 2, 0.01)
gui.add(pointLight, 'distance', 0, 100000, 1000)
gui.add(pointLight, 'decay', 0, 2, 0.01)
gui.add(ambientLight, 'intensity', 0, 2, 0.01)
gui.hide()
}

// call render function
this.render()
}

DefaultScene.prototype.updateRendererSize = function() {
  if(!this.renderer.domElement.parentElement) {
    console.error('cant calc size without parent dom element')
    return
  }

  if(this.camera && this.renderer.domElement.parentElement) {
    const parentElement = this.renderer.domElement.parentElement
    const aspect = parentElement.clientWidth/parentElement.clientHeight
    const width = parentElement.clientWidth
    const height = parentElement.clientHeight
    const factor = 1.25
    this.renderer.setSize(width, height)
    this.camera.aspect = aspect
    if (this.camera.type === 'OrthographicCamera') {
      this.camera.left = -width / factor
      this.camera.right = width / factor
      this.camera.top = height / factor
      this.camera.bottom = -height / factor
    }
    this.camera.updateProjectionMatrix()
  }
}

```

```
DefaultScene.prototype.setCamera = function(type) {  
  if(!this.renderer.domElement.parentElement) {  
    console.error('cant calc size without parent dom element')  
    return  
  }  
  
  if(this.camera && this.camera.type.toLowerCase() === type.toLowerCase()) {  
    return;  
  }  
  
  if(this.camera && this.camera.parent) {  
    this.camera.parent.remove(this.camera)  
  }  
  
  const parentElement = this.renderer.domElement.parentElement  
  
  switch(type.toLowerCase()) {  
    case 'perspectivecamera':  
      this.camera = new PerspectiveCamera(75,  
parentElement.clientWidth/parentElement.clientHeight, 0.1, 1000000)  
      this.camera.position.z = 1000  
      this.camera.add(new PointLight(0xffffff, .5))  
      break;  
    case 'orthographiccamera':  
      const factor = 1.25  
      this.camera = new OrthographicCamera(  
-parentElement.clientWidth / factor,  
parentElement.clientWidth / factor,  
parentElement.clientHeight / factor,  
-parentElement.clientHeight / factor,  
-30000, 100000  
)  
      this.camera.position.z = 10  
      this.camera.add(new PointLight(0xffffff, .3))  
      break;  
  }
```

```

    default:
      console.error('undefined camera type')
    }

    this.scene.add(this.camera)

    if(this.controls) {
      this.controls.object = this.camera
      // this.controls.update()
    } else {
      this.controls = new OrbitControls(this.camera, parentElement)
    }
  }
}

```

```

DefaultScene.prototype.toggleView = function(id, r = 1000) {

```

```

  if (id < 1 && id > 6) {
    console.error('undefined view side')
    return
  }

```

```

  const data = cam_data[id - 1] // [h, v]

```

```

  const vec = polarToVector3(data[0], data[1], r)

```

```

  this.controls.reset()

```

```

  this.camera.position.copy(vec)

```

```

  if (this.camera.type === 'OrthographicCamera') {

```

```

    const w = Math.min(

```

```

      Math.abs(this.camera.left) + Math.abs(this.camera.right),

```

```

      Math.abs(this.camera.top) + Math.abs(this.camera.bottom)

```

```

    )

```

```

    this.camera.zoom = w / (r * 2)

```

```

    this.camera.updateProjectionMatrix()
  }

```

```

  this.controls.update()
}

```



```
DefaultScene.prototype.render = function() {  
  this.stats.update()  
  this.scene.traverse(child => {  
    if (typeof child.update == 'function') child.update()  
  })  
  requestAnimationFrame(this.render.bind(this))  
  
  if(this.camera) {  
    this.renderer.render(this.scene, this.camera)  
  }  
}  
  
export { DefaultScene }
```