

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

Кафедра цифрової економіки та бізнес-аналітики

КУРСОВА РОБОТА
з навчальної дисципліни „Проектування та адміністрування БД і
СД”

на тему:

Інформаційна система з курсів програмування

Галузь знань: 05 «Соціальні та поведінкові науки»

Спеціальність: 051 «Економіка»

Спеціалізація: «Інформаційні технології в бізнесі»

Освітній ступінь: бакалавр

Науковий керівник:
к.е.н., доц. каф Старух А. І.
(науковий ступінь, посада, прізвище, ініціали)
“ ___ ” травня 2022 р.
(підпис)

Виконавець:
студент(ка) групи УФЕ-31 с
Кузьма Б.О
“ ___ ” травня 2022 р.
(підпис)

Загальна кількість балів _____

(підпис, ППЧ членів комісії)

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ	5
1.1 Постановка завдання.....	5
1.2 Розробка моделі	6
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ.....	7
2.1 Опис моделі бази даних	7
2.2 Визначення типів даних.....	11
2.3 Обмеження цілісності даних	14
2.4 Реалізація SQL–скрипту	16
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ	19
3.1 Структура і макет веб-сайту	19
3.2 Програмування клієнтської частини	22
ВИСНОВКИ.....	25
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	26
ДОДАТКИ	27

ВСТУП

Актуальність теми дослідження. Сьогоднішній світ перейшов на новий етап життя, де головну роль виконує інформація, а також економіка, що будується на ній. Сучасний розвиток інформаційного суспільства безпосередньо пов'язаний з необхідністю збору, обробки і передачі величезних об'ємів інформації, перетворенням інформації у товар, як правило, значної вартості. Це стало причиною глобального переходу від індустріального суспільства до інформаційного. Поява всесвітньої мережі Інтернет спричинила масштабне зростання міжнародних спілкувань у різних сферах людського життя.

Інформація є одним з найцінніших ресурсів суспільства поруч з традиційними матеріальними видами ресурсів, як нафта, метал, корисні копалини тощо, тому, процес переробки інформації, подібно до процесів переробки матеріальних ресурсів можна сприймати як технологію. Інформаційна технологія передбачає вміння грамотно працювати з інформацією і обчислювальною технікою.

ІТ сфера не стоїть на місці. І те, що було затребуване вчора, сьогодні це вже практично ні кому не потрібне. З'являються складніші технології та завдання. Зокрема, мови програмування, мають бути найсвіжіших на сьогоднішній день версій.

ІТ сфера є однією з найдинамічніших у світі. Постійно з'являються нові мови програмування, програмні ресурси і бази, технологічні рішення. Постійно виходять доповнення і зміни вже існуючих інструментів розробки, і, як наслідок, змінюються вимоги до претендентів на ринку праці.

Користувач, який бажає отримати сучасну ІТ професію важливо отримувати знання про технології, актуальні саме на сьогоднішній день, а також перспективні на день завтрашній.

З цієї причини, користувач вибираючи для себе курси програмування, звертає увагу на те, як часто переглядається навчальна програма курсів, наскільки детально вона опрацьована. Йому важливо розуміти, наскільки дана програма близька йому.

Мета і завдання дослідження. Метою курсової роботи є розробка бази даних та макету веб-сайту на якому можна переглянути та зареєструватися на курси програмування.

Виходячи з цього, для досягнення мети в роботі поставлено й вирішено такі теоретичні та практичні завдання.

- охарактеризувати інформаційну систему для курсів програмування;
- проаналізувати предметну область;
- розробити архітектуру та спроектувати структуру бази даних;
- розробити інформаційну систему(макет веб-сайту), що дає змогу переглянути курси програмування та зареєструватися на них.

Об'єкт дослідження є особливість розробки інформаційної системи для курсів програмування компанії за допомогою системи управління базами даних MySQL.

Предмет досліджень є теоретичні, методичні та практичні аспекти розробки програмного забезпечення за допомогою різних мовою проектування.

Використане програмне забезпечення. Для досягнення результату використовувалось середовище розробки Visual Studio Code, MySQL та мови програмування: HTML, CSS, JS.

Структура роботи. Курсова робота складається з трьох розділів («Аналіз вимог», «Розробка бази даних» та «Розробка веб-додатку»), висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 45 сторінок

РОЗДІЛ 1. АНАЛІЗ ВИМОГ

1.1 Постановка завдання

На сьогодні інформаційні технології займають велике місце в нашому житті. Сучасні інформаційні технології стають одним з найбільш прибуткових та швидко зростаючих секторів економіки. Інформація стала важливим виробничим і комерційним ресурсом.

Інформаційні технології дуже швидко перетворилися на життєво важливий стимул розвитку не тільки світової економіки, а й інших сфер людської діяльності. На сьогодні практично неможливо знайти сферу, в якій зараз не використовуються інформаційні технології.

Збільшується швидкість обміну інформацією і з'являється можливість проводити складні математичні розрахунки за кілька секунд і багато іншого. Інформаційні технології це один із сучасних способів спілкування, головними перевагами якого є загальнодоступність. Використовуючи інформаційні технології можна з легкістю отримати доступ до цікавить вас інформації, а також поспілкуватися з живою людиною. З одного боку це має негативний ефект, оскільки люди все менше спілкуються "вживу", при безпосередньому контакті, але з іншого боку дозволять спілкуватися з людиною, яка знаходиться на іншому кінці світу.

Зарахувавши підсумок, можна сказати, що інформаційні технології глибоко проникли в наше життя і сучасне суспільство, яке не зможе в нинішньому вигляді існувати без них.

Інформаційна система для курсів програмування призначена для ознайомлення користувача з курсами, є невеликий опис кожного курсу програмування. Також можна перелягнути менторів і почитати відгуки користувачів. Для того щоб записатися на курс, необхідно зареєструватися.

1.2 Розробка моделі

Інформаційна система курсів програмування має два елементи керування

- користувач
- адміністратор

Розглянемо usecase діаграму (рис. 1.1)

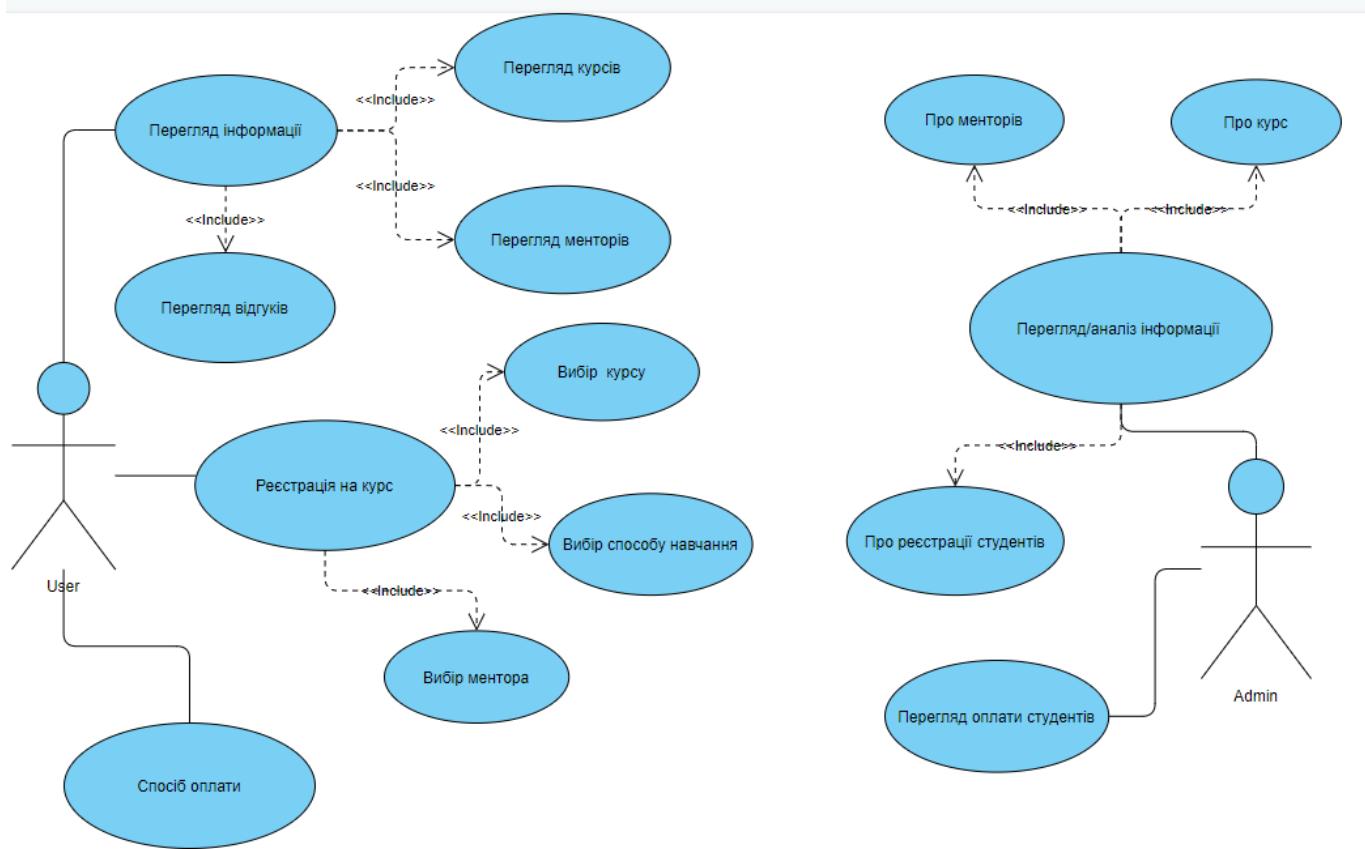


Рис. 1.1 Usecase діаграма курсів програмування

Користувач має змогу переглядати інформацію про курси і менторів. Щоб потрапити на курс, користувачу потрібно зареєструватися і оплатити обраний курс. Адміністратор в свою чергу може переглядати та аналізувати курси програмування. Адміністратор може відслідковувати статистику курсів, та викладачів. Також адмін має доступ до реєстрацій і оплат користувачів які записалися на курс.

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1 Опис моделі бази даних

База даних — це певний набір даних, які пов'язані між собою спільною ознакою або властивістю, та впорядковані, наприклад, за алфавітом.

Об'єднання великої кількості даних в єдину базу дає змогу для формування безлічі варіацій групування інформації — особисті дані клієнта, історія замовлень, каталог товарів та будь-що інше.

Головною перевагою БД є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам, які використовуються для баз даних, можна легко знаходити необхідні дані всього за декілька секунд. Також в базі даних існує певний взаємозв'язок інформації: зміна в одному рядку може спричинити зміни в інших рядках — це допомагає працювати з інформацією простіше і швидше.

Бази даних класифікують за різними критеріями.

За моделлю організації даних розрізняють такі бази даних:

- Ієрархічна. Ієрархічна база даних може бути представлена як дерево, що складається з об'єктів різних рівнів. Між об'єктами існують зв'язки типу «предок-нащадок». При цьому можлива ситуація, коли об'єкт не має нащадків або має їх декілька, тоді як у об'єкта-нащадка обов'язково тільки один предок.
- Мережна. Така база даних подібна до ієрархічної, за винятком того, що кожен об'єкт може мати більше одного предка.
- Реляційна. Реляційна база даних зберігає дані у вигляді таблиць. Найвживаніші СКБД використовують реляційну модель даних.
- Об'єктно-орієнтована. У базі даних цього виду дані оформляють у вигляді моделей об'єктів.

Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

- кожен елемент таблиці — один елемент даних;

- всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають одинаковий тип (числовий, символний тощо);
- кожен стовпчик має унікальне ім'я;
- одинакові рядки в таблиці відсутні;

порядок проходження рядків і стовпчиків може бути довільним.

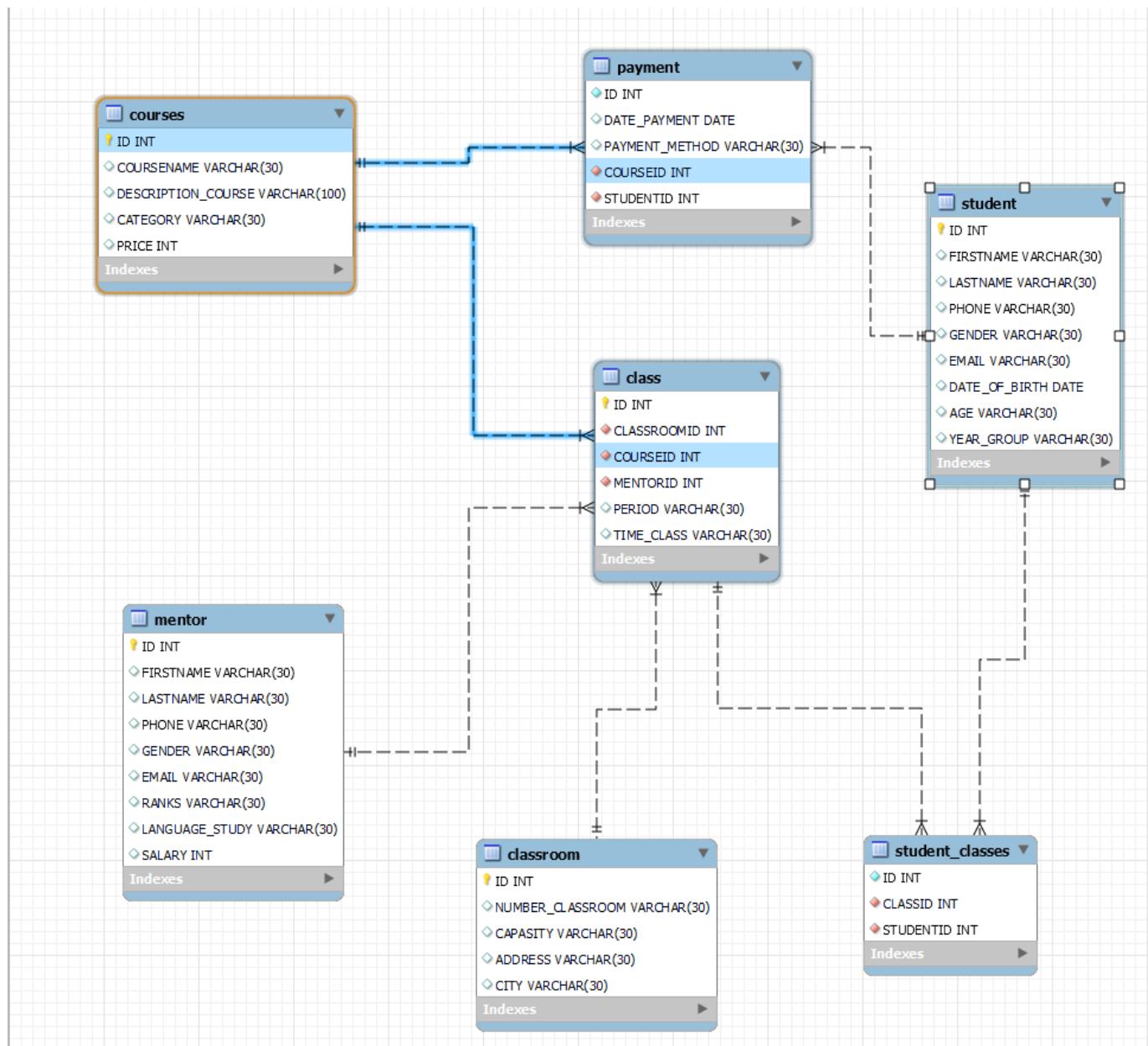


Рис. 2.1 Діаграма сущностей та зв'язків

На рисунку (Рис 2.1) можна побачити, що дана модель складається з 7 таблиць:

class, courses, payment, student, student_classes, classroom, mentor.

Class таблиця містить загальну інформацію про курси, яка складається з таких полів як:

1. CLASSID – індентифікаційний код інформації про курси;
2. CLASSROOMID - індентифікаційний код аудиторії;
3. MENTORID – індентифікаційний код ментора;
4. PERIOD – тривалість курсу;
5. TIME_CLASS – час заняття.

Courses таблиця містить інформацію про курси, яка складається з таких полів як:

1. COURSEID – індентифікаційний код курсу;
2. COURSENAME – назва курсу;
3. DESCRIPTION_COURSE – опис;
4. CATEGORY – категорія;
5. PRICE – ціна.

Payment таблиця містить інформацію про оплату, яка складається з таких полів як:

1. PAYMENTID – індентифікаційний код оплати;
2. DATA_PAYMENT – час оплати;
3. PAYMENT_METHOD – метод оплати;
4. COURSEID – індентифікаційний код курсу;
5. STUDENTID – індентифікаційний код студента;

Student таблиця містить інформацію про студента, яка складається з таких полів як:

1. STUDENTID – індентифікаційний код студента;
2. FIRSTNAME – ім'я студента;
3. LASTNAME – прізвище студента;
4. PHONE – номер телефону студента;
5. GENDER – стать студента;
6. EMAIL – пошта студента;
7. DATE_OF_BIRTH – дата народження студента;

8. AGE – індентифікаційний код студента;
9. YEAR_GROUP – індентифікаційний код студента;

Student_classes таблиця містить інформацію про групи студентів, яка складається з таких полів як:

1. STUDENT_CLASSES – індентифікаційний код групи студентів;
2. STUDENTID – індентифікаційний код студента;
3. CLASSID – індентифікаційний код інформації про курс.

Classroom таблиця містить інформацію про ментора, яка складається з таких полів як:

1. CLASSROOMID – індентифікаційний код аудиторії;
2. NUMBER_CLASSROOM – номер аудиторії;
3. CAPASITY – вмістимість аудиторії;
4. ADDRESS – вулиця на якій розташована аудиторія;
5. CITY – місто на якій розташована аудиторія;

Mentor таблиця містить інформацію про ментора, яка складається з таких полів як:

1. MENTORID – індентифікаційний код ментора;
2. FIRSTNAME – ім’я ментора;
3. LASTNAME – прізвище ментора;
4. PHONE – номер телефону ментора;
5. GENDER – стать ментора;
6. EMAIL – пошта ментора;
7. RANKS – рівень ментора;
8. LANGUAGE_STUDY – мова викладання ментора;
9. SALARY – зарплата ментора.

2.2 Визначення типів даних

Типи даних у SQL, як і будь-яка інша мова програмування, визначають, який тип даних може містити стовпець чи змінну. Тип даних може бути будь-яким - від символу, через серію символів, через числа, до дати та часу. Тип даних є дуже важливою концепцією щільно пов'язаного програмування. Це допомагає вирішити більшість помилок сумісності типів під час самої компіляції програми.

Основні типи даних, які використовувались:

- VARCHAR – може зберігати не більше 255 символів. На відміну від CHAR, для зберігання значення даного типу виділяється необхідна кількість пам'яті;
- INT – діапазон від -2 147 483 648 до 2 147 483 647;
- DATE – Дата. Формат: YYYY-MM-DD Підтримується діапазон від '1000-01-01' до '9999-12-31'

Таблиця 2.1

Склад таблиці «class»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	CLASSID	INT	11
2	CLASSROOMID	INT	11
3	COURSEID	INT	11
4	MENTOR	INT	11
5	PERIOD	VARCHAR	30
6	TIME_CLASS	VARCHAR	30

Таблиця 2.2

Склад таблиці «courses»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	COURSEID	INT	11
2	COURSENAME	VARCHAR	30
3	DESCRIPTION_COURSE	VARCHAR	30
4	CATEGORY	VARCHAR	30
5	PRICE	INT	11

Таблиця 2.3

Склад таблиці «payment»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	PAYMENTID	INT	11
2	DATA_PAYMENT	DATE	9999
3	PAYMENT_METHOD	VARCHAR	30
4	COURSEID	INT	11
5	STUDENTID	INT	11

Таблиця 2.4

Склад таблиці «student»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	STUDENTID	INT	11
2	FISTNAME	VARCHAR	30
3	LASTNAME	VARCHAR	30
4	COURSEID	INT	11
5	PHONE	VARCHAR	30
6	GENDER	VARCHAR	30
7	EMAIL	VARCHAR	30
8	DATE_OF_BIRTH	DATE	9999
9	AGE	VARCHAR	30
10	YEAR_GROUP	VARCHAR	30

Таблиця 2.5

Склад таблиці «student_classes»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	STUDENT CLASSESID	INT	11
2	STUDENTID	INT	11
3	CLASSID	INT	11

Таблиця 2.6

Склад таблиці «classroom»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	CLASSROOMID	INT	11
2	NUMBER_CLASSROOM	VARCHAR	30
3	CAPASITY	VARCHAR	30
4	ADDRESS	VARCHAR	30
5	CITY	VARCHAR	30

Таблиця 2.7

Склад таблиці «mentor»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	MENTORID	INT	11
2	FISTNAME	VARCHAR	30
3	LASTNAME	VARCHAR	30
5	PHONE	VARCHAR	30
6	GENDER	VARCHAR	30
7	EMAIL	VARCHAR	30
8	RANKS	VARCHAR	30
9	LANGUAGE_STUDY	VARCHAR	30
10	SALARY	INT	30

2.3 Обмеження цілісності даних

Обмеження цілісності — це правила, які обмежують усі можливі стани бази даних, а також переходи з одного стану в інший. Таким чином, обмеження цілісності визначають множину «допустимих» станів і переходів між ними. База даних перебуває в цілісному стані, якщо вона відповідає всім визначенням для неї вимогам цілісності.

Є такі типи правил цілісності

- CHECK — Контроль допустимих значень атрибутів;
- NOT NULL/NNULL — Заборона/ дозвіл на використання не заданих або не визначених значень.
- UNIQUE — Контроль унікальності значень атрибутів.
- PRIMARY KEY — Первинний ключ.

PRIMARY KEY (PK) використовується для ідентифікації рядків таблиці, в нього є певні особливості:

- В одній таблиці БД може бути тільки один PK.
- Рядки, які мають це правило, не можуть мати не визначені або не задані дані.
- FOREIGN KEY — Зовнішній ключ.

FOREIGN KEY (FG) використовується для того, щоб зв'язати дві таблиці, в нього є такі особливості:

- Поле у головній таблиці, на яке буде адресуватися FG повинне бути оголошене PK або UNIQUE.
- Поле у головній таблиці і поле з якої відбувається адресація повинні мати один й теж самий тип даних

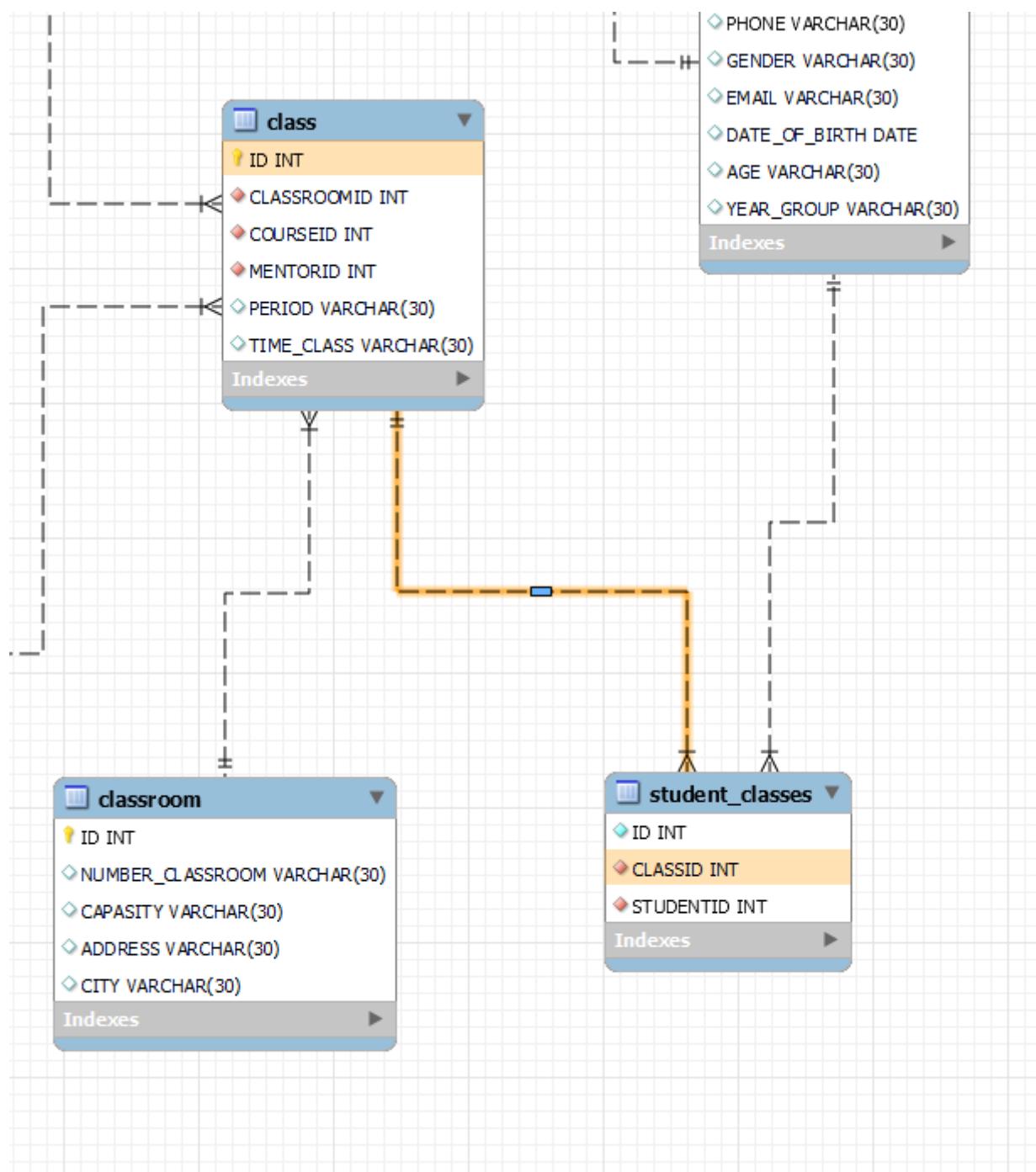


Рис. 2.2 Діаграма зв'язків створених за допомогою PRIMARY KEY і FOREIGN KEY

При розробці даної бази даних „курси програмування” було використано: primary key, foreign key(Рис. 2.2)

2.4 Реалізація SQL-скрипту

До бази даних курсів програмування були підібрані такі запити:

```
SELECT      PAYMENT.ID,           STUDENT.FIRSTNAME,        STUDENT.LASTNAME,
PAYMENT.DATE_PAYMENT, COURSENAME, PRICE FROM PAYMENT
LEFT JOIN STUDENT ON PAYMENT.STUDENTID = STUDENT.ID
LEFT JOIN COURSES ON PAYMENT.COURSEID = COURSES.ID
WHERE COURSENAME = 'JAVA DEVELOPER';
```

Цей запит показує скільки студентів обрали курс JAVA DEVELOPER

The screenshot shows a software interface for viewing database results. At the top, there are tabs for 'Result Grid' (which is selected), 'List', 'Filter Rows:', and 'Export:' options. Below the grid, there are buttons for 'Wrap Cell Content' and other export formats. The data grid has columns labeled ID, FIRSTNAME, LASTNAME, DATE_PAYMENT, COURSENAME, and PRICE. There are two rows of data: the first row contains ID 5, FIRSTNAME OLEH, LASTNAME BARYLKO, DATE_PAYMENT 2022-03-01, COURSENAME JAVA DEVELOPER, and PRICE 16200; the second row contains ID 10, FIRSTNAME NAZARIY, LASTNAME OLEKSEVYCH, DATE_PAYMENT 2022-03-15, COURSENAME JAVA DEVELOPER, and PRICE 16200.

	ID	FIRSTNAME	LASTNAME	DATE_PAYMENT	COURSENAME	PRICE
▶	5	OLEH	BARYLKO	2022-03-01	JAVA DEVELOPER	16200
	10	NAZARIY	OLEKSEVYCH	2022-03-15	JAVA DEVELOPER	16200

Рис 2.3 Результат виконання запиту

```
SELECT      CLASS.ID,           MENTOR.FIRSTNAME,        MENTOR.LASTNAME,        COURSENAME,
NUMBER_CLASSROOM,    CAPASITY,     ADDRESS,          CITY,          PERIOD,          TIME_CLASS,
LANGUAGE_STUDY FROM CLASS
LEFT JOIN MENTOR  on  CLASS.MENTORID = MENTOR.ID
LEFT JOIN COURSES ON CLASS.COURSEID = COURSES.ID
LEFT JOIN CLASSROOM ON CLASS.CLASSROOMID = CLASSROOM.ID ;
```

Цей запит демонструє повну інформацію про курс, назву курсу, де він проходить, хто викладає, якою мовою викладає, тривалість курсу.

ID	FIRSTNAME	LASTNAME	COURSENAME	NUMBER_CLASSROOM	CAPASITY	ADDRESS	CITY	PERIOD	TIME_CLASS	LANGUAGE_STUDY
1	TARAS	KRAVETS	FRONTEND DEVELOPER	101	BIG	PASICHNA 30	LVIV	2022-06-01 to 2022-12-01	2 HOUR	English and Ukrainian
2	YURIY	SALABAY	JAVA DEVELOPER	102	SMALL	PASICHNA 30	LVIV	2022-07-01 to 2022-01-01	2 HOUR	Ukrainian
3	OLEKSAND	HORODULOVKIY	C# DEVELOPER	310	BIG	PASICHNA 30	LVIV	2022-06-01 to 2022-12-01	2 HOUR	Ukrainian
4	OLEH	DUBEY	PYTHON DEVELOPER	312	SMALL	PASICHNA 30	LVIV	2022-07-01 to 2023-01-01	2 HOUR	English and Ukrainian
5	TARAS	KRAVETS	FRONTEND DEVELOPER ADVANCED	204	BIG	LVIVSKA 98A	LVIV	2022-04-01 to 2022-10-01	3 HOUR	English and Ukrainian
6	YULIA	KOVALSKA	JAVA DEVELOPER ADVANCED	206	SMALL	LVIVSKA 98A	LVIV	2022-04-01 to 2022-10-01	3 HOUR	Ukrainian
7	ROMAN	VOSKALO	C# DEVELOPER ADVANCED	201	BIG	LVIVSKA 98A	LVIV	2022-05-01 to 2022-11-01	3 HOUR	Ukrainian
8	EVGEN	SUMONENKO	PYTHON ADVANCED	107	SMALL	SHASHKEVICHNA 101	LVIV	2022-06-01 to 2022-11-01	3 HOUR	Ukrainian
9	OREST	GYPALO	FAST FRONTEND DEVELOPER	109	BIG	SHASHKEVICHNA 101	LVIV	2022-08-01 to 2022-12-01	1.5 HOUR	Ukrainian
10	EVGEN	SUMONENKO	FAST JAVA DEVELOPER	103	SMALL	SHASHKEVICHNA 101	LVIV	2022-08-01 to 2022-12-01	1.5 HOUR	Ukrainian
11	VIKTORIA	DEMUD	FAST C# DEVELOPER	105	BIG	SHASHKEVICHNA 101	LVIV	2022-06-01 to 2022-10-01	1.5 HOUR	Ukrainian
12	TARAS	KRAVETS	FAST PYTHON DEVELOPER	101	SMALL	SHASHKEVICHNA 101	LVIV	2022-07-01 to 2022-11-01	1.5 HOUR	English and Ukrainian

Рис. 2.4 Результат виконання запиту

```

SELECT          PAYMENT.ID,                      STUDENT.FIRSTNAME,          STUDENT.LASTNAME,
PAYMENT.PAYMENT_METHOD, DATE_PAYMENT FROM PAYMENT
LEFT JOIN STUDENT ON PAYMENT.STUDENTID = STUDENT.ID
WHERE PAYMENT_METHOD = 'CASH';

```

Цей запит показує який студент здійснив оплату, яким методом і коли.

ID	FIRSTNAME	LASTNAME	PAYMENT_METHOD	DATE_PAYMENT
1	BOHDAN	KUZMA	CASH	2022-03-01
3	OLEH	SEMKIV	CASH	2022-03-01
5	OLEH	BARYLKO	CASH	2022-03-01
8	OLEH	MRZA	CASH	2022-03-15
11	OLEH	VASYTUK	CASH	2022-03-15
14	ANDRIY	YABLONSKIY	CASH	2022-03-31
16	VITALIY	ANDRYSISHIN	CASH	2022-03-31
18	BOHDAN	KAPOHRUS	CASH	2022-03-31
20	SERHIY	TEKH	CASH	2022-03-31

Рис. 2.5 Результат виконання запиту

```

SELECT MENTOR.ID, FIRSTNAME, LASTNAME FROM MENTOR
LEFT JOIN CLASS ON CLASS.MENTORID = MENTOR.ID
GROUP BY MENTOR.ID HAVING COUNT(COURSEID) > 1;

```

Цей запит показує менторів які ведуть більше 1 курсу

	ID	FIRSTNAME	LASTNAME
▶	1	TARAS	KRAVETS
	7	EVGEN	SUMONENKO

Рис. 2.6 Результат виконання запиту

```

SELECT      CLASS.ID,          MENTOR.FIRSTNAME,      MENTOR.LASTNAME,      COURSENAME,
LANGUAGE_STUDY,  SALARY FROM CLASS
LEFT JOIN MENTOR  on  CLASS.MENTORID = MENTOR.ID
LEFT JOIN COURSES ON CLASS.COURSEID = COURSES.ID;

```

Цей запит демонструє яку ментор отримує заробітню плату і за який курс

ID	FIRSTNAME	LASTNAME	COURSENAME	LANGUAGE_STUDY	SALARY
1	TARAS	KRAVETS	FRONTEND DEVELOPER	English and Ukrainian	22500
2	YURIY	SALABAY	JAVA DEVELOPER	Ukrainian	15000
3	OLEKSAND	HORODULOVKIY	C# DEVELOPER	Ukrainian	15000
4	OLEH	DUBEY	PYTHON DEVELOPER	English and Ukrainian	16500
5	TARAS	KRAVETS	FRONTEND DEVELOPER ADVANCED	English and Ukrainian	22500
6	YULIA	KOVALSKA	JAVA DEVELOPER ADVANCED	Ukrainian	14000
7	ROMAN	VOSKALO	C# DEVELOPER ADVANCED	Ukrainian	14000
8	EVGEN	SUMONENKO	PYTHON ADVANCED	Ukrainian	20000
9	OREST	GYPALO	FAST FRONTEND DEVELOPER	Ukrainian	14000
10	EVGEN	SUMONENKO	FAST JAVA DEVELOPER	Ukrainian	20000
11	VIKTORIA	DEMUD	FAST C# DEVELOPER	Ukrainian	14000
12	TARAS	KRAVETS	FAST PYTHON DEVELOPER	English and Ukrainian	22500

Рис. 2.7 Результат виконання запиту

РОЗДІЛ 3. РОЗРОБКА МАКЕТУ ВЕБ-САЙТУ

3.1 Структура і макет веб-сайту

Ключова складова веб-сайту є структура. Вона впливає на його якість, зручність для користувачів і індексацію пошуковими системами. За допомогою коректно складеної структури вирішується ряд завдань.

Структура web сайту — це логічна побудова сторінок ресурсу, його архітектура, позиції елементів оформлення, зв'язок між розділами і сторінками всередині сайту. Сайт курсів програмування за своєю структурою має вигляд лендінгу. Як правило, лендінг складається з однієї сторінки, на якій розміщена вся необхідна користувачеві інформація про курси. Потрапляючи на сторінку курсів програмування користувач може ознайомитись з інформацією про саму компанію, з її викладачами та переліком курсів і відгуками. Структуру сайту складається з таких елементів:

- Шапка сайту. Містить назву компанії, коротку інформацію, меню і кнопку записатися на курс (Рис 3.1);

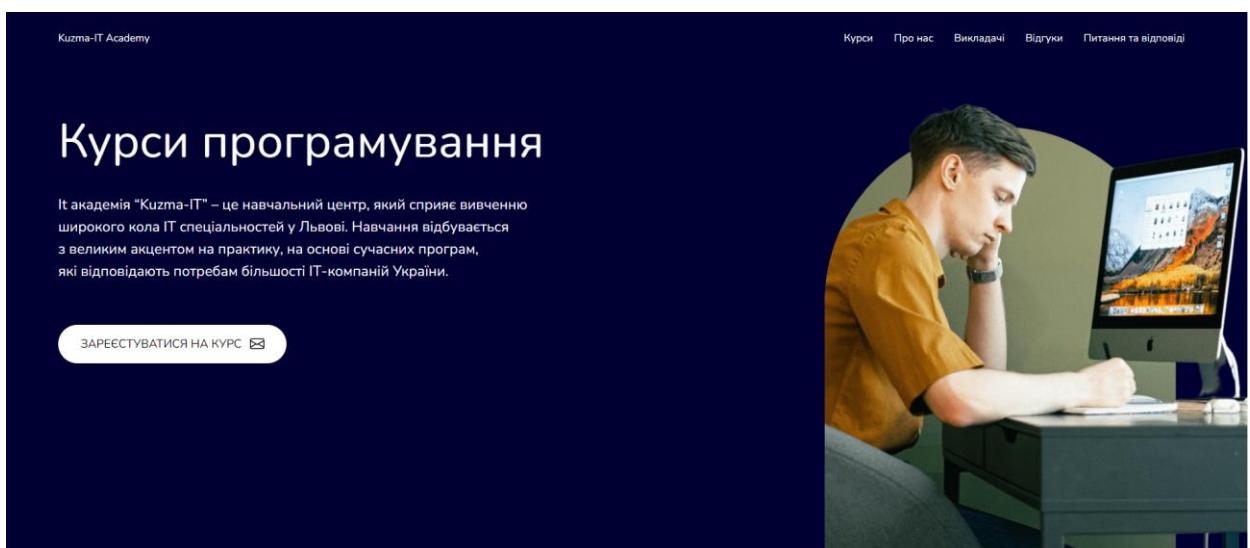


Рис 3.1 Шапка сайту

- Секція представлених курсів. Основний вміст складається з короткої інформації кожного представленого курсу (Рис.3.2);



Pредставлені курси



JAVA DEVELOPER

Мова Java багато років займає перші рядки рейтингів мов програмування.

Знаючи цю мову, ви можете створювати найрізноманітніші кросплатформені додатки – веб сайти, мобільні додатки, корпоративні системи.

Ціна - 14500 грн



FRONTEND DEVELOPER

FrontEnd розробник – одна з найбільш популярних IT спеціальностей. В цьому курсі Ви навчитеся створювати веб-сторінки, використовуючи HTML, CSS і Bootstrap.

Ви вивчите всі елементи мови JavaScript і TypeScript

Ціна - 16800 грн

Рис. 3.2 Представлені курси

- Інформаційний сегмент про компанію. Інформація про компанію (Рис. 3.3);

Про Нас

10
Років на ринку

25 тис
Студентів

2 400 \$
середня заробітна плата
Front-end розробника

587
вакансій, що підходять тобі
після проходження курсу на сайті [work.ua](#)

Комплексна система навчання в нас – це:

Вивчення теорії і практики програмування на заняттях в навчальному центрі Виконання практичних завдань, створення власних курсових проектів Програма отримання досвіду після завершення курсу навчання Власна система онлайн тестування і сертифікації фахівців Ми протягом багатьох років входимо до п'ятірки кращих навчальних центрів в Україні. Навчання проводять сертифіковані тренери з великим досвідом у галузі розробки програмного забезпечення.

Рис. 3.3 Інформація про компанію

- Секція викладачів та менторів. Ознайомлює користувача з викладачами які надаються компанією (Додаток В);
- Секція відгуків. В цій секції користувач може передивитися відгуки різних студентів які закінчили курс. (Додаток Г);
- Секція питання-відповіді. Секція тут користувач може подивится найактуальніші питання (Додаток Д);

- Футер. В цій секції є карта з адресою курсів, контакти і посилання соціальні мережі (Додаток Е);

3.2 Програмування клієнтської частини

Для написання коду я використовував Visual Studio Code — засіб для створення, редагування та зневадження сучасних вебзастосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

При роботі використовував такі мови програмування: HTML, CSS, JS

HTML — стандартизована мова розмітки документів для перегляду вебсторінок у браузері. Веб-браузери отримують HTML документ від сервера за протоколами HTTP/HTTPS або відкривають з локального диска, далі інтерпретують код в інтерфейс, який відображатиметься на екрані монітора.

CSS — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних.

JS — динамічна, об'єктно-орієнтована прототипна мова програмування.

Приклад використання мови HTML

Лістинг 1.1

```
<section class="top-nav">
  <div style="color: white;">
    Kuzma-IT Academy
  </div>
  <input id="menu-toggle" type="checkbox" />
  <label class='menu-button-container' for="menu-toggle">
    <div class='menu-button'></div>
  </label>
  <ul class="menu">
    <li><a class="nav-link-color" style="text-decoration: none;" href="#course">Курси</a></li>
    <li><a class="nav-link-color" style="text-decoration: none;" href="#aboutUs">Про нас</a></li>
```

```

<li><a class="nav-link-color" style="text-decoration: none;" href="#prof">Викладачі</a></li>
<li><a class="nav-link-color" style="text-decoration: none;" href="#revision">Відгуки</a></li>
<li><a class="nav-link-color" style="text-decoration: none;" href="#questions">Питання та відповіді</a></li>
</ul>
</section>
```

Приклад використання мови CSS

Лістинг 1.2

```
.menu-button-container {
    display: none;
    height: 100%;
    width: 30px;
    cursor: pointer;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
```

Приклад використання мови JS

Лістинг 1.3

```
document.querySelector('.btn-danger').addEventListener('click', () => {
    document.querySelector('.firstName').value = "";
    document.querySelector('.secondName').value = "";
    document.querySelector('.selectCourse').value = 'value1';
    document.querySelector('.radioBut').checked = true;
    document.querySelector('.selectProf').value = 'value1';
    document.querySelector('.phoneValue').value = "";
```

На сайті також зроблено адаптив під мобільні пристрой

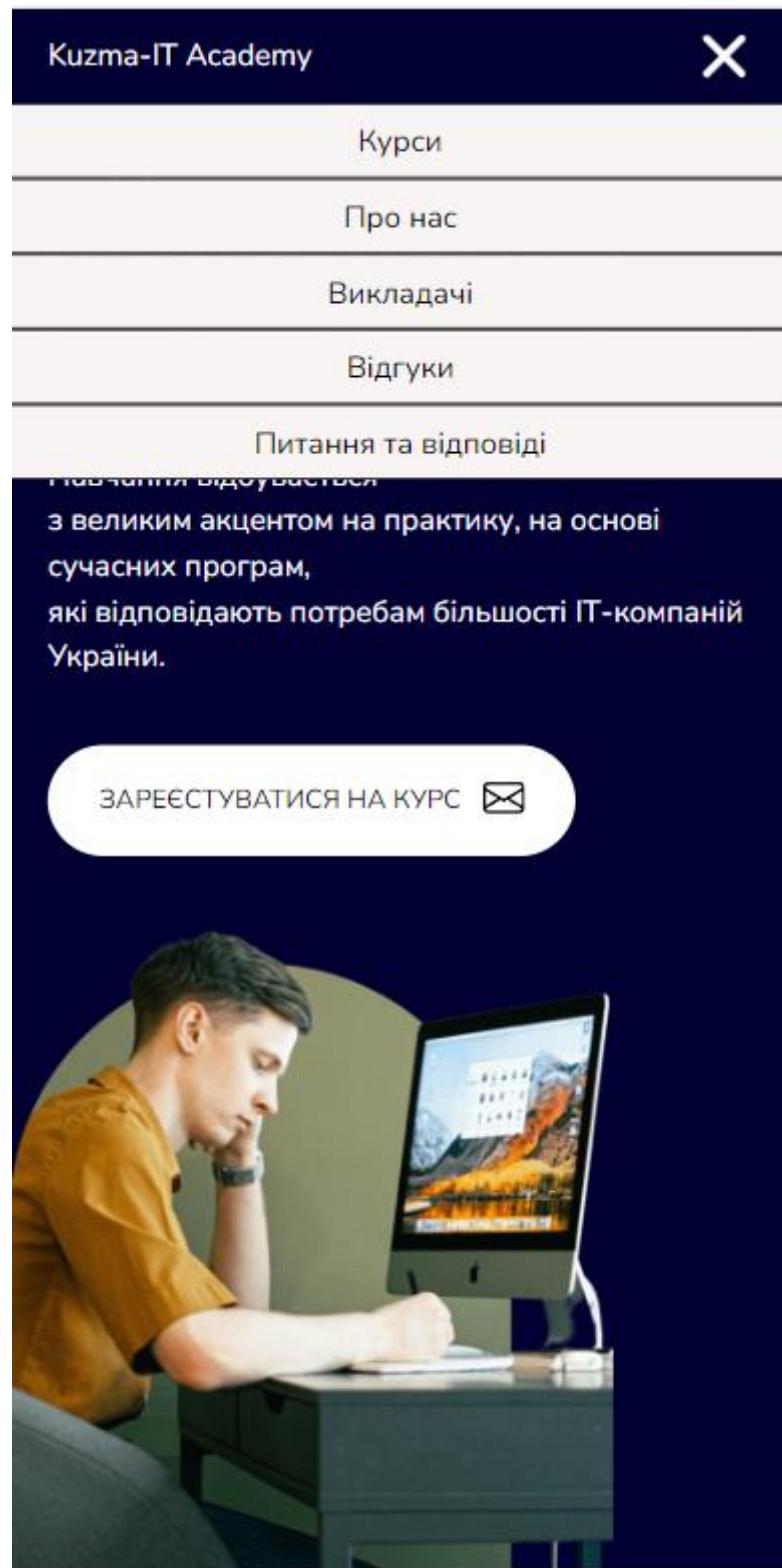


Рис 3.4 Адаптивна версія шапки сайту

ВИСНОВКИ

Під час виконання курсової роботи проаналізовано можливості використання HTML, CSS, JS та роботу із базами даних. Для закріплення набутих знань розроблено базу даних курсів програмування, а також сайт, який дає змогу зареєструватися на обраний курс. На початковому етапі виконання було проаналізовано предметну область, поставлена мета і завдання у відповідності до змісту роботи та виданого завдання, а саме для інформаційної системи курсів програмування.

Після цього у другому розділі була реалізовано база даних. А саме: детальний аналіз предметної області, EER-модель. Також визначені первинний і вторинний ключі. Створено 7 таблиць («class», «student», «courses», «classroom», «student_classes», «payment», «mentor»)

В останньому розділі висвітлено зовнішній вигляд сайту, його макет. Наведені приклади коду програмування користувацької частини сайту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

Додаток А



PYTHON DEVELOPER

Python досить проста мова для вивчення, але при цьому може використовуватися для вирішення дуже складних задач.

Мета курсу Python розробник – дати необхідну базу знань і навичок для застосування на практиці. Ви навчитеся працювати з базами даних і фреймворком Django для створення web-додатків.

Ціна - 12300 грн



C#/ .NET DEVELOPER

.NET розробник – популярна та високооплачувана спеціальність. Основна мова програмування, що використовує .NET розробник – C #.

Починаючи з елементарних речей і закінчуючи новітніми технологіями, які потрібні для комерційної розробки. Ви крок за кроком опануєте все, що повинен знати .NET

Ціна - 14800 грн

Рис 3.5 Вигляд представлених курсів

Додаток Б

Готуємся до реального життя в IT компанії

Захист диплому - це типовий сценарій, який підготує тебе до роботи Front-end розробником. Приступаючи до розробки дипломного проекту ти використаєш всі технології, які вивчив протягом курсу, а також вдосконалиш свої софт скілі при його захисті на випускному перед нашим відділом працевлаштування та партнерами з IT компаній.



Рис 3.6 Секція з інформацією про компанію

Додаток В

Викладачі та ментори

Ми пишаємся нашою командою!



Тарас Погорецький

Frontend Developer компанії
CodeLions.

6+ років досвіду. Викладач та куратор курсу Frontend Developer. Hard skills: HTML | CSS | SASS/SCSS, JavaScript, TypeScript, Angular Material, GitHub, RxJS, ES6, AJAX, Git, Bootstrap, LESS, jQuery, jQuery UI, Gulp.js, також є досвід у проектах написаними мовою Java.



Олександр Гуцуляк

Python Developer, NDA Company

6+ років досвіду IT. Hard Skills: Python; Selenium / Django / Pandas / Scrapy / BeautifulSoup. CMS: WordPress відповідно - html / css / js / php (laravel) / XPath. БД: MySQL / SQLite / MongoDB / C# / Java. Практичний досвід: фулстек по WordPress, автоматизоване тестування, автоматизація процесів.



Добрянський Юрій

Java Software Engineer

Володимир займається програмуванням понад 4 роки, 3 з яких програмує на Java. Розробляє ERP-рішення на основі хмарних технологій. Володіє знаннями Java, JavaScript з багатьма його фреймворками, бази даних MySQL, PostgreSQL, MongoDB / C# / Java. Має досвід у адмініструванні Ubuntu-based

Рис 3.7 Секція з викладачами

Додаток Г

Відгуки наших клієнтів



Курс Frontend

Дізнався про курси від знайомого. Він же й порадив придивитись до спеціальності Frontend, оскільки сам вже довгий час працює фронтендинчиком. Після відвідування сайту вагався недовго, і після коротких роздумів записався сюди на курси. Під час навчання особливих складнощів не виникало.



Роман Кучик
28 років



Курс JAVA

Спеціальність Java Developer зацікавила тим, що було багато докладної інформації про курс: на час, що буде вивчатися і т.д. Набув багато навичок та умінь, не шкодуючи, що зважився піти на курси, для мене це виявилося доступнішим для розуміння, ніж навчатися самому.



Сергій Рибак
31 рік



Курс C#

Уже більше 3 років я рекомендую курси всім своїм знайомим, які вивчають курси для того, щоб вивчити нові технології або підвищити свою кваліфікацію та упорядкувати вже наявні знання. Курси по C# для початківців зовсім непрості і зкладають дуже хороши базові знання платформи .Net і ООП



Наталія Шулік
27 років

Рис 3.8 Секція з відгуками

Додаток Д

Питання та відповіді

Я нічого не розумію в програмуванні. Чи зможу я навчатися на ваших курсах?

Так. Більшість курсів розрахована на новачків. Технічна освіта тож не обов'язкова. Якщо у Вас є сумніви, або ж ви вагаєтесь запрошуємо пройти безкоштовний курс по "Основах програмування".

Які мінімальні знання та вимоги для початку навчання на курсах програмування?

Чи є у Вас пробні заняття?

Чим Kuzma-IT відрізняється від інших навчальних закладів?

Чи потрібно носити ноутбук на заняття?

Не потрібно, оскільки усі аудиторії Kuzma-IT Academy обладнані необхідною сучасною технікою та безпровідним інтернетом.

Чи задаєте ви домашні завдання?

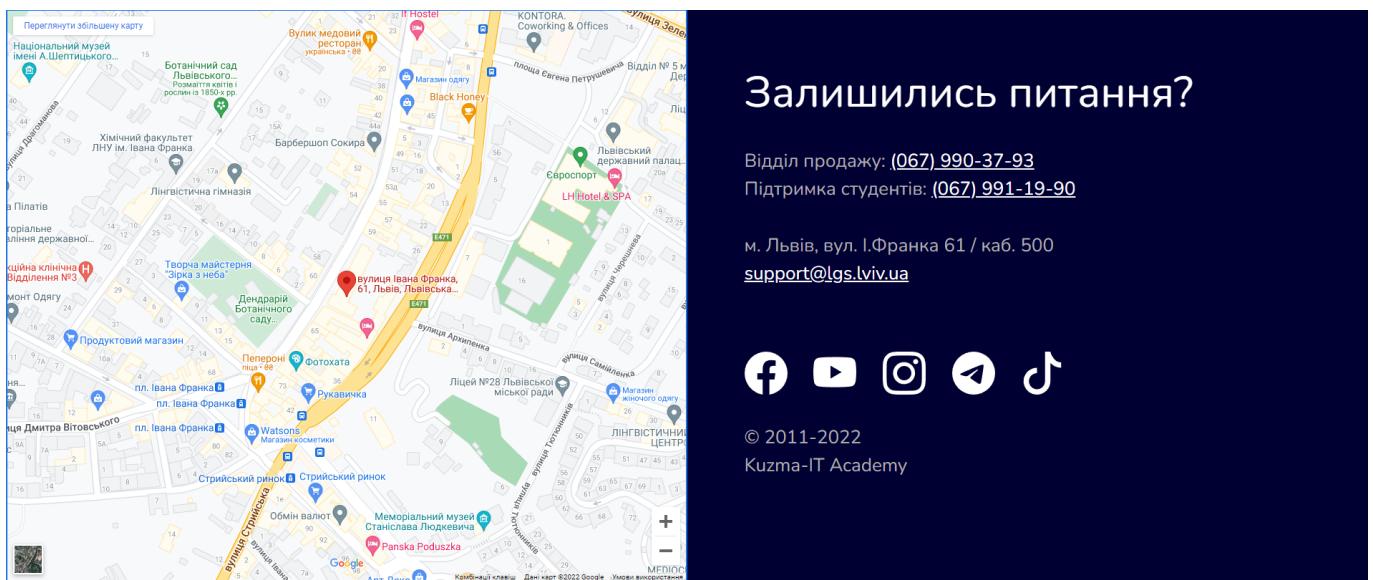
Як зазвичай проходить заняття?

Все залежить від теми. Зазвичай перші 10-15 хвилин відведені на обговорення домашнього завдання та питання по ньому, весь наступний час практичне заняття під час якого лектор показує на проекторі новий матеріал який слухачі відразу тестиють. Конспекти ніхто не пише, презентації та теоретичний матеріал подається в електронній формі, домашні завдання також віддаються на емейл.

Чи ви видаєте сертифікат про закінчення курсів?

Рис 3.9 Секція питання-відповіді

Додаток Е



Залишились питання?

Відділ продажу: **(067) 990-37-93**
Підтримка студентів: **(067) 991-19-90**

м. Львів, вул. І.Франка 61 / каб. 500
support@lgs.lviv.ua



© 2011-2022
Kuzma-IT Academy

Рис 3.10 Футер сайту

Додаток Е

```
DROP DATABASE COURSE;

CREATE DATABASE COURSE;

USE COURSE;

CREATE TABLE CLASS (ID INT NOT NULL, CLASSROOMID INT NOT NULL,
COURSEID INT NOT NULL, MENTORID INT NOT NULL, PERIOD VARCHAR(30),
TIME_CLASS VARCHAR(30));

CREATE TABLE COURSES (ID INT NOT NULL, COURSENAMe VARCHAR(30),
DESCRIPTION_COURSE VARCHAR(100), CATEGORY VARCHAR(30), PRICE INT );

CREATE TABLE STUDENT_CLASSES (ID INT NOT NULL, CLASSID INT NOT NULL,
STUDENTID INT NOT NULL);

CREATE TABLE STUDENT (ID INT NOT NULL, FIRSTNAME VARCHAR(30), LASTNAME
VARCHAR(30), PHONE VARCHAR(30), GENDER VARCHAR(30), EMAIL VARCHAR(30),
DATE_OF_BIRTH DATE, AGE VARCHAR(30), YEAR_GROUP VARCHAR (30) );

CREATE TABLE MENTOR (ID INT NOT NULL, FIRSTNAME VARCHAR(30), LASTNAME
VARCHAR(30), PHONE VARCHAR(30), GENDER VARCHAR(30), EMAIL VARCHAR(30),
RANKS VARCHAR(30), LANGUAGE_STUDY VARCHAR(30), SALARY INT);

CREATE TABLE CLASSROOM (ID INT NOT NULL, NUMBER_CLASSROOM VARCHAR(30),
CAPASITY VARCHAR(30), ADDRESS VARCHAR(30), CITY VARCHAR(30) );

CREATE TABLE PAYMENT (ID INT NOT NULL, DATE_PAYMENT DATE,
PAYMENT_METHOD VARCHAR(30), COURSEID INT NOT NULL, STUDENTID INT NOT
NULL );

ALTER TABLE COURSES ADD CONSTRAINT COURSEID PRIMARY KEY (ID);

ALTER TABLE CLASS ADD FOREIGN KEY (COURSEID) REFERENCES COURSES(ID);

ALTER TABLE MENTOR ADD CONSTRAINT MENTORID PRIMARY KEY (ID);

ALTER TABLE CLASS ADD FOREIGN KEY (MENTORID) REFERENCES MENTOR(ID);
```

```
ALTER TABLE STUDENT ADD CONSTRAINT STUDENTID PRIMARY KEY (ID);

ALTER TABLE STUDENT_CLASSES ADD FOREIGN KEY (STUDENTID) REFERENCES
STUDENT(ID);
```

```
ALTER TABLE CLASS ADD CONSTRAINT CLASSID PRIMARY KEY (ID);

ALTER TABLE STUDENT_CLASSES ADD FOREIGN KEY (CLASSID) REFERENCES
CLASS(ID);
```

```
ALTER TABLE CLASSROOM ADD CONSTRAINT CLASSROOMID PRIMARY KEY (ID);

ALTER TABLE CLASS ADD FOREIGN KEY (CLASSROOMID) REFERENCES
CLASSROOM(ID);
```

```
ALTER TABLE PAYMENT ADD FOREIGN KEY (STUDENTID) REFERENCES
STUDENT(ID);
```

```
ALTER TABLE PAYMENT ADD FOREIGN KEY (COURSEID) REFERENCES COURSES(ID);
```

Заповнення таблиць даними

```
INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (1, 'BOHDAN',
'KUZMA', '380685054926', 'bohdankuzma@gmail.com', 'M', '2002-01-07',
'20', '18-30');
```

```
INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (2, 'ROMAN', 'BILL',
'380687054526', 'romanbill@gmail.com', 'M', '2002-10-30', '19', '18-
30');
```

```
INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (3, 'OLEH',
'SEMKIV', '380685634956', 'olehsemkiv@gmail.com', 'M', '2002-02-
17', '20', '18-30');
```

```
INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (4, 'YAROSLAV',
'RUVKO', '380684074626', 'yaroslavruvko@gmail.com', 'M', '2001-10-
11', '20', '18-30');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (5, 'OLEH',
'BARYLKO', '380683321742', 'olehbarylko@gmail.com', 'M', '2001-09-
18', '20', '18-30');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (6, 'DENYS',
'SUPKO', '380686134853', 'denyssypko@gmail.com', 'M', '2006-11-
25', '15', '0-18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (7, 'DIANA', 'IRZA',
'380683212626', 'dianairza@gmail.com', 'F', '2006-02-03', '16', '0-
18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (8, 'OLEH',
'MRZA', '380684272712', 'olehmrza@gmail.com', 'M', '2005-01-07', '17',
'0-18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (9, 'IHOR',
'TRYSH', '380683244712', 'ihortrysh@gmail.com', 'M', '2007-10-12', '14',
'0-18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (10, 'NAZARIY',
'OLEKSEVYCH', '380688234642', 'nazariyleksevych@gmail.com', 'M', '2008-
06-12', '13', '0-18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (11, 'OLEH',
'VASYTUK', '380683321742', 'olehbill@gmail.com', 'M', '2002-09-
18', '19', '18-30');
```

```
INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (12, 'DENYS',
'OLIYNUK', '380686134853', 'denysoliynuk@gmail.com', 'M', '2006-03-
01', '16', '0-18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (13, 'DIANA', 'HOVZUN',
'380683432626', 'dianahovzun@gmail.com', 'F', '2006-09-01', '15', '0-
18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (14, 'ANDRIY',
'YABLONSKIY', '380682172712', 'andriyyablonskiy@gmail.com', 'M', '2005-
10-07', '16', '0-18');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (15, 'IHOR',
'LESKIV', '380683054712', 'ihorleskiv@gmail.com', 'M', '1998-10-
12', '23', '18-30');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (16, 'VITALIY',
'ANDRYSISHIN', '380684134642', 'vitaliyandrysishin@gmail.com', 'M',
'1999-06-12', '22', '18-30');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (17, 'VITALIY', 'DUTKO',
'380683212126', 'vitaliydutko@gmail.com', 'M', '1986-03-30', '36', '30-
40');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (18, 'BOHDAN',
'KAPOHRUS', '380683262712', 'bohdankapohrus@gmail.com', 'M', '1989-02-
24', '33', '30-40');

INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (19, 'DENYS',
'ANDRYSIV', '380684243712', 'denysandrusiv@gmail.com', 'M', '1991-10-
12', '30', '30-40');
```

```
INSERT INTO STUDENT(ID, FIRSTNAME, LASTNAME, PHONE, EMAIL, GENDER,
DATE_OF_BIRTH, AGE, YEAR_GROUP) VALUES (20, 'SERHIY',
'TEKH', '380682224642', 'serhiytek@gmail.com', 'M', '2002-03-04', '20',
'18-30');
```

```
INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE, GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (1, 'TARAS',
'KRAVETS', '380681223742', 'M', 'taraskravets@gmail.com', 'Senior',
'English and Ukrainian ', '18500');
```

```
INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE, GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (2, 'YULIA', 'KOVALSKA',
'380683044926', 'F', 'yuliakovalska@gmail.com', 'Middle', 'Ukrainian',
'14000');
```

```
INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE, GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (3, 'OLEH',
'DUBEY', '380684024646', 'M', 'olehdubey@gmail.com', 'Middle', 'English
and Ukrainian ', '16500');
```

```
INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE, GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (4, 'YURIY',
'SALABAY', '380685054426', 'M', 'yuriysalabay@gmail.com', 'Senior',
'Ukrainian', '15000');
```

```
INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE, GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (5, 'OLEKSAND',
'HORODULOVKIY', '380682044216', 'M', 'oleksandhorodulovsky@gmail.com',
'Senior', 'Ukrainian', '15000');
```

```
INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE, GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (6, 'ROMAN',
'VOSKALO', '380683054107', 'M', 'romanvoskalo@gmail.com', 'Middle',
'Ukrainian', '14000');
```

```
INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE, GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (7, 'EVGEN', 'SUMONENKO',
```

```
'380682344236', 'M', 'evgeniusmonenko@gmail.com', 'Senior',
'Ukrainian', '20000');

INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE,GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (8, 'OREST',
'GYPALO','380682344826', 'M', 'orestgypalo@gmail.com', 'Middle',
'Ukrainian', '14000');

INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE,GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (9, 'MYKHAILO', 'YUKALO',
'380682344236', 'M', 'mykhailoyukalo@gmail.com', 'Senior', 'English
and Ukrainian ', '18500');

INSERT INTO MENTOR(ID, FIRSTNAME, LASTNAME, PHONE,GENDER, EMAIL,
RANKS, LANGUAGE_STUDY, SALARY) VALUES (10, 'VIKTORIA',
'DEMUD','380682344826', 'F', 'viktoriademud@gmail.com', 'Middle',
'Ukrainian', '14000');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION_COURSE , CATEGORY,
PRICE) VALUES (1, 'FRONTEND DEVELOPER', 'HTML/CSS/JS','OFFLINE',
'14500');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION_COURSE , CATEGORY,
PRICE) VALUES (2, 'JAVA DEVELOPER', 'JAVA','OFFLINE', '16200');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION_COURSE , CATEGORY,
PRICE) VALUES (3, 'C# DEVELOPER', 'C#','OFFLINE', '12400');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION_COURSE , CATEGORY,
PRICE) VALUES (4, 'PYTHON DEVELOPER ', 'PYTHON','OFFLINE', '13100');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION_COURSE , CATEGORY,
PRICE) VALUES (5, 'FRONTEND DEVELOPER ADVANCED',
'HTML/CSS/JS','OFFLINE', '18500');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION_COURSE , CATEGORY,
PRICE) VALUES (6, 'JAVA DEVELOPER ADVANCED' , 'JAVA','OFFLINE',
'20000');
```

```
INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION.Course , CATEGORY,
PRICE) VALUES (7, 'C# DEVELOPER ADVANCED', 'C#','OFFLINE', '15450');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION.Course , CATEGORY,
PRICE) VALUES (8, 'PYTHON ADVANCED', 'PYTHON','OFFLINE', '17500');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION.Course , CATEGORY,
PRICE) VALUES (9, 'FAST FRONTEND DEVELOPER ',
'HTML/CSS/JS','OFFLINE', '9500');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION.Course , CATEGORY,
PRICE) VALUES (10, 'FAST JAVA DEVELOPER ' , 'JAVA','OFFLINE',
'11000');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION.Course , CATEGORY,
PRICE) VALUES (11, 'FAST C# DEVELOPER ' , 'C#','OFFLINE', '8600');

INSERT INTO COURSES(ID, COURSENAME , DESCRIPTION.Course , CATEGORY,
PRICE) VALUES (12, 'FAST PYTHON DEVELOPER', 'PYTHON','OFFLINE',
'9000');
```

```
INSERT INTO CLASSROOM (ID, NUMBER.CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (1, '101' , 'BIG','PASICHNA 30', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER.CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (2, '102' , 'SMALL','PASICHNA 30', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER.CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (3, '310', 'BIG','PASICHNA 30', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER.CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (4, '312', 'SMALL','PASICHNA 30', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER.CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (5, '204' , 'BIG','LVIVSKA 98A', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER.CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (6, '206' , 'SMALL','LVIVSKA 98A', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER.CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (7, '201', 'BIG','LVIVSKA 98A', 'LVIV');
```

```
INSERT INTO CLASSROOM (ID, NUMBER_CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (8, '107', 'SMALL','SHASHKEVICHА 101', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER_CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (9, '109' , 'BIG','SHASHKEVICHА 101', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER_CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (10, '103' , 'SMALL','SHASHKEVICHА 101', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER_CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (11, '105', 'BIG','SHASHKEVICHА 101', 'LVIV');

INSERT INTO CLASSROOM (ID, NUMBER_CLASSROOM, CAPASITY, ADDRESS , CITY
) VALUES (12, '101', 'SMALL','SHASHKEVICHА 101', 'LVIV');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(1,1,1,1,'2022-06-01 to 2022-12-01','2 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(2,2,2,4,'2022-07-01 to 2022-01-01','2 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(3,3,3,5,'2022-06-01 to 2022-12-01','2 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(4,4,4,3,'2022-07-01 to 2023-01-01','2 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(5,5,5,1,'2022-04-01 to 2022-10-01','3 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(6,6,6,2,'2022-04-01 to 2022-10-01','3 HOUR');
```

```
INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(7,7,7,6,'2022-05-01 to 2022-11-01','3 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(8,8,8,7,'2022-06-01 to 2022-11-01','3 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(9,9,9,8,'2022-08-01 to 2022-12-01','1.5 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(10,10,10,7,'2022-08-01 to 2022-12-01','1.5 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(11,11,11,10,'2022-06-01 to 2022-10-01','1.5 HOUR');

INSERT INTO CLASS (ID, CLASSROOMID, COURSEID, MENTORID, PERIOD,
TIME_CLASS) VALUES(12,12,12,1,'2022-07-01 to 2022-11-01','1.5 HOUR');

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (1,1,1);
INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (2,1,2);
INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (3,1,3);
INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (4,1,4);
INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (5,1,5);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (6,2,6);
INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (7,2,7);
```

```
INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (8,2,8);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (9,2,9);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (10,2,10);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (11,3,11);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (12,3,12);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (13,3,13);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (14,3,14);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (15,3,15);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (16,4,16);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (17,4,17);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (18,4,18);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (19,4,19);

INSERT INTO STUDENT_CLASSES(ID, CLASSID, STUDENTID) VALUES (20,4,20);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (1, '2022-03-01','CASH', '7', 1);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (2, '2022-03-01','CARD', '1', 2);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (3, '2022-03-01','CASH', '3', 3);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (4, '2022-03-01','CARD', '4', 4);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (5, '2022-03-01','CASH', '2', 5);
```

```
INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (6, '2022-03-15','CARD', '12', 6);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (7, '2022-03-15','CARD', '11', 7);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (8, '2022-03-15','CASH', '6', 8);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (9, '2022-03-15','CARD', '4', 9);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (10, '2022-03-15','CARD', '2', 10);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (11, '2022-03-15','CASH', '1', 11);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (12, '2022-03-15','CARD', '9', 12);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (13, '2022-03-31','CARD', '10', 13);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (14, '2022-03-31','CASH', '11', 14);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (15, '2022-03-31','CARD', '8', 15);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (16, '2022-03-31','CASH', '6', 16);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (17, '2022-03-31','CARD', '4', 17);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (18, '2022-03-31','CASH', '11', 18);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (19, '2022-03-31','CARD', '12', 19);

INSERT INTO PAYMENT (ID , DATE_PAYMENT, PAYMENT_METHOD, COURSEID,
STUDENTID) VALUES (20, '2022-03-31','CASH', '1', 20);
```

Запити бази даних

```
SELECT * FROM STUDENT;
```

```
SELECT * FROM MENTOR;
```

```
SELECT * FROM COURSES;
```

```
SELECT * FROM CLASSROOM;
```

```
SELECT * FROM CLASS;
```

```
SELECT * FROM STUDENT_CLASSES;
```

```
SELECT * FROM PAYMENT;
```

*/*СТАТЬ СТУДЕНТА (ADMIN) */*

```
SELECT STUDENT.ID, FIRSTNAME, LASTNAME, GENDER FROM STUDENT
WHERE GENDER = 'M';
```

*/*СТАТЬ СТУДЕНТА (ADMIN) */*

```
SELECT STUDENT.ID, FIRSTNAME, LASTNAME, GENDER FROM STUDENT
WHERE GENDER = 'F';
```

*/*ВІКОВІ ГРУПИ СТУДЕНТІВ (ADMIN) */*

```
SELECT * FROM STUDENT
WHERE YEAR_GROUP = '0-18';
```

*/*ВІКОВІ ГРУПИ СТУДЕНТІВ (ADMIN) */*

```
SELECT * FROM STUDENT
WHERE YEAR_GROUP = '18-30';
```

*/*ВІКОВІ ГРУПИ СТУДЕНТІВ (ADMIN) */*

```

SELECT * FROM STUDENT

WHERE YEAR_GROUP = '30-40';

/* НАЙСТАРШИЙ СТУДЕНТ (ADMIN) */

SELECT FIRSTNAME, LASTNAME, AGE FROM STUDENT

WHERE AGE = (SELECT MAX(AGE)

FROM STUDENT);

/* НАЙМОЛОДШИЙ СТУДЕНТ (ADMIN) */

SELECT FIRSTNAME, LASTNAME, AGE FROM STUDENT

WHERE AGE = (SELECT MIN(AGE)

FROM STUDENT);

/* ІНФОРМАЦІЯ ПРО СТУДЕНТА ЯКИЙ ОПЛАТИВ КУРС (ADMIN) */

SELECT PAYMENT.ID, STUDENT.FIRSTNAME, STUDENT.LASTNAME,
PAYMENT.DATE_PAYMENT, COURSENAME, PRICE FROM PAYMENT

LEFT JOIN STUDENT ON PAYMENT.STUDENTID = STUDENT.ID

LEFT JOIN COURSES ON PAYMENT.COURSEID = COURSES.ID

WHERE COURSENAME = 'JAVA DEVELOPER';

/* ІНФОРМАЦІЯ ПРО СТУДЕНТА ЯКИЙ ОПЛАТИВ КУРС (ADMIN) */

SELECT PAYMENT.ID, STUDENT.FIRSTNAME, STUDENT.LASTNAME,
PAYMENT.DATE_PAYMENT, COURSENAME, PRICE FROM PAYMENT

LEFT JOIN STUDENT ON PAYMENT.STUDENTID = STUDENT.ID

LEFT JOIN COURSES ON PAYMENT.COURSEID = COURSES.ID

WHERE COURSENAME = 'C# DEVELOPER ADVANCED';

```

```
/* РАНГ МЕНТОРА (USER) */
```

```
SELECT FIRSTNAME, LASTNAME, RANKS FROM MENTOR
WHERE RANKS = 'Senior';
```

```
/* РАНГ МЕНТОРА (USER) */
```

```
SELECT FIRSTNAME, LASTNAME, RANKS FROM MENTOR
WHERE RANKS = 'Middle';
```

```
/* МОВА НАВЧАННЯ МЕНТОРА (USER) */
```

```
SELECT ID, LASTNAME, LANGUAGE_STUDY FROM MENTOR
WHERE LANGUAGE_STUDY LIKE 'Ukrainian';
```

```
/* МОВА НАВЧАННЯ МЕНТОРА (USER) */
```

```
SELECT ID, LASTNAME, LANGUAGE_STUDY FROM MENTOR
WHERE LANGUAGE_STUDY LIKE 'English and Ukrainian ';
```

```
/* МЕНТОРИ ЯКІ ВИКЛАДАЮТЬ БІЛЬШЕ ОДНОГО КУРСУ (USER) */
```

```
SELECT MENTOR.ID, FIRSTNAME, LASTNAME FROM MENTOR
LEFT JOIN CLASS ON CLASS.MENTORID = MENTOR.ID
GROUP BY MENTOR.ID HAVING COUNT(COURSEID) > 1;
```

```
/* ІНФОРМАЦІЯ ПРО МЕНТОРІВ (USER) */
```

```
SELECT CLASS.ID, MENTOR.FIRSTNAME, MENTOR.LASTNAME, COURSENAME,
LANGUAGE_STUDY, SALARY FROM CLASS
LEFT JOIN MENTOR on CLASS.MENTORID = MENTOR.ID
LEFT JOIN COURSES ON CLASS.COURSEID = COURSES.ID;
```

```
/*ПОВНА ІНФОРМАЦІЯ ПРО ЗАНЯТТЯ (USER) */

SELECT CLASS.ID, MENTOR.FIRSTNAME, MENTOR.LASTNAME, COURSENAME,
NUMBER_CLASSROOM, CAPASITY, ADDRESS, CITY, PERIOD, TIME_CLASS,
LANGUAGE_STUDY FROM CLASS

LEFT JOIN MENTOR on CLASS.MENTORID = MENTOR.ID

LEFT JOIN COURSES ON CLASS.COURSEID = COURSES.ID

LEFT JOIN CLASSROOM ON CLASS.CLASSROOMID = CLASSROOM.ID ;
```

```
/* НАЙДОРОЖЧИЙ КУРС (USER) */
```

```
SELECT COURSENAME, PRICE FROM COURSES

WHERE PRICE = (SELECT MAX(PRICE)

FROM COURSES);
```

```
/* НАЙДЕШЕВШИЙ КУРС (USER) */
```

```
SELECT COURSENAME, PRICE FROM COURSES

WHERE PRICE = (SELECT MIN(PRICE)

FROM COURSES);
```

```
/* КУРСИ В ЯКИХ ЦІНА МЕНША ЗА 10000 (USER) */
```

```
SELECT COURSENAME, CATEGORY, PRICE FROM COURSES

WHERE PRICE < 10000;
```

```
/* СТУДЕНТИ ЯКІ ОБРАЛИ СПОСІБ ОПЛАТИ ГОТИВКА (USER) (ADMIN) */
```

```
SELECT PAYMENT.ID, STUDENT.FIRSTNAME, STUDENT.LASTNAME,
PAYMENT.PAYMENT_METHOD, DATE_PAYMENT FROM PAYMENT

LEFT JOIN STUDENT ON PAYMENT.STUDENTID = STUDENT.ID

WHERE PAYMENT_METHOD = 'CASH';
```

```
/* СТУДЕНТИ ЯКІ ОБРАЛИ СПОСІБ ОПЛАТИ КАРТКА (USER) (ADMIN) */  
SELECT PAYMENT.ID, STUDENT.FIRSTNAME, STUDENT.LASTNAME,  
PAYMENT.PAYMENT_METHOD, DATE_PAYMENT FROM PAYMENT  
LEFT JOIN STUDENT ON PAYMENT.STUDENTID = STUDENT.ID  
WHERE PAYMENT_METHOD = 'CARD';
```