**Лабораторна робота №1 HTTP, MVC»**

Мета:Навчитись створювати прості веб додатки. Розібратись з протоколом HTTP. Освоїти шаблон MVC.

Завдання:

1. Створити проект NetCore MVC

2. Створити новий контолер, відмінний від HomeController

3. Створити кілька Action-ів в контролері

4. Створити для них VIEW

5. Зв’язати в’юхи між собою посиланнями

6. В одній з в’юх відобразити інформацію передану через Get-параметр

7. Створити меню в Layout.cshtml файлі.

8. Виводити в лаяут текстом назву поточної в’юхи (title)

Перелік тем для контрольних запитань на опрацювання:

1. Що таке шаблон MVC?

2. Протокол HTTP, заголовки, версії протоколу, переваги, недоліки, детали роботи запиту та відповіді.

3. Маршрутизація (Routing) в ASP MVC

4. Layout файли

**Лабораторна робота №2 «Повторення ключових механізмів»**

Мета: Повторити основні аспекти основ програмування, системного програмування, та об'єктно орієнтованого програмування.

Завдання:

* Створити сутності відповідно до варіанту. Створити окрему збірку в якій розмістити класи для серіалізації сутностей використати бінарну серіалізацію та серіалізацію за контрактом. Створити контролер, реалізувати методи, що при зверненні користувача повернуть бінарно серіалізовані дані та серіалізовані за контрактом відповідно, в вигляді файлу.

3.1

* Реалізувати логіку завантаження даних з файлу в момент запуску додатку.
* Створити клас в якій помістити матрицю розмірністю 10000 на 10000, створити клас, що виконувати обрахунок суми всіх елементів матриці. Розпаралелити процес обрахунку.

Перелік тем для контрольних запитань на опрацювання:

1. Знати та вміти пояснити різницю між Task і Thread.

2. Пояснити як влаштована робота з файлами та механізми серіалізації.

3. Знати та вміти пояснити основні ООП.

**Лабораторна робота №3 «Memory storage»**

Мета: Розібратись з механізмом Dependency Injection. Освоїти розмежування контролерів та класів даних.

Завдання:

* Реалізувати репозиторії які будуть зберігати дані в пам’яті(в якості джерела даних використати статичний клас контейнер). Реалізувати CRUD (Create, Read, Update, Delete) операції відповідно до варіанту для кожної сутності.
* Створити кілька Transient і Scoped сервісів. Наприклад Transient сервіс може здійснювати якісь обчислення.
* Загально доступні значення змінних, налаштування необхідні для роботи сервісу помістити в файл конфігурації.
* Реалізувати користувацький інтерфейс, що дозволяє виконувати весь круд операцій з сутностями.

Перелік тем для контрольних запитань на опрацювання:

1. Розібрати структуру шаблону Repository. Вміти відобразити діаграму класів шаблону.

2. Пояснити як влаштований DI. Яким чином відбувається реєстрація залежностей. Яким чином можна задати інший контейнер.Назвати та пояснити наявні життєві цикли. Способи завантаження конфігурацій з файлів конфігурації?

3. Аргументувати вибір способу передачі даних на представлення.

**Лабораторна робота №4 « EF CRUD»**

Мета: Розібратись з роботою з базами даних в Asp.net. Розібратись з шаблоном Repository при роботі з базами даних. Навчитись створювати міграції.

Завдання:

* Реалізувати репозиторії які будуть зберігати дані в базі даних. Реалізувати CRUD (Create, Read, Update, Delete) операції відповідно до варіанту.
* Створити кілька Transient і Scoped сервісів. Наприклад Transient сервіс може здійснювати якісь обчислення.
* Створити DTOs (Data transfer objects) для відображення моделі на користувацькому інтерфейсі.

* Строчка з'єднання з базою даних повинна міститись в файлі конфігурації.
* Реалізувати користувацький інтерфейс, що дозволяє виконувати весь круд операцій з сутностями.

Перелік тем для контрольних запитань на опрацювання:

1. Розібрати структуру шаблону Repository. Вміти відобразити діаграму класів шаблону. (В випадку використання асинхронних операцій пояснити як влаштовані асинхронні операції в .net core.)

2. Пояснити призначення Data transfer objects.

3. Роботу з конфігураціями.

4. Пояснити поняття ORM, спосіб обробки даних та їх збереження, що таке DbContext. Різницю між IEnumerable і IQueryable. Вміти пояснити та вказати в коді місця приведення одного інтерфейсу до іншого. Пояснити різницю між DbSet<TEntity>.Set<TEntity>, DbSet<TEntity>.Entry<TEntity>.Пояснити чому DbContext імплементує IDisposable.

**Лабораторна робота №5 «Web Api»**

Мета: Розібратись з REST-API. Знати що таке REST. Навчитись створювати API з яким будуть спілкуватись інші додатки.

Завдання:

* Створити окремий web-проект в тому самому рішенні, що включало в себе минулі лаб. роботи, вказавши в якості типу проекту REST API.
* Створити по окремому контролеру для кожної сутності, вказаної в варіанті. В кожному контролері реалізувати CRUD (Create, Read, Update, Delete) операцій для сутностей використавши DTO та притримуючись концепцій побудови RESTFUL API.
* При реалізації завдання керуватись принципами ООП та S.O.L.I.D., та уникати дублювання коду.
* Викликати кожну операцію, в кожному контролері.
* Для тестування використати Rest client (Postman).

Перелік тем для контрольних запитань на опрацювання:

1. Вміти пояснити механізм обробки запиту, розуміти яким чином використовується серіалізація та вміти дати визначення поняттю серіалізація. (В випадку використання асинхронних операцій пояснити як влаштовані асинхронні операції в .net core.)

2. Пояснити ключові аспекти побудови коректного RESTFUL API, сферу застосування.

**Лабораторна робота №6 «Авторизація у web-застосунках»**

Мета: Розібратись як створювати авторизацію в web-застосунках.

Основне завдання (60% від балу за лаб. роботу):

* Додати форми реєстрації та авторизації на рівень представлення.
* Додати форми по редагування акаунту, за необхідності форми для поведінки авторизованих користувачів.
* Інтегрувати механізм identity authentication на рівні роботи з базою даних - використати identity dbcontext.
* Реалізувати логіку логіну, реєстрації та виходу з акаунту, логіку для поведінки авторизованих користувачів відповідно до варіанту.

Додаткове завдання (+20% від балу за лаб. роботу)

* Фейкове підтвердження пошти, код підтвердження виводиться на консоль в сервері, потім його вводять в браузері.

Додаткове завдання (+40% від балу за лаб. роботу)

* Аналогічно реалізувати логіку авторизації на основі JWT для RESTFUL API.
* Підтвердження пошти через рельний поштовий клієнт
* Зміна пароля
* Додаткові поля при реєстрації

Перелік тем для контрольних запитань на опрацювання:

1. Вміти пояснити поняття та різницю між JWT та Claims аутентифікацією, та механізм роботи identity. Розуміти різницю між DbContext IdentityDbContext. .

Варіанти завдань на лабораторні роботи

* Бібліотека
* Сутності : (читач, книги, розділи, автор, книга автора, картка читача)
* картка читача посилання на книгу и читача дата отримання дата здачі
* книга автора має посилання на автора і на книгу
* Кожна книга містить список розділів

**Логічна поведінка:**

* Пошук книги по автору, по назві, по назві розділу.
* Пошук авторів по імені, даті народження, кількості книг.
* Заповнення картки читача, видача книги.
* Товарообіг
* Сутності : (товар, склад, товар на складі, магазин, товар в магазині, заявка на поставку, елемент поставки)
* Товар на складі містить посилання на товар та на склад
* товар в магазині містить посилання на магазин і на товар
* елемент поставки містять посилання на товар та на заявку на поставку
* заявка на поставку містить список товарів та посилання на магазин чи склад замовник

**Логічна поведінка:**

* Пошук товару по по назві, по номеру заявки, по окремо взятому складі чи магазині.
* Пошук заявок по назві, даті, кількості елементів.
* Заповнення заявки на поставку.
* Карта
* Сутності : (країна місто, вулиця, будинок квартира, жителі, жителі квартири)
* Країна містить список міст , місто містить список вулиць, вулиця має список будинків, будинок містить список квартир
* Будинок має координати і посилання на вулицю
* жителі квартири містять посилання на жителя та на квартиру

**Логічна поведінка:**

* Пошук міста по по назві, по кількості жителів, по назві вулиці врахувати перевірку номеру будинку.
* Пошук жителя по ФІО, даті народження, кількості квартир.
* Внесення інформації про нового мешканця до квартири, за умови, що площа на одну особу не менше ніж 9 кв.м.
* Парковка
* Сутності: (клієнт, автомобілі клієнта, автомобіль, парковка, парковочне місце, парковочний талон)
* Парковочний талон – дата і час, коли людина залишила автомобіль, і коли забрала його, також посилання на парковочне місце
* Клієнт може мати кілька автомобілів (автомобілі клієнта містять посилання на автомобіль і на клієнта)
* Автомобіль має список талонів
* Талон має посилання на автомобіль

**Логічна поведінка:**

* Пошук клієнта по по назві авто, по номеру талону на паркування,по ФІО,
* Пошук найблищої парковки з вільним місцем чи місцем що звільниться за вказаний проміжок. Пошук по номеру білету на паркування, назві автомобіля та даті парковки.
* Створення паркувального талону.
* КПП офісного центру
* Сутності: фірма, співробітник, офіс, відношення співробітника до офісу
* Фірма може орендувати кілька офісів
* Співробітник має доступ до всіх або до частини офісів цієї фірми (реалізується через таблицю відношення співробітника до офісу яка містить посилання на офіс та співробітника)
* Фірма має список офісів та співробітників

**Логічна поведінка:**

* Пошук співробітника по ФІО, даті народження, назві відділу та даті прийняття на роботу
* Пошук фірми по даті створення, по назві департаменту, по кількості співробітників.
* Прийняття співробітника на роботу, відкриття нового відділу, перевірка належності співробітника до даної фірми.
* Рухоме і нерухоме майно
* Класи: Людина, право власності, автомобіль, будинок
* кожна людина може мати одну, або кілька автівок
* аналогічно з будинками
* Відповідно в класі Людина має бути список будинків і список автомобілів

**Логічна поведінка:**

* Пошук власності по ФІО, даті народження, людини,по даті створення права власності.
* Пошук людей яким належить автомобіль по назві, чи будинок по назві вулиці чи частині адреси.
* Реалізація створення та подальшої обробки права власності.
* Банківську операції
* Класи: Людина, рахунок, транзакція, акціонери компанії, компанія
* Кожна людина має певну кількість рахунків
* Людина може виступати акціонером у багатьох компаній (акціонери компанії містять посилання на компанію та людину)
* Рахунок зберігає в собі баланс, та список транзакцій
* Транзакція містить:
* Посилання на рахунок, з якого здійснюють переказ
* Посилання на раунок, куди здійснюють переказ

Дату+час (DateTimeOffset) операції.

**Логічна поведінка:**

* Пошук людини по ФІО, даті народження, номеру рахунку, номеру транзакції, назві компанії.
* Пошук акціонерів компанії по даті створення, по назві компанії.
* Створення та подальша обробка транзакції, створення рахунку, компанії, її акціонерів.
* Магазин
* Сутності: Товар, покупець, характеристики, характеристики товару, замовлення
* Товар – найменування, ціна,…
* Покупець може здійснити кілька походів в магазин, отже матиме не один чек
* замовлення містить список покупок даного користувача, дату покупки
* характеристики товару містять посилання на товар і на характеристику

**Логічна поведінка:**

* Пошук людини(людей) по ФІО, даті народження, номеру замовлення, покупці чи списку покупок.
* Пошук товарів по вказаним характеристикам.
* Створення та подальша обробка замовлення.
* Футбольний менеджер
* Сутності: Команда, Гравець, Зіграний матч, Чемпіонат, Команди чемпіонату.
* Зіграний матч містить:
* посилання на дві команди,
* рахунок
* дату проведення
* Команда містить
* список гравців
* Команди чемпіонату

1. містить посилання на чемпіонат та команду

**Логічна поведінка:**

* Пошук команди(команд) по назві, даті створення, назві та році проведення чемпіонату .
* Пошук матчу(матчів) по назві команд, по рахунку, по назві чемпіонату.
* Створення чемпіонатів, проведення матчів, створення трансфер гравців.
* Ремонт комп’ютерів
* Сутності: Власник, комп’ютер, комплектуючі комп’ютера, комплектуючі, замовлення
* Власник містить список комп’ютерів
* Замовлення містить список комплектуючих для заміни, дату
* замовлення містить посилання на комплектуючі на користувача
* комплектуючі комп’ютера
* Зберігає посилання на комп’ютер та комплектуючі

**Логічна поведінка:**

* Пошук комплектуючої(комплектуючих) по назві, даті виготовлення, номеру замовлення моделі комп’ютера.
* Пошук користувача за ФІО, назвами придбаних запчастин, номеру замовлення, даті замовлення, моделі комп’ютера.
* Створення заявки на ремонт, що містить встановлений внаслідок перевірки список не працюючих деталей, можливість чи не можливість проведення ремонту (якщо немає запчастин ремонт не можливий, оціночну вартість ремонту).
* ІТ-компанія
* Сутності: Відділ, співробітник відділу, співробітник, задачі
* співробітник відділу містить посилання на працівника і відділ
* В кожного співробітника є список задач

**Логічна поведінка:**

* Пошук співробітника(співробітників) по ФІО, даті народження, задачам, назві відділу.
* Пошук відділу(відділів) по назві, по інформації про співробітника, по задачам.
* Створення співробітників, відділів, задач і відслідковування їх виконання,
* Комп'ютерна мережа
* Сутності: маршрутизатор, комп'ютер, корпус, кімната, обладнання в кімнаті
* Корпус містить список кімнат
* обладнання в кімнаті містить посилання на кімнату та на одиницю обладнання
* Кожен комп’ютер під'єднаний до одного маршрутизатору
* Маршрутизатор містить список комп’ютерів

**Логічна поведінка:**

* Пошук пристрою(пристроїв) по логічній, фізичній адресі, номеру кімнати, назві корпусу, даті виготовлення, моделі. (пристроєм вважається як ПК так і маршрутизатор)
* Створення та графічне моделювання локальної мережі, що включає в себе перевірку правильності задання діапазону ip адрес.
* Тех підтриимка
* Сутності: (спеціаліст служби підтримки, користувач, запит в службу підтримки, повідомлення)
* Запит в службу підтримки містить посилання на користувача спеціаліста служби підтримки текст запиту і статус вирішений чи ні, і дата вирішення
* Під час діалогу між користувачем і спеціалістом, в запис додаються повідомлення

**Логічна поведінка:**

* Пошук запиту(запитів) по даті, тегам, часовому проміжку, статусу запиту, по не прочитаним повідомленням, по імені спеціаліста.
* Відображення спеціаліста(спеціалістів) по значенню опрацьованих заявок що більше за середнє значення кількості заявок на спеціаліста, по відсутності активних запитів, по повідомленням користувачів чи запитам, що розміщені раніше ніж вказана дата.
* Процес подачі заявки, знаходження найменш завантаженого спеціаліста, після утвердження спеціаліста побудова повідомлення спеціаліста, в випадку відсутності повідомлень користувача та прийнятого ним рішення що заявка не актуальна зміна її статусу.
* Блог
* Сутності: (користувач, блог користувача , пости користувача до блогів)
* Блог користувача містить посилання на користувача та конкретний блог, користувач містить список блогів
* пости користувача до блогу містить посилання на користувача і на блог в якій він розміщений

**Логічна поведінка:**

* Пошук посту(постів) користувача по контенту(частковому або повному входженні), по назві посту.
* Пошук блогу(блогів) за часовим періодом найбільшої активності (користувач вказує період в вигляді двух дат після чого отримує список блогів з відображенням кількості постів зроблених за вказаний час та користувачів,що створювали пости) .
* Процес створення ведення та перегляду сторонніх блогів.
* Виробництво
* Сутності: (Компонентна база, шаблон виробництва для деталі, виробничі процеси, категорії виробничих процесів)
* Компонентна база деталей (список деталей деталь описати власноруч)
* Шаблон виробництва для деталі містить посилання на компонентну базу як вхідний елемент та на компонентну базу як вихідний компонент і посилання на виробничий процес
* Виробничі процеси містить назву та посилання на компонентну базу як результат процесу посилання на категорію виробничих процесів
* Категорії виробничих процесів назва, тип … , список елементів виробничого процесу

**Логічна поведінка:**

* Пошук деталі(деталей) по назві, даті виготовлення, відповідно до обраного шаблону шаблонів виробництва(результат повинен бути відображений в вигляді атомарних запчастин - - запчастин які не містять складових компонентів).
* Пошук шаблону(шаблонів) виробництва по вказаній компонентній базі.
* Процес побудови запчастин по шаблону виробництва за вказаною користувачем компонентою базою.
* Мессенджер
* Сутності: (користувач, чат, користувачі в чаті, повідомлення в чаті, повідомлення, пристрої користувача)
* користувачі в чаті містить посилання на чат та посилання на користувача
* Повідомлення в чаті містять посилання на чат і посилання на повідомлення
* Повідомлення містить посилання на користувача (автора)
* Користувач містить список пристроїв
* пристрої користувача IP, час входу

**Логічна поведінка:**

* Перегляд активності за вказаний проміжок часу, вивід пристрій ip адреса, кількість відправлених повідомлень.
* Пошук користувача(користувачів) за номером телефону, за ip адресою, електронною поштою, ФІО.
* Логіка створення чату передача повідомлень(додатково передача файлів, групові чати).
* Медіа програвач
* Класи: (виконавець, виконавці альбому, альбом, трек, виконавці треку)
* виконавці альбому містить посилання на виконавця та на альбом
* альбом містить список треків

**Логічна поведінка:**

* Пошук альбому по даті виходу, виконавцю(виконавцям), треку чи списку треків.
* Логіка збереження треків та альбомів на сервері, відтворення обраних музикальних композицій.