**Комунальний заклад освіти**

**«Дніпровський науковий ліцей інформаційних технологій»**

**ВИПУСКНА РОБОТА**

**На тему: Створення сайту**

**вантажно-логістичних перевезень**

**Виконав:**

Ліцеїст11-В-1 класу

Пригара Кирило Юрійович

**Керівник:**

Лавренюк І. В. \_\_\_\_\_\_\_\_

Дніпро

2024

ЗМІСТ

[ВСТУП 3](#_Toc182249522)

[Тема роботи: 3](#_Toc182249523)

[Актуальність роботи: 3](#_Toc182249524)

[Мета роботи: 3](#_Toc182249525)

[Основні завдання роботи 3](#_Toc182249526)

[Основна частина 4](#_Toc182249527)

[Теоретична частина з Інформатики: 4](#_Toc182249528)

[Структура БД: 4](#_Toc182249529)

[Основні додатки проєкту: 6](#_Toc182249530)

[Опис роботи 9](#_Toc182249531)

[Посібник користувача 9](#_Toc182249532)

[Структурна схема програми 10](#_Toc182249533)

[Напрямки де можна використати роботу 14](#_Toc182249534)

[Апаратні вимоги 14](#_Toc182249535)

[Комплектація програми 15](#_Toc182249536)

[Використані програмні засоби 15](#_Toc182249537)

[Висновки 16](#_Toc182249538)

[Список Використаних джерел 17](#_Toc182249539)

[Додатки 17](#_Toc182249540)

# ВСТУП

## Тема роботи:

"Створення сайту вантажно-логістичних перевезень".

## Актуальність роботи:

Вантажні перевезення є основою для розвитку бізнесу у всіх сферах економіки. Зі збільшенням обсягів міжнародної торгівлі та внутрішньої логістики, компанії стикаються з необхідністю автоматизації своїх процесів для зменшення витрат, покращення швидкості доставки та підвищення якості обслуговування. Сайт для управління логістичними операціями вирішує низку актуальних проблем, таких як складність відстеження вантажів, довгі процеси обробки замовлень та неефективне управління ресурсами. Усе це сприяє необхідності створення комплексних вебрішень, що надають можливості для автоматизації логістичних процесів та підвищення зручності для клієнтів.

Сучасний розвиток інформаційних технологій надає унікальні можливості для створення інноваційних рішень у сфері логістики. Створення спеціалізованого сайту допоможе не лише оптимізувати роботу логістичних компаній, а й значно підвищити їхню конкурентоспроможність на ринку, що робить цю тему надзвичайно актуальною у сьогоднішніх умовах.

Цей проект буде корисним для логістичних компаній, які прагнуть оптимізувати свою діяльність, а також для розробників, що спеціалізуються на створенні вебрішень для бізнесу.

## Мета роботи:

Створення функціонального сайту, який допоможе логістичним компаніям полегшити управління вантажно-логістичними перевезеннями.

Основні завдання роботи:

* Проведення аналізу існуючих рішень у сфері логістичних вебсайтів і вибір найкращих підходів для створення системи.
* Розробка зручного інтерфейсу для клієнтів і співробітників логістичних компаній.
* Створення сторінок для реестрації та логування.
* Створення функціоналу додавання вантажу та транспорту для перевезення.
* Імпортувати таблицю з Країнами, Регіонами, Містами(близько 3млн м.)
* Створення фільтру для пошуку вантажу, транспорту, та відображення на сайті.
* Створення функціоналу взяття вантажу, заморожування вантажу перевізником, підтверження взяття вантажу з детальною інформацією про транспорт та перевізника.
* Створення документів(актів, рахунок фактури) на основі об`єкту перевезення.
* Створення відгуків, скарг, рейтингу користувачів.

# Основна частина

## Теоретична частина з Інформатики:

### Изображение выглядит как текст, снимок экрана, Шрифт, документСтруктура БД:

* auth:
  + Це додаток, що відповідає за автентифікацію та управління правами користувачів.
  + Моделі:
    - auth\_group: групи користувачів.
    - auth\_group\_permissions: права груп.
    - auth\_permission: індивідуальні права доступу для користувачів.
* captcha:
  + Модуль для роботи з CAPTCHA.
  + Модель:
    - captcha\_captchastore: зберігання даних CAPTCHA.
* cargo:
  + Додаток для роботи з вантажем.
  + Моделі:
    - cargo\_bodytype: типи кузова для вантажу.
    - cargo\_cargo: інформація про вантаж.
    - cargo\_cargo\_body\_type: зв'язок між вантажем і типами кузова.
    - cargo\_cargo\_load\_unload: завантаження/розвантаження вантажу.
    - cargo\_permissions: права для взаємодії з вантажем.
* django (вбудовані додатки Django):
  + Моделі:
    - django\_admin\_log: журнал подій адміністратора.
    - django\_content\_type: типи контенту.
    - django\_migrations: міграції бази даних.
    - django\_session: сесії користувачів.
* main:
  + Основний додаток сайту.
  + Моделі:
    - main\_event: події.
    - main\_frozen: модель для заморожених записів які чекають на підтвердження.
* transport:
  + Додаток для управління транспортом.
  + Моделі:
    - transport\_bodytype: типи кузовів транспорту.
    - transport\_loadunload: завантаження/розвантаження транспорту.
    - transport\_transport: інформація про транспортні засоби.
    - transport\_transport\_body\_type: зв'язок між транспортними засобами і типами кузовів.
    - transport\_transport\_load\_unload: зв'язок між транспортом і завантаженням/розвантаженням.
    - transport\_transport\_permissions: права доступу до транспортних засобів.
* users:
  + Додаток для управління користувачами.
  + Моделі:
    - users\_activitydirection: напрямки діяльності користувачів.
    - users\_complaint: скарги користувачів.
    - users\_review: відгуки користувачів.
    - users\_user: інформація про користувачів.
    - users\_user\_activity\_directions: зв'язок між користувачами і напрямками їхньої діяльності.
    - users\_user\_groups: зв'язок між користувачами і групами.
    - users\_user\_user\_permissions: права користувачів.

### Основні додатки проєкту:

1. cargo:

* Цей додаток відповідає за роботу з вантажем.
* У папці є файли:
  + admin.py: конфігурація для адміністративної панелі Django, де можна керувати моделями додатку через інтерфейс адміністратора.
  + apps.py: налаштування самого додатку, де визначається конфігурація додатку (наприклад, ім'я).
  + forms.py: визначення форм, які будуть використовуватись для обробки введення даних користувачами.
  + models.py: визначення моделей, тобто структури даних і логіки бази даних.
  + tests.py: тести для додатку, які дозволяють перевірити правильність роботи.
  + urls.py: маршрутизація URL для додатку cargo, тобто визначає шляхи до різних сторінок цього модуля.
  + utils.py: утиліти та допоміжні функції для додатку.
  + views.py: представлення, що відповідає за логіку відображення даних на сторінках (контролери).

1. main:

* Основний додаток, відповідає за головну сторінку і основну логіку сайту.
* Структура та сама як і в cargo.

1. media:

* Папка, де, зберігаються завантажені медіа-файли, такі як зображення, документи, відео тощо.

1. transport:

* Цей додаток відповідає за управління транспортом. Структура аналогічна додатку cargo:
* admin.py, apps.py, forms.py, models.py, tests.py, urls.py, views.py: ці файли виконують аналогічні функції, що й у додатку cargo, але з фокусом на роботу з транспортом.

1. users:

* Додаток для управління користувачами:
  + admin.py: конфігурація адміністративної панелі для керування користувачами.
  + apps.py: конфігурація додатку користувачів.
  + forms.py: форми для реєстрації, авторизації та інших взаємодій користувачів.
  + models.py: моделі для зберігання інформації про користувачів.
  + tests.py: тести для перевірки функцій користувачів.
  + urls.py: маршрутизація URL для додатку користувачів.
  + views.py: логіка відображення і обробки запитів користувачів.

1. TransHub:

* Це основна конфігураційна папка проєкту. Файл urls.py всередині цієї папки містить загальну маршрутизацію для всього сайту.

urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('users/', include('users.urls', namespace='users')),  
 path('cargo/', include('cargo.urls', namespace='cargo')),  
 path('transport/', include('transport.urls', namespace='transport')),  
 path('', include('main.urls')),  
]

path('admin/', admin.site.urls): Це шлях до адміністративної панелі Django, де адміністратор може керувати моделями всіх додатків через інтерфейс адміністратора.

path('users/', include('users.urls', namespace='users')): Цей шлях вказує на те, що всі URL, які починаються з /users/, будуть оброблятись додатком users через його файл urls.py. Простір імен (namespace) використовується для унікалізації маршрутів.

path('cargo/', include('cargo.urls', namespace='cargo')): Всі URL, які починаються з /cargo/, будуть оброблятись додатком cargo.

path('transport/', include('transport.urls', namespace='transport')): Всі URL, що починаються з /transport/, обробляються додатком transport.

path('', include('main.urls')): Це шлях для кореневого URL (головної сторінки). Він вказує на те, що всі запити до головної сторінки сайту будуть оброблятись додатком main.

**Загальна структура проєкту:**

Проєкт має чітко структуровану архітектуру, де кожен додаток відповідає за окрему функціональність: користувачі, вантажі, транспорт та головна сторінка. Основна конфігурація проєкту знаходиться в папці TransHub, яка включає маршрутизацію та адміністративну панель.

**Як працює маршрутизація:**

Запити до адміністративної панелі обробляються через /admin/.

Запити, пов'язані з користувачами, обробляються через /users/ і далі передаються додатку users.

Запити, що стосуються вантажів, обробляються через /cargo/ додатком cargo.

Запити, пов'язані з транспортними засобами, проходять через /transport/.

Кореневі запити (/) спрямовуються до додатку main, який, івідповідає за відображення головної сторінки сайту.

## Опис роботи

Проєкт «Сайт вантажно-логістичних перевезень» створено для автоматизації ключових процесів логістичних компаній. Сайт покликаний допомогти логістичним компаніям швидше обробляти замовлення, ефективніше керувати ресурсами, а також покращити зручність взаємодії з клієнтами. Основна функціональність сайту включає:

1. **Управління замовленнями**: Клієнти можуть створювати нові замовлення, переглядати статус перевезення, а також отримувати повідомлення про етапи виконання (наприклад, «вантаж прийнято», «вантаж в дорозі», «вантаж доставлено»).
2. **Інтеграція відгуків і скарг**: Клієнти можуть залишати відгуки про надані послуги та подавати скарги, що дозволяє компаніям покращувати якість обслуговування на основі зворотного зв'язку.
3. **Управління транспортними засобами**: Логістичні компанії можуть додавати, оновлювати інформацію про транспортні засоби (наприклад, тип кузова, вантажопідйомність) і розподіляти їх для виконання певних замовлень.
4. **Система прав доступу**: Реалізовано можливість управління правами доступу для різних категорій користувачів: клієнти, водії, адміністратори. Це дозволяє забезпечити безпеку даних і контроль за доступом до важливої інформації.

## Посібник користувача

1. Реєстрація та авторизація: Користувачі можуть створити обліковий запис, пройти процедуру авторизації з підтвердженням CAPTCHA.
2. Функції для клієнтів: Користувачі можуть замовляти логістичні послуги, переглядати статус замовлень, залишати відгуки та подавати скарги.
3. Адміністрування: Адміністратори мають доступ до панелі керування для обробки замовлень, управління транспортними засобами та керування правами користувачів.
4. Перегляд подій: Користувачі можуть слідкувати за статусом вантажу завдяки повідомленням про події, пов’язані із завантаженням та розвантаженням.

## Структурна схема програми

**1. Моделі (Models)**

**1.1. Модель Country**

Модель, що зберігає інформацію про країни.

* **id**: Унікальний ідентифікатор (Primary Key).
* **name**: Назва країни (CharField, max\_length=255).
* **code**: Код країни (CharField, max\_length=10).
* **str()**: Повертає назву країни.

**1.2. Модель Region**

Модель, що зберігає інформацію про регіони.

* **id**: Унікальний ідентифікатор (Primary Key).
* **name**: Назва регіону (CharField, max\_length=255).
* **code**: Код регіону (CharField, max\_length=10).
* **country**: Зовнішній ключ на модель Country (ForeignKey), що вказує на країну, до якої належить регіон.
* **str()**: Повертає назву регіону.

**1.3. Модель City**

Модель, що зберігає інформацію про міста.

* **id**: Унікальний ідентифікатор (Primary Key).
* **name**: Назва міста (CharField, max\_length=255).
* **region**: Зовнішній ключ на модель Region (ForeignKey), що вказує на регіон, до якого належить місто.
* **country**: Зовнішній ключ на модель Country (ForeignKey), що вказує на країну, до якої належить місто.
* **latitude**: Широта міста (FloatField).
* **longitude**: Довгота міста (FloatField).
* **str()**: Повертає назву міста.

**1.4. Модель Cargo**

Модель для представлення вантажу.

* **id**: Унікальний ідентифікатор.
* **title**: Назва вантажу (CharField).
* **country\_from**: Країна відправлення (ForeignKey на модель Country).
* **region\_from**: Регіон відправлення (ForeignKey на модель Region).
* **city\_from**: Місто відправлення (ForeignKey на модель City).
* **country\_to**: Країна призначення (ForeignKey на модель Country).
* **region\_to**: Регіон призначення (ForeignKey на модель Region).
* **city\_to**: Місто призначення (ForeignKey на модель City).
* **weight\_max**: Максимальна вага вантажу (PositiveIntegerField).
* **volume\_max**: Максимальний об'єм вантажу (PositiveIntegerField).
* **length, width, height**: Розміри вантажу (DecimalField).
* **packaging**: Тип упаковки (CharField з варіантами).
* **payment\_type**: Тип оплати (CharField з варіантами).
* **payment\_moment**: Момент оплати (CharField з варіантами).
* **amount**: Сума грошей (PositiveIntegerField).
* **currency**: Валюта (CharField).
* **photo**: Фото вантажу (ImageField).
* **notes**: Примітки до вантажу (TextField).
* **published**: Статус публікації (BooleanField).
* **created\_at, updated\_at**: Дати створення і оновлення (DateTimeField).
* **freeze**: Заморожування вантажу (BooleanField).
* **status**: Статус вантажу (CharField з варіантами).
* **author**: Автор вантажу (ForeignKey на користувача).

**1.5. Модель Transport**

Модель для транспорту.

* **id**: Унікальний ідентифікатор.
* **loading\_date**: Дата завантаження (DateField).
* **country\_from, region\_from, city\_from**: Місце відправлення.
* **country\_to, region\_to, city\_to**: Місце призначення.
* **brand**: Бренд транспорту (CharField).
* **model**: Модель транспорту (CharField).
* **license\_plate**: Номерний знак транспорту (CharField).
* **year**: Рік випуску (PositiveIntegerField).
* **max\_weight**: Максимальна вага транспорту (DecimalField).
* **max\_volume**: Максимальний об'єм транспорту (DecimalField).
* **body\_type**: Тип кузова (ManyToManyField на модель BodyType).
* **load\_unload**: Завантаження/розвантаження (ManyToManyField на модель LoadUnload).
* **permissions**: Дозволи на транспорту (ManyToManyField на модель Permissions).
* **photo**: Фото транспорту (ImageField).
* **notes**: Примітки до транспорту (TextField).
* **published**: Статус публікації транспорту (BooleanField).
* **created\_at, updated\_at**: Дати створення і оновлення (DateTimeField).
* **freeze**: Заморожування транспорту (BooleanField).
* **status**: Статус транспорту (CharField з варіантами).
* **author**: Автор транспорту (ForeignKey на користувача).

**1.6. Модель LoadUnload**

Модель для завантаження/розвантаження.

* **id**: Унікальний ідентифікатор.
* **name**: Назва типу завантаження/розвантаження (CharField).
* **str()**: Повертає назву завантаження/розвантаження.

**1.7. Модель Permissions**

Модель для дозволів на транспорту.

* **id**: Унікальний ідентифікатор.
* **name**: Назва дозволу (CharField).
* **str()**: Повертає назву дозволу.

**1.8. Модель BodyType**

Модель для типів кузова.

* **id**: Унікальний ідентифікатор.
* **slug**: Унікальний slug (CharField), генерується автоматично на основі назви.
* **name**: Назва типу кузова (CharField).
* **save()**: Перевизначений метод для автоматичного створення slug.
* **str()**: Повертає назву типу кузова.

**1.9. Модель Event**

Модель для подій, що пов'язані з вантажем та транспортом.

* **cargo**: Зовнішній ключ на вантаж (ForeignKey).
* **transport**: Зовнішній ключ на транспорт (ForeignKey).
* **event\_date**: Дата події (DateField).
* **arrived**: Статус прибуття (BooleanField).
* **act\_file**: Файл акту (FileField).
* **invoice\_file**: Файл рахунку (FileField).

**1.10. Модель Frozen**

Модель для "заморожування" вантажу.

* **cargo**: Зовнішній ключ на вантаж (ForeignKey).
* **transport**: Зовнішній ключ на транспорт (ForeignKey).
* **frozen\_at**: Дата і час заморожування (DateTimeField).
* **str()**: Повертає строку з ідентифікаторами вантажу та транспорту.

Можливості по створенню актів і рахунків

1. Акти:

* Створюються автоматично або вручну при завершенні доставки (коли поле arrived встановлюється як True).
* Зберігаються в act\_file та доступні в особистому кабінеті клієнта для підтвердження передачі вантажу.

1. Рахунки-фактури:

* Можуть бути створені при відправленні вантажу або після завершення перевезення.
* Зберігаються у полі invoice\_file, доступному для завантаження. Рахунки містять інформацію про вартість послуг та реквізити для оплати.

Зв'язки між моделями:

* **ForeignKey**: Зовнішні ключі використовуються для створення зв'язків між моделями. Наприклад, вантаж може належати певній країні, регіону та місту. Подібні зв'язки визначені для транспорту і дозволяють відслідковувати зв'язки між об'єктами в базі.
* **ManyToManyField**: Це зв'язки для полів, де один об'єкт може бути пов'язаний з кількома іншими об'єктами. Наприклад, вантаж може мати кілька типів кузова або дозволів, що відображаються через ці зв'язки.

Моделі для обробки відстаней та часу:

* Моделі Cargo і Transport мають методи для обчислення відстані між двома містами, використовуючи формулу Haversine.
* Обидві моделі також мають методи для розрахунку часу, який пройшов з останнього оновлення об'єкта.

## Напрямки де можна використати роботу

Проєкт можна використовувати в логістичних компаніях, службах доставки та великих підприємствах, що потребують автоматизації вантажних перевезень. Система підходить для компаній, що працюють з великими обсягами перевезень, завдяки зручному управлінню транспортом і оптимізації логістичних процесів.

## Апаратні вимоги

Для успішного розгортання проєкту рекомендовані такі вимоги:

1. Сервер:

* **Процесор**: 2 ядра з частотою 2,5 ГГц або вище (наприклад, Intel Xeon або AMD Ryzen).
* **Оперативна пам'ять**: не менше 4 ГБ, для великих навантажень рекомендовано 8 ГБ.
* **Сховище**: 20 ГБ або більше для зберігання бази даних і медіафайлів (залежно від обсягу завантажуваних файлів).
* **Операційна система**: Ubuntu, CentOS, або інша ОС Linux з підтримкою Python 3.8+.

1. Клієнтські пристрої:

* Веббраузери: Chrome, Firefox, Edge (останні версії).
* Доступ з мобільних пристроїв також підтримується завдяки адаптивній верстці сайту.

## Комплектація програми

1. Кодова база:
   * Основний код написаний на Python 3.8+ з використанням фреймворку Django.
   * Компоненти фронтенду створені з використанням HTML, CSS, JavaScript, а також бібліотеки Bootstrap для адаптивності.
2. База даних:
   * SQL база даних (може бути використана SQLite на етапі тестування та PostgreSQL для продуктивного середовища).
   * Моделі зберігають інформацію про користувачів, вантажі, транспорт, а також події для відстеження змін статусів.
3. Медіа-файли:
   * Зображення, відео та документи, завантажені користувачами, зберігаються в папці media.

## Використані програмні засоби

1. Django: основний фреймворк для побудови серверної логіки.
2. Bootstrap: для створення адаптивного інтерфейсу, який зручно відображається на всіх пристроях.
3. HTML, CSS, JavaScript: для створення інтуїтивного інтерфейсу користувача.
4. SQLite: SQL база для зберігання структурованої інформації про користувачів, вантаж

# Висновки

1. Гнучкість та розширюваність системи: Структура бази даних, яка включає зв'язки між різними моделями, забезпечує гнучкість і можливість розширення. Використання зовнішніх ключів дозволяє чітко структурувати дані за категоріями (країни, регіони, міста, типи транспорту тощо), що забезпечує зручний доступ і обробку інформації. Це дозволяє легко додавати нові функціональності, наприклад, підтримку нових типів вантажів або транспорту.
2. Забезпечення цілісності даних: Зовнішні ключі та типи зв'язків між моделями гарантують цілісність даних, запобігаючи появі несумісних або неповних записів. Встановлені зв'язки між моделями забезпечують коректне збереження та перевірку даних на рівні бази даних, що дозволяє мінімізувати можливість виникнення помилок.
3. Ефективність пошуку та обробки даних: Моделі з такими полями, як ManyToManyField та ForeignKey, сприяють більш швидкому та ефективному пошуку і фільтрації даних. Наприклад, фільтрація вантажів або транспорту за критеріями, як-от країна, регіон чи тип кузова, буде виконуватися швидше завдяки добре спроектованій структурі зв'язків.
4. Підтримка різних варіантів та типів даних: Вибір відповідних типів полів для зберігання даних (наприклад, CharField, DecimalField, PositiveIntegerField) дозволяє ефективно працювати з різними типами інформації — від назв та описів до числових значень, таких як ваги чи об'єми. Це дає змогу забезпечити точність збереження даних і зручність роботи з ними.
5. Можливість інтеграції з іншими системами: Завдяки детально спроектованій структурі, база даних легко інтегрується з іншими модулями та системами (наприклад, системами управління логістикою, фінансовими сервісами або інтерфейсами для обробки платежів). Це дозволяє будувати масштабовані рішення з урахуванням потреб підприємств різного масштабу.
6. Простота в управлінні даними: Створення моделей для типів кузова, дозволів на транспорт та інших елементів дозволяє легко керувати різноманітними аспектами діяльності. Зокрема, можливість додавати нові типи транспорту або вантажів без значних змін у базі даних забезпечує гнучкість у налаштуванні й розширенні системи відповідно до змін на ринку або в компанії.
7. Масштабованість і підтримка великих обсягів даних: Оскільки система спроектована з урахуванням потреби в масштабованості, вона може ефективно обробляти великі обсяги даних (наприклад, при збільшенні кількості вантажів чи транспортних одиниць). Крім того, правильне використання індексів у базі даних і продуманих зв'язків між моделями забезпечить високу продуктивність навіть за умов великого навантаження.
8. Система звітності та аналізу: Завдяки організації даних у вигляді моделей, система здатна швидко генерувати звіти та проводити аналіз ефективності транспортних перевезень, вантажів, оплат і інших процесів, що є важливим для управління логістичними ланцюгами.

# Список Використаних джерел:

<https://docs.djangoproject.com/en/stable/topics/db/models/>

<https://en.wikipedia.org/wiki/Database_normalization>

<https://en.wikipedia.org/wiki/Haversine_formula>

<https://docs.python.org/3/library/datetime.html>

<https://www.postgresql.org/docs/>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

[https://en.wikipedia.org/wiki/Entity–relationship\_model](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model)

https://www.geeksforgeeks.org/normal-forms-in-dbms/

https://www.w3schools.com/sql/sql\_intro.asp

<https://en.wikipedia.org/wiki/Representational_state_transfer>

<https://docs.djangoproject.com/en/stable/ref/models/fields/>

<https://www.restapitutorial.com/>

https://www.json.org/json-en.html

<https://www.mongodb.com/docs/manual/>

https://www.redhat.com/en/topics/containers/what-is-docker

<https://en.wikipedia.org/wiki/ACID>

https://www.geeksforgeeks.org/types-of-database-management-systems/

<https://en.wikipedia.org/wiki/SQL_injection>

https://flask.palletsprojects.com/en/2.0.x/

https://cloud.google.com/sql

<https://docs.djangoproject.com/en/stable/topics/db/models/>

<https://en.wikipedia.org/wiki/Database_normalization>

<https://en.wikipedia.org/wiki/Haversine_formula>

<https://docs.python.org/3/library/datetime.html>

<https://www.postgresql.org/docs/>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

[https://en.wikipedia.org/wiki/Entity–relationship\_model](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model)

https://www.geeksforgeeks.org/normal-forms-in-dbms/

https://www.w3schools.com/sql/sql\_intro.asp

<https://en.wikipedia.org/wiki/Representational_state_transfer>

<https://docs.djangoproject.com/en/stable/ref/models/fields/>

<https://www.restapitutorial.com/>

https://www.json.org/json-en.html

<https://www.mongodb.com/docs/manual/>

https://www.redhat.com/en/topics/containers/what-is-docker

<https://en.wikipedia.org/wiki/ACID>

https://www.geeksforgeeks.org/types-of-database-management-systems/

<https://en.wikipedia.org/wiki/SQL_injection>

https://flask.palletsprojects.com/en/2.0.x/

https://cloud.google.com/sql