НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Курсова робота

з навчальної дисципліни “Бази даних та інформаційні системи-2”

на тему:

«Організація івентів для студентів»

Виконав :

студент IV курсу, групи КМ-63

Артеменко Ярослав Костянтинович \_\_\_\_\_\_\_\_\_\_

Керівник:

Ковальчук-Химюк Л. О. \_\_\_\_\_\_\_\_\_\_

Київ – 2019

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Створити систему, що буде надавати користувачу можливість шукати заході, які можуть бути цікавими для даного користувача. Для цього потрібно виконати перед проектне дослідження (проаналізувати бізнес-процеси, бізнес-правила та елементарні події), визначити сутності, що міститимуться у базі даних, проаналізувати операції, що можуть виконуватись над кожною із сутностей, забезпечити цілісність системи. Додатково виконати кореляційний та кластерний аналіз. Розгорнути завдання на віддаленому сервері.

АНОТАЦІЯ

В пояснювальній записці до курсової роботи описано основні етапи розробки програмного забезпечення для поставленої задачі. Перший розділ містить інформацію про виконання перед проектного дослідження. У другому розділі надано детальну постановку задачі. В третьому розділі виконано моделювання бізнес-процесів. У четвертому розділі визначено модель даних, описано сутності інформаційної системи, а в розділі 5 надано графічне представлення моделі.

РЕФЕРАТ

Для реалізації даного онлайн сервісу було використано фреймворк для WEB розробки Flask. База даних, за допомогою якої виконувалось завдання – PostgreSQL 10. Власне модель створюється за допомогою мови програмування Python.

Веб-застосунок дозволяє переглядати дані, які знаходяться в даній системі, також є можливість створювати, змінювати та видаляти дані у всіх сутностях. Додавання даних в сутності та редактування даних валідується за допомогою WTFForms. З Python до HTML-сторінок передається інформація та відображається в браузері за допомогою шаблонізатора Jinja, а веб-сторінки зверстані за допомогою Bootstap-4.

Для більш зручного користування сайтом було розроблено сторінку авторизації, де користувач може або зареєструватися в системі, або увійти, якщо вже такі дані існують в базі даних. Після входу користувач може знайти івент, який він би хотів відвідати. Пошук відбувається за критеріями, які можна обрати в випадаючому списку (або за іменем івенту, або за датою проведення івенту). Користувач, який заходить в дану систему з паролем адміністратора, має можливість редагувати дані та додавати нові. Також переглядати всю інформацію, яка вже знаходиться в системі. В якості схеми нейронної мережі було використано структуру PNN. В даній роботі також присутні кластерний та кореляційний аналіз, результат яких відображається на окремій сторінці, так само і для нейронної мережі виділена окрема вкладка, де буде відображено результат її роботи.

# ЗМІСТ

[СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ 5](#_Toc29248700)

[ВСТУП 6](#_Toc29248701)

[1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ 7](#_Toc29248702)

[2 ПОСТАНОВКА ЗАДАЧІ 8](#_Toc29248703)

[2.1 Категорії користувача 8](#_Toc29248704)

[2.2 Класи даних 9](#_Toc29248705)

[2.3 Бізнес правила 10](#_Toc29248706)

[2.4 Матриця елементарних подій 11](#_Toc29248707)

[3 МОДЕЛЮВАННЯ БІЗНЕС ПРОЦЕСІВ 14](#_Toc29248708)

[4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ 15](#_Toc29248709)

[5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ 17](#_Toc29248710)

[ВИСНОВКИ 19](#_Toc29248711)

[СПИСОК ВИКОРИСТАННИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ 20](#_Toc29248712)

# СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

БД – база даних.

СУБД – система управління базами даних

НМ – нейронна мережа

CRUD – create, read, update, delete

# ВСТУП

В теперішній час інформаційні технології розвиваються занадто швидко і за дуже короткий час було створено багато сервісів та додатків, які полегшують життя населення нашої планети майже у всіх можливих сферах життя. Також через те, що молоде покоління (і не тільки) активно користується соціальними мережами і різними засобами для пошуку різних заходів або для знайомств з іншими людьми, темою даної роботи було обрано «Організація івентів для студентів».

На даний момент вже існує достатня кількість різних сервісів та додатків, які за допомогою яких можна шукати різні заходи у будь-якому куточку світу, такі як All Events in City, Universe та інші.

Основною задачею даної роботи являється створити підґрунтя для мікросервісу, який буде організовувати пошук івенту, який буде цікавим для людини, де сам пошук відбувається за критеріями, серед яких користувач має можливість обрати. Після вибору критерія пошуку та введення інформації для пошуку, користувач отримає бажану інформацію. Даний мікросервіс може бути цікавим для використання людям з різної вікової категорії, які бажають відвідати якісь цікаві заходи і не витрачати багато часу для пошуку даного івенту.

# 1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ

У короткому представленні ієрархія процесів наведена на рисунку 1.1.

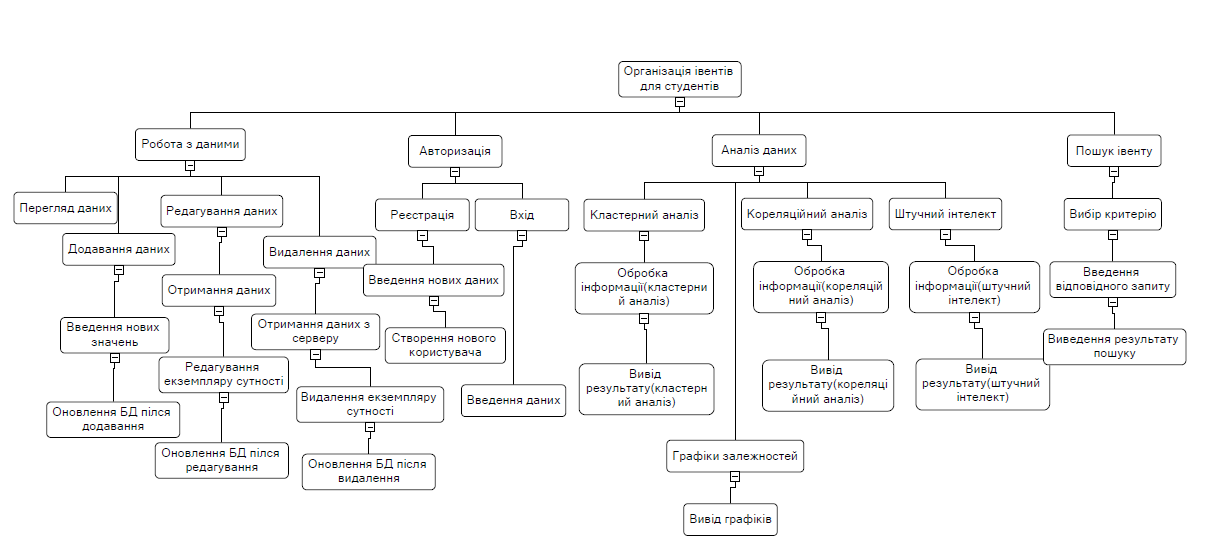


Рисунок 1.1 – Ієрархія процесів

# 2 ПОСТАНОВКА ЗАДАЧІ

Основною функцією мікросервісу було обрано організацію пошуку івенту, який являється цікавим для людини. Сам пошук відбувається по критеріям, які користувач може обрати.

Основною відмінністю простої автоматизації робочого процесу від пропонованого мікросервісу в тому, що завдяки пропонованому мікросервісу користувач може знаходити по своєму запиту івенти, які потім зможе відвідати. Такий пошук буде відбуватись по критеріям, таким як назва івенту або дата проведення івенту.

Вхідними даними для мікросервісу буде запит, який вводиться користувачем відповідно до обраної категорії.

Результат роботи мікросервісу: Список з івентів, які пропонуються після виконання пошуку.

Задача кореляційного аналізу: існує залежність від місця проведення івенту та кількості людей, які будуть відвідувати даний івент.

Задача кластеризації: користувачу ставиться у відповідність івент.

Задача машинного навчання: визначається, чи буде івент коштовним чи ні, і виводиться інформація відносно цього.

# 2.1 Категорії користувача

В даному курсовому проекті реалізовано авторизацію, за допомогою якої можна виконати вхід до сервісу як звичайний користувач або з правами адміністратора, з чого слідує, що організовано дві категорії користувачів. Існує адміністратор, у якого є доступ до виконання будь-яких CRUD-операцій прямо на сервісі, не використовуючи при цьому інтерфейс PostgreSQL та переглядати всю інформацію, яка існує в сервісі. Звичайний користувач має можливість лише виконувати пошук та переглядати результат пошуку.

# 2.2 Класи даних

Описання наявних у системі класів даних наведено у таблицях 2.2.1 – 2.2.4.

Таблиця 2.2.1 – Клас даних «Людина»

|  |  |  |
| --- | --- | --- |
| Сутність | Людина | |
| Опис сутності | Людина, яка є основним користувачем запропонованої системи | |
| Атрибути сутності | Опис атрибуту | Пов’язана сутність з атрибутом |
| Електронна пошта | Електронна адреса користувача, за допомогою якої здійснюється вхід або реєстрація в системі. | Івент (назва івенту) |
| Ім’я людини | Ім’я користувача системи | - |
| Телефон | Номер телефону користувача | - |
| Дата народження | Дата народження користувача | - |
| Пароль | Пароль, який користувач задає при реєстрації і який потім буде використовуватись для входу в систему | - |

Таблиця 2.2.2 – Клас даних «Івент»

|  |  |  |
| --- | --- | --- |
| Сутність | Івент | |
| Опис сутності | Захід, який може бути відвіданим людиною | |
| Атрибути сутності | Опис атрибуту | Пов’язана сутність з атрибутом |
| Назва івенту | Назва івенту, який запропоновано для відвідування | Конкурс (назва конкурсу),  Місце (назва місця) |
| Дата івенту | Дата проведення івенту | - |

Таблиця 2.2.3 – Клас даних «Конкурс»

|  |  |  |
| --- | --- | --- |
| Сутність | Конкурс | |
| Опис сутності | Конкурси, які проводяться на відповідному івенті | |
| Атрибути сутності | Опис атрибуту | Пов’язана сутність з атрибутом |
| Назва конкурсу | Назва конкурсу | - |

Таблиця 2.2.4 – Клас даних «Місце»

|  |  |  |
| --- | --- | --- |
| Сутність | Місце | |
| Опис сутності | Інформація про місця, де буде відбуватись івент | |
| Атрибути сутності | Опис атрибуту | Пов’язана сутність з атрибутом |
| Назва місця | Назва місця проведення | Івент (назва івенту) |
| Адреса місця | Адреса, за якою буде проводитись івент | - |
| Ціна | Ціна місця проведення | - |

# 2.3 Бізнес правила

Бізнес правила для даної системи мають наступний вигляд:

1. Користуватись сервісом може лише зареєстрований користувач.
2. Користувач може мати лише один профіль.
3. Якщо вже існує профіль з аналогічними даними, то новий створити не можна.
4. Увійти можна лише в існуючий профіль.
5. Увійти можна лише в свій профіль.
6. Бачити всі дані може лише користувач з правами адміністратора.
7. Редагувати дані може лише користувач з правами адміністратора.
8. У звичайного користувача є тільки можливість пошуку.
9. Дані повинні відповідати дійсності.
10. Пошук івентів відбувається лише за назвою і датою проведення.

# 2.4 Матриця елементарних подій

Таблиця 2.4.1 – сценарій для кейсу «Вхід»

|  |  |
| --- | --- |
| Актори | Користувач, Система |
| Мета | Увійти в систему |
| Передумови | Користувач хоче увійти в систему |
| Успішний сценарій:  Користувач бачить поля для введення даних(електронної пошти та паролю). Успішним вхід супроводжується переходом до функціоналу сервісу. | |
| Результат | Вхід у профіль |

Таблиця 2.4.2 – сценарій для кейсу «Реєстрація»

|  |  |
| --- | --- |
| Актори | Користувач, Система |
| Мета | Створити нового користувача |
| Передумови | Користувач хоче створити свій профіль |
| Успішний сценарій:  Користувач, у якого немає власного профілю, переходить на сторінку реєстрації, на якій він бачить поля для введення даних(електронної пошти, паролю та підтвердження паролю). Після успішної реєстрації користувач може увійти в систему. | |
| Результат | Створення нового користувача. |

Таблиця 2.4.3 – сценарій для кейсу «Пошук»

|  |  |
| --- | --- |
| Актори | Користувач, Система |
| Мета | Знайти потрібний івент |
| Передумови | Користувач хоче знайти потрібний івент |
| Успішний сценарій:  Після вибору категорії для пошуку і введення даних для пошуку, користувачу виводиться результат пошуку. | |
| Результат | Бажаний івент |

Таблиця 2.4.4 – сценарій для кейсу «Перегляд даних»

|  |  |
| --- | --- |
| Актори | Користувач, Система |
| Мета | Переглянути вже існуючі дані |

Продовження таблиці 2.4.4

|  |  |
| --- | --- |
| Передумови | Користувач хоче переглянути існуючі дані |
| Успішний сценарій:  Переглядати вже існуючі дані може лише користувач, який має права адміністратора. Перегляд даних відбувається шляхом переходу на відповідну сторінку з сутністю, яку користувач хоче переглянути. Після переходу користувач бачить всі дані. | |
| Результат | Перегляд існуючих даних |

Таблиця 2.4.5 – сценарій для кейсу «Редактування даних»

|  |  |
| --- | --- |
| Актори | Користувач, Система |
| Мета | Змінити дані |
| Передумови | Користувач хоче змінити існуючі дані |
| Успішний сценарій:  Редактувати вже існуючі дані може лише користувач, який має права адміністратора. Редактування даних відбувається шляхом переходу на відповідну сторінку з сутністю, яку користувач хоче відредактувати. Після вибору екземпляра сутності, який пілягає редактуванню, користувачу потрібно буде змінити існуючі дані на нові, після чого дані оновлюються. | |
| Результат | Відредактовані дані |

Таблиця 2.4.6 – сценарій для кейсу «Додавання даних»

|  |  |
| --- | --- |
| Актори | Користувач, Система |
| Мета | Додати дані |
| Передумови | Користувач хоче додати нові дані |
| Успішний сценарій:  Додавати дані може лише користувач, який має права адміністратора. Додавання даних відбувається шляхом переходу на відповідну сторінку з сутністю, до якої користувач хоче додати нові дані. Після переходу, користувачу потрібно буде ввести нові дані, після чого дані оновлюються. | |
| Результат | Створюються нові дані |

Таблиця 2.4.7 – сценарій для кейсу «Видалення даних»

|  |  |
| --- | --- |
| Актори | Користувач, Система |
| Мета | Видалити дані |
| Передумови | Користувач хоче видалити дані |
| Успішний сценарій:  Видаляти дані може лише користувач, який має права адміністратора. Видалення даних відбувається шляхом переходу на відповідну сторінку з сутністю, з якої користувач хоче додати нові дані. Після переходу, користувачу потрібно буде обрати дані, які потрібно видалити і натиснути відповідну клавішу, після чого дані оновлюються. | |
| Результат | Видаляються дані |

# 3 МОДЕЛЮВАННЯ БІЗНЕС ПРОЦЕСІВ

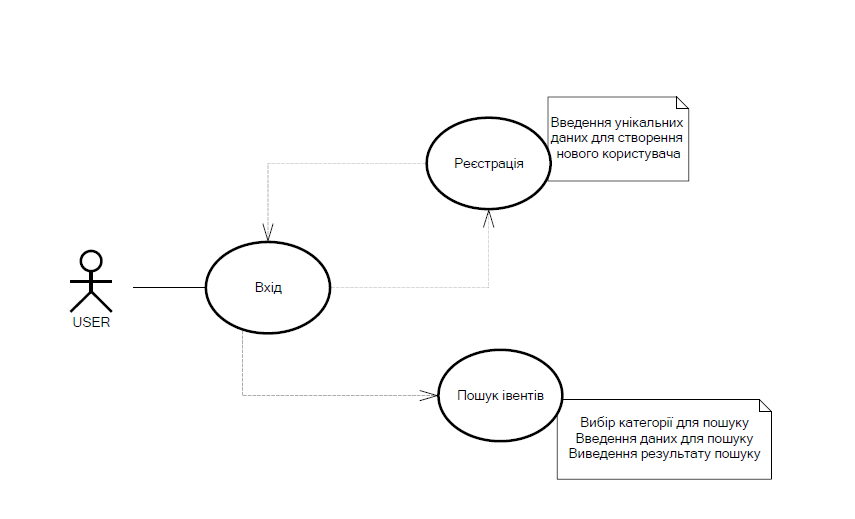


Рисунок 3.1 – Use-Case Diagram №1 для веб-застосунку

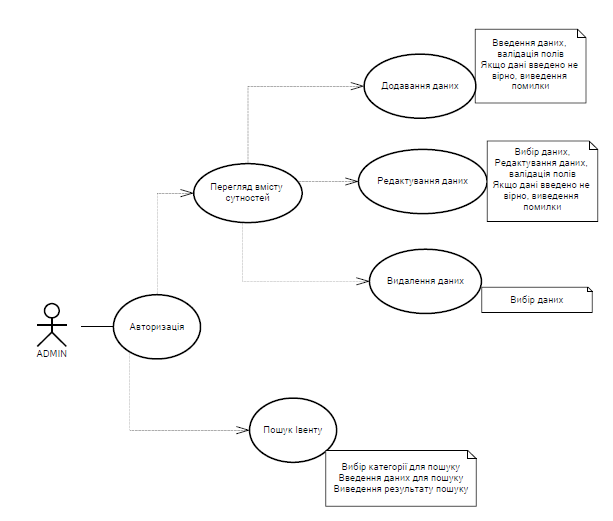


Рисунок 3.2 – Use-Case Diagram №2 для веб-застосунку

# 4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

В даному розділі сутності та зв’язки будуть змодельовані звичною мовою.

1. Людина хоче знайти івент, який має бажання відвідати. Людина має власну пошту, ім’я, номер телефону та дату народження. Оскільки людині потрібно буде проходити авторизацію, то людина також повинна мати пароль. З допомогою електронної пошти та паролю людина може або реєструватись, або робити вхід.

Відносно зв’язку, то одна людина може мати багато івентів, які потім буде мати можливість відвідати.

1. Івенти, які людина хоче відвідати, повинні надавати інформацію те, як називається івент та яка дата проведення івенту.

Дана сутність пов’язується відразу з двома сутностями: «Конкурс» та «Місце».

У зв’язку між івентом та місцем проведення івенту існує зв’язок «багато до багатьох», оскільки івент може мати багато місць для його проведення, а в одному місці може бути проведено багато івентів.

У зв’язку з івентом та конкурсами для івенту існує зв’язок один до багатьох, оскільки на івенті може бути проведено багато конкурсів.

1. Конкурси не надають багато інформації, лише назву конкурсу. Пов’язані з івентами.
2. Місця, в яких будуть проводитись івенти, містять інформацію про назву місця, адресу даного місця та вартість івенту в цьому місці.

Зв’язуються з івентами зв’язком багато до багатьох.

Концептуальна модель зображена на рисунку 4.1.

­­­­­

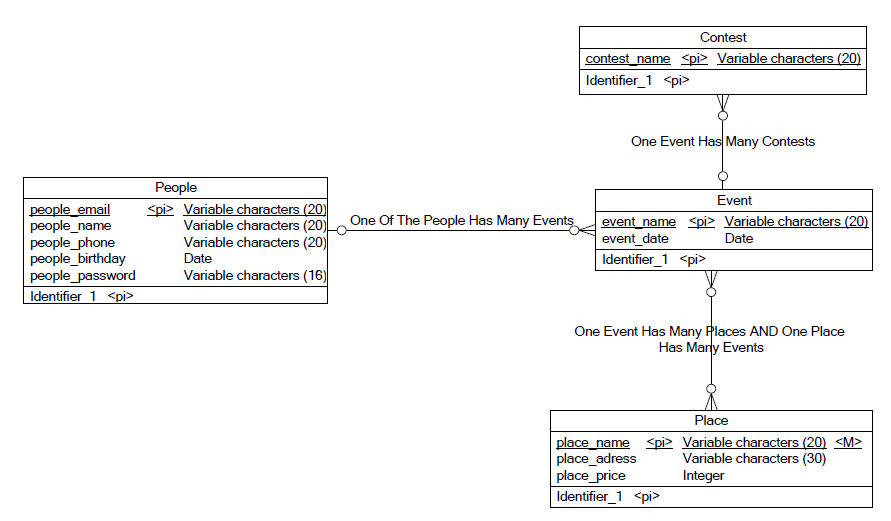


Рисунок 4.1 – Концептуальна діаграма бази даних

# 5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

Після створення концептуальної моделі наступним кроком створюється логічна модель бази даних. Логічна модель зображена на рисунку 5.1.

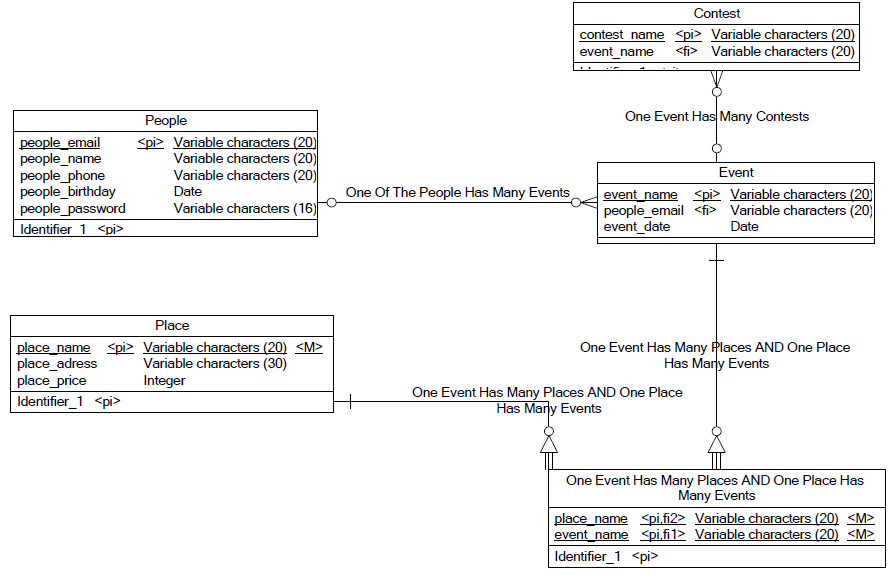


Рисунок 5.1 – Логічна модель бази даних

Далі логічна модель перетворюється на фізичну. Для цього в інтерфейсі Power Designer обираємо PostgreSQL (в цій версії програми наявна версія 8). Формується модель, що зображена на рисунку 5.2. Разом із моделлю Power Designer надає код, завдяки якому можна створити та заповнити таблиці в інтерфейсі PostgreSQL просто скопіювавши його з створених файлів, в одному з яких згенеровано код для створення таблиць та зв’язків, а в іншому заповнення бази даних.

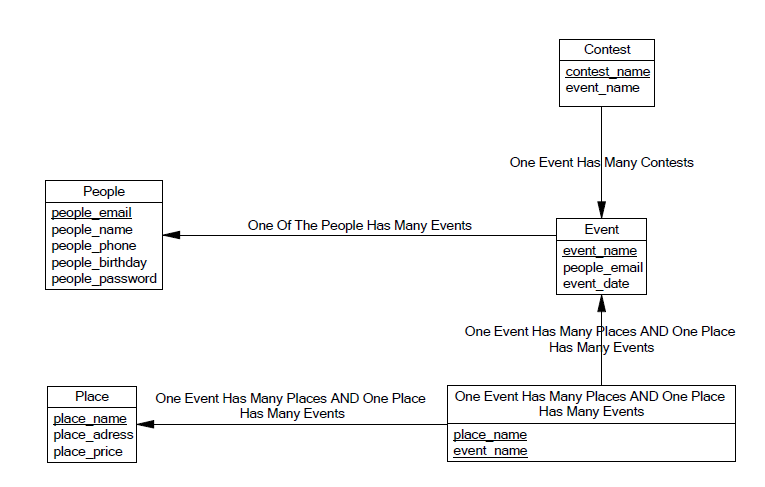


Рисунок 5.2 – Фізична модель бази даних

# ВИСНОВКИ

В ході виконання курсової роботи було визначено наступні етапи впровадження проекту: побудова ієрархії процесів, обґрунтування постановки задачі, визначення сценаріїв перебігу елементарних процесів, визначення класів даних, сутностей та атрибутів, побудова моделей даних та інше.

Як результат виконання проекту досягнуто: інформаційна система, що дозволяє переглядати, додавати, іноді змінювати та видаляти зміст сутностей. База даних, що взята за основу – PostgreSQL, а на Python – SQLAlchemy. Веб-застосунок збудований на базі фреймворків Flask та Bootstrap-4. Ресурс розгорнутий на хостингу Heroku.

Було створено два типи користувачів. Користувач, який має права адміністратора, може переглядати всі дані, які є в базі даних та виконувати CRUD-операції з веб частини. Користувач, який не має прав адміністратора, може лише шукати івент. Також було створено реєстрацію, після якої користувач, даних якого не було в системі, може увійти в неї.

Також існують окремі вкладки, де користувач з правами адміністратора може переглянути результат роботи кореляційного та кластерного аналізу, а також результат роботи машинного навчання, яке визначає, чи буде івент коштувати дорого або навпаки.

# СПИСОК ВИКОРИСТАННИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. SQLAlchemy 1.3 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.sqlalchemy.org/en/13/>.
2. WTForms Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://wtforms.readthedocs.io/en/stable/>.
3. Python Heroku Deployment [Електронний ресурс] – Режим доступу до ресурсу:

<https://gist.github.com/bradtraversy/0029d655269c8a972df726ed0ac56b88>

1. Зміст головної частини курсової роботи [Електронний ресурс] – Режим доступу до ресурсу:

<http://amis.fpm.kpi.ua/dbis/dbis-control/70-course-work-content>

1. Вимоги до оформлення курсової роботи [Електронний ресурс] – Режим доступу до ресурсу:

<http://amis.fpm.kpi.ua/dbis/dbis-control/68-rules-of-study-text-requirements>