

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Фізико-математичний факультет
Кафедра інформатики та прикладної математики

«Допущено до захисту»

Завідувач кафедри

_____ Соловйов В.М.

«_____» _____ 2019 р.

Реєстраційний № _____

«_____» _____ 2019 р.

ПРОГРАМНА ПІДТРИМКА МОНІТОРИНГУ ПРАВОПОРУШЕНЬ

Кваліфікаційна робота студентки
групи Ім-14
ступеня вищої освіти «магістр»
спеціальності 014.09 Середня освіта (Інформати-
ка)

Хомінятич Альбіни Вячеславівни

Керівник: к. ф.-м.н.

Мерзликін Павло Володимирович

Оцінка:

Національна шкала _____

Шкала ECTS ___ Кількість балів ___

Голова ЕК _____

Члени ЕК _____

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ ПРАВОПОРУШЕНЬ	5
1.1. Особливості соціальних геоінформаційних систем	5
1.2. Огляд наявних програмних реалізацій	8
1.3. Огляд цифрових картографічних інструментів	12
1.4. Постановка задачі	21
ВИСНОВКИ ДО РОЗДІЛУ 1	22
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ПРАВОПОРУШЕНЬ	23
2.1. Логічна структура сайту	23
2.2. Обґрунтування вибору інструментів розробки	26
3.3. Проектування бази даних	35
ВИСНОВКИ ДО РОЗДІЛУ 2	38
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ	40
3.1. Налаштування фреймворку Django	40
3.2. Реалізація клієнтської частини сайту	51
3.3. Інструкція користувача	64
ВИСНОВКИ ДО РОЗДІЛУ 3	69
ВИСНОВКИ	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71

ВСТУП

Актуальність теми. Англійський філософ-матеріаліст, один із засновників теорії громадського договору й теорії державного суверенітету Томас Гоббс писав: держава необхідна не для того, щоб зробити життя людей раєм, а для того, щоб воно не стало пеклом [1].

Соціальна відповідальність відноситься до об'єктивної необхідності нести відповідальність за порушення соціальних норм, вона ґрунтується на соціальній природі поведінки людини. Давно сказано: не можна жити в суспільстві й бути вільним від суспільства. Це виражає характер взаємовідносин індивіда з суспільством, державою, колективом, іншими соціальними групами та утвореннями – з усім оточенням. Взаємні права, обов'язки та соціальні норми забезпечують порядок і спокій в суспільстві, тому ті, хто їх не дотримується, можуть нести в тій чи іншій формі соціальну відповідальність у вигляді певних несприятливих наслідків (осуд оточення, тиск громадської думки, осуд, догана, виключення з тієї чи іншої групи, організації, відмова в довірі, товариський суд і т.д.). Саме такого роду заходами підтримується дотримання соціальних норм і правил [4].

Як показує практика, покладатися на моральні цінності кожної людини загрожує наслідками. Тому існує правова система. Завдяки сучасним технологіям є можливість простіше регулювати такі питання. Особливо це актуально для великих міст, де існує велика транспортна розв'язка та живуть люди з різним соціальним становищем. Для таких цілей вже майже десятиліття існує багато інформаційних систем і програм для забезпечення моніторингу правопорушень, які допомагають працівникам правоохоронних органів.

Об'єкт дослідження: моніторинг правопорушень.

Предмет дослідження: програмні засоби для підтримки моніторингу правопорушень.

Мета: створити веб-орієнтований програмний засіб для підтримки моніторингу правопорушень.

Для досягнення мети слід розв'язати такі **задачі:**

- проаналізувати принципи роботи геоінформаційних систем моніторингу правопорушень;

- порівняти наявні програмні засоби, виявити їх переваги та недоліки;

- висунути функціональні вимоги до системи моніторингу правопорушень;

- розробити програмний засіб для підтримки моніторингу правопорушень;

- здійснити тестування й удосконалення створеної системи.

Методи дослідження: теоретичний аналіз наукової та навчальної літератури, аналіз, синтез, порівняння.

Новизна роботи полягає в тому, що в її рамках створено систему монітору правопорушень, адаптовану до умов вітчизняного законодавства.

Практичне значення одержаних результатів полягає в можливості використання розробленої програми для підтримки моніторингу правопорушень.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ ПРАВОПОРУШЕНЬ

1.1. Особливості соціальних геоінформаційних систем

Інструментарій соціальної картографії уможлиблює перетворювати соціальні дані в наочну картографічну форму, тобто створювати карти соціальних явищ і процесів, що протікають в будь-якій соціально-територіальній спільності. Соціальна реальність піддається негативному впливу злочинності, наркоманії, алкоголізму і т.д. [4]. Злочинність серед них – одне з найбільш значущих. Розробка соціальних карт, методів картографування визначається їх значимістю в пізнанні механізмів функціонування суспільства. Карта являє собою інструмент комунікації населення, адміністрації, розробників, громадських діячів для запобігання негативним тенденціям розвитку суспільства на певній території.

Карта правопорушень може дати відповіді на наступні питання:

- чому злочинність зосереджується на певній території?
- чому злочинці обирають цю територію, а не іншу?
- як пояснити географічні зміни в окремих видах злочинності?
- чи досягають рішення щодо певного виду злочинності на конкретній території очікуваного ефекту?

Таким чином, картографування правопорушень передбачає візуальне представлення скупчень злочинності та їх характеристик і служить цінним інструментом в розробці й реалізації стратегій боротьби зі злочинністю.

Картографія злочинності (Рис.1.1.) як галузь соціальної картографії надає можливості для картографування стану, структури, ди іки злочинності та правопорушень, її фонових явищ та ін. В коло проблем входить і дослідження можливостей використання карт в управлінні процесом боротьби зі злочинністю, прогнозування й планування профілактики злочинів.

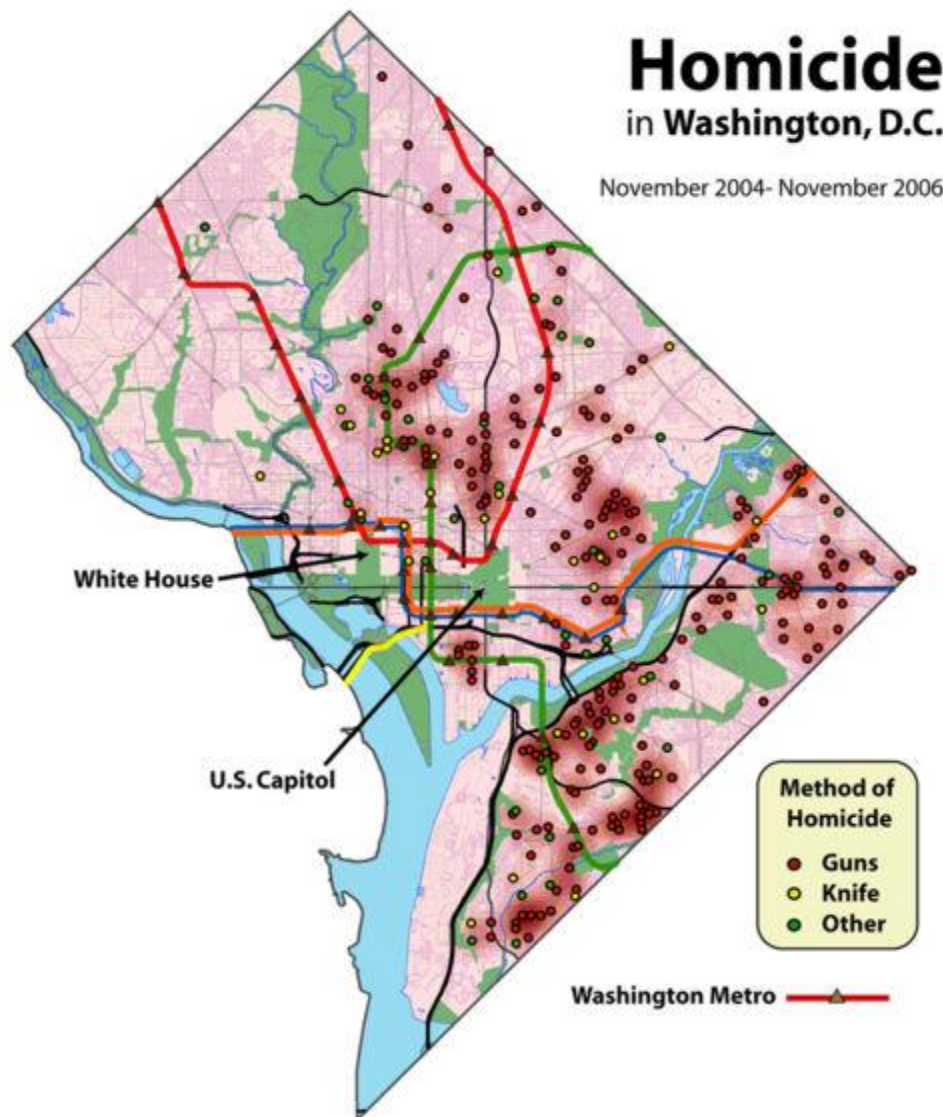


Рис.1.1. Приклад теплової кримінальної карти скоєних вбивств за листопад 2004 - листопад 2006 року. Вашингтон, Округ Колумбія, США[32].

Виділяють наступні групи карт злочинності та правопорушень:

- карти фонових явищ (причин і умов) злочинності;
- рівня злочинності;
- динаміки злочинності та правопорушень;
- карти, що відображають загальні характеристики особистості злочинця;
- карти, що характеризують заходи боротьби зі злочинністю.

Важливим залишається завдання розробки типологічних карт (з використанням типології населених пунктів, створеної економіко-географами) і карт районування.

Аналітики злочинності використовують картування й аналіз злочинності, щоб допомогти працівникам правоохоронних органів (наприклад, начальнику поліції) приймати більш правильні рішення, виділяти ресурси й формулювати стратегії, а також проводити тактичний аналіз (наприклад, прогнозування злочинності, географічне профілювання). Нью-Йорк робить це за допомогою підходу CompStat [15], хоча цей спосіб мислення має більше значення в короткостроковій перспективі. Існують і інші підходи, пов'язані з такими термінами, як інформаційна політика, інтелектуальна політика, проблемно-орієнтована політика й політика співтовариства. У деяких правоохоронних органах криміналісти працюють на цивільних посадах, в той час як в інших агентствах криміналісти є присяжними офіцерами.

З погляду досліджень і політики картування злочинності використовується для розуміння закономірностей позбавлення волі й рецидивізму, надання допомоги в плануванні ресурсів і програм, оцінки програм з попередження злочинності або скорочення злочинності і подальше розуміння причин злочину. Тут треба згадати про теорію «розбитих вікон» [3] – це кримінологічна теорія, яка стверджує, що видимі ознаки злочинності, антигромадської поведінки й цивільних заворушень створюють міське середовище, яке заохочує подальші злочини та заворушення, включаючи серйозні злочини. Теорія передбачає, що методи поліцейської діяльності, спрямовані проти дрібних злочинів, таких як вандалізм, суспільного вжитку алкоголю і ухилення від плати за проїзд, допомагають створювати атмосферу порядку й законності, тим самим запобігаючи більш серйозні злочини.

Бум інтернет-технологій, зокрема технологій геоінформаційних систем на основі веб-технологій, відкриває нові можливості для використання картування злочинності для підтримки попередження злочинності.

Дослідження [15] показують, що функції, які надаються в веб-картографуванні злочинності, менші, ніж в більшості традиційних програм для складання карт злочинності. На закінчення слід зазначити, що в наявних роботах зі складання карт злочинності в Інтернеті основна увага приділяється підтримці поліцейської діяльності на рівні спільноти, а не аналітичним функціям, таким як аналіз моделей і прогнозування.

1.2. Огляд наявних програмних реалізацій

В Україні досвід кримінальної картографії не був перейнятий місцевими відділками міліції чи регіональних ГУ МВС. Проте існував соціальний проект з візуалізації злочинності zloch.in.ua [5] (Рис.1.2.), який зображав криміногенність в розрізі областей за деякими видами злочинів, а також створюється «гарячу» карту злочинності по Києву. Карта дозволяла візуалізувати рівень злочинності у місті та визначити проблемні вулиці. В планах розробників були ідеї додавання карти погано освітлених вулиць, розбитих ліхтарів, кіосків з продажем алкоголю тощо, які допомагали б пояснити соціальні причини злочинності. Все це лягає в основу кримінальної картографії – аналітичного способу боротьби із злочинністю, що поруч із виявленням гарячих кримінальних точок і тенденцій, знаходить соціальні передумови виникнення правопорушень. На даний час цей проект не функціонує.



Рис. 1.2. Головна сторінка соціального проекту з візуалізації злочинності zloch.in.ua

У 2018 році журналісти новинного порталу zaxid.net на основі опрацювання даних поліції з правопорушень за 2015-2017 рр. створюється теплова карта злочинів м. Львова[2] (Рис.1.3.).



Рис.1.3. Теплова карта правопорушень м. Львова від новинного порталу zaxid.net

CrimeMapping.com (Рис.1.4.) був розроблений TriTech Software Systems, щоб допомогти правоохоронним органам по всій Північній Америці надавати громадськості цінну інформацію про недавню злочинну діяльність в їхньому районі. Метою є допомогти поліцейським департаментам знизити рівень злочинності за рахунок підвищення інформованості громадян. Створення більшої відповідальності серед членів спільноти, орієнтованих на співпрацю з поліцією, і цей проект довів свою ефективність в боротьбі зі злочинністю [19]. CrimeMapping.com використовує передовий картографічний рушій, який допомагає забезпечувати високий рівень функціональності, а також гнучкість для користувачів. Дані про злочинність беруться на регулярній основі з системи обліку кожного департаменту, тому інформація, яку можна оглянути через веб-браузер, є найактуальнішою з доступних. Ці дані завжди перевіряються на точність, а вся адресна інформація узагальнюється за блоками, щоб забезпечити захист конфіденційності.

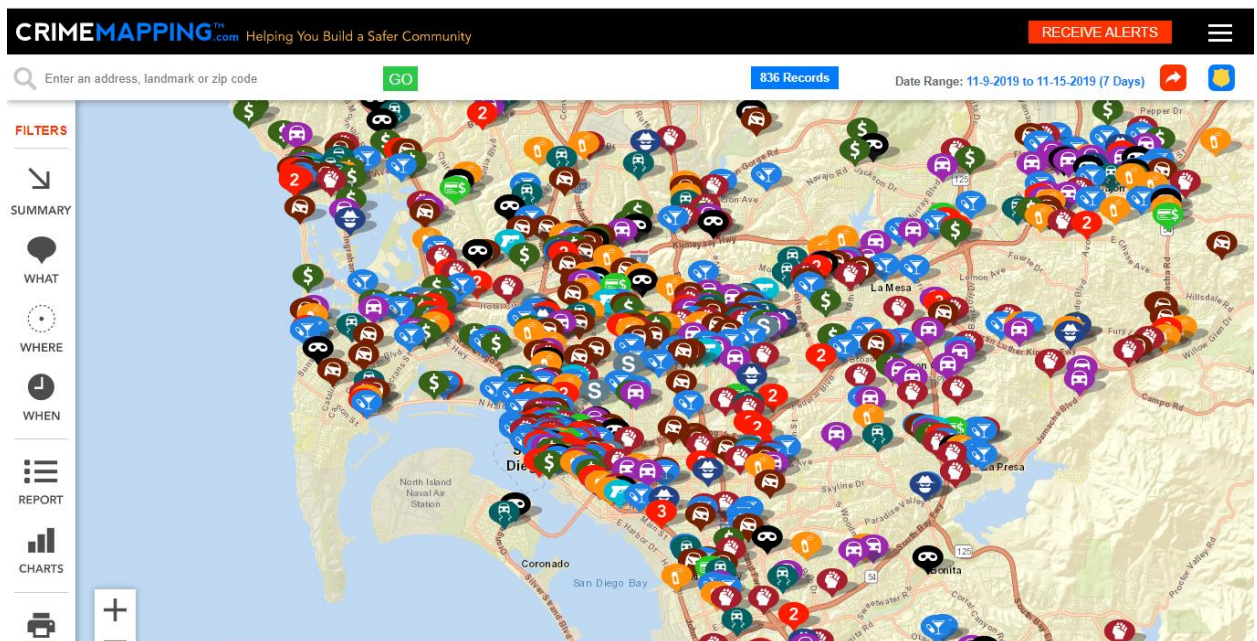


Рис.1.4. Головна сторінка сайту CrimeMapping.com

SpotCrime.com (Рис. 1.5.) – це Балтиморська компанія, заснована в жовтні 2007 року і знаходиться в приватній власності ReportSee, Inc.. Вона

надає загальнонаціональну злочинну інформацію про арешти, підпали, напади, крадіжки зі зломом, грабежі, перестрілки, крадіжки й вандалізм на карті Google. SpotCrime буде відображати дані для будь-якого поліцейського агентства, яке надає відкритий і необмежений доступ до даних про злочинність. Дані про злочинність в основному з поліцейських управлінь і новин. Будь-хто може отримати доступ до цих карт і мати можливість підписатися на отримання безкоштовних попереджень про злочини по електронній пошті і текстових повідомлень (SMS). Повідомлення по електронній пошті включає карту й відомості про злочин, який стався в зазначеному регіоні. У 2012 році SpotCrime.com запустив власний кримінальний сервіс Crimetip.us, що дозволяє користувачам анонімно повідомляти про злочини в своїй області.

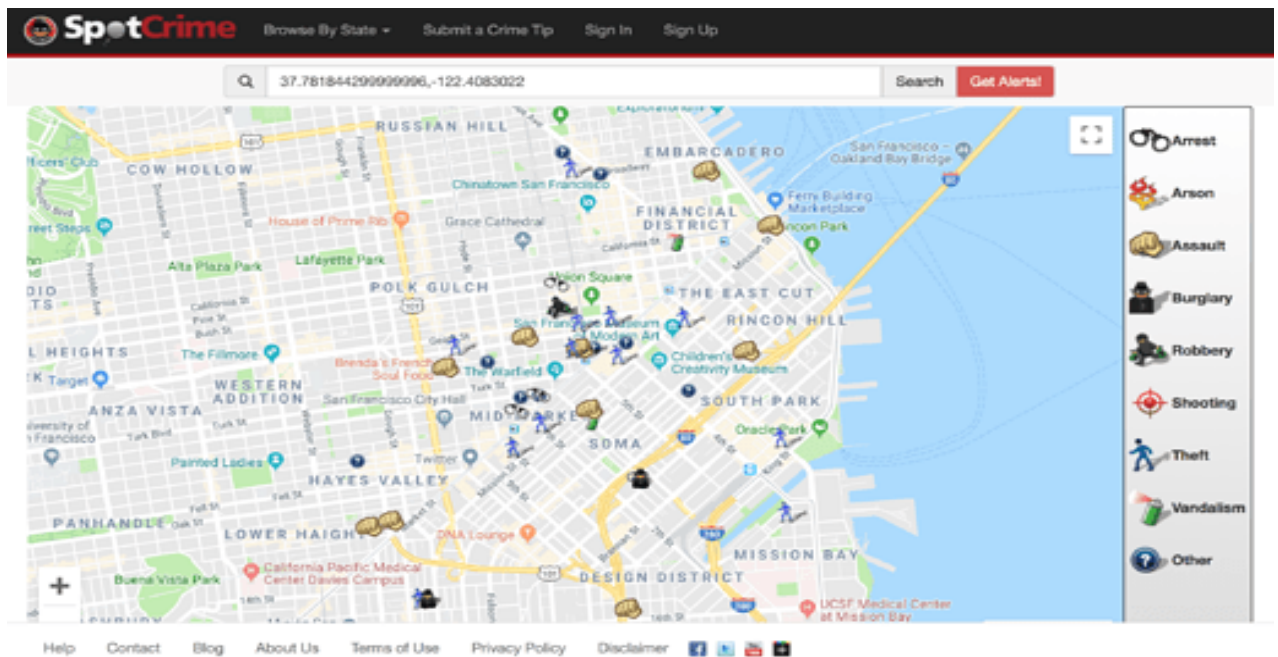


Рис.1.5. Головна сторінка сайту SpotCrime.com [33]

Карта LexisNexis® (Рис.1.6.) – це відкрита карта правопорушень, яка дозволяє правозахисним органам ділитися даними про правонарушення з громадськістю. Цей проект включає в себе основні карти, панелі моніторингу та аналітики. Додаток безкоштовний для суспільних та правозахисних органів.

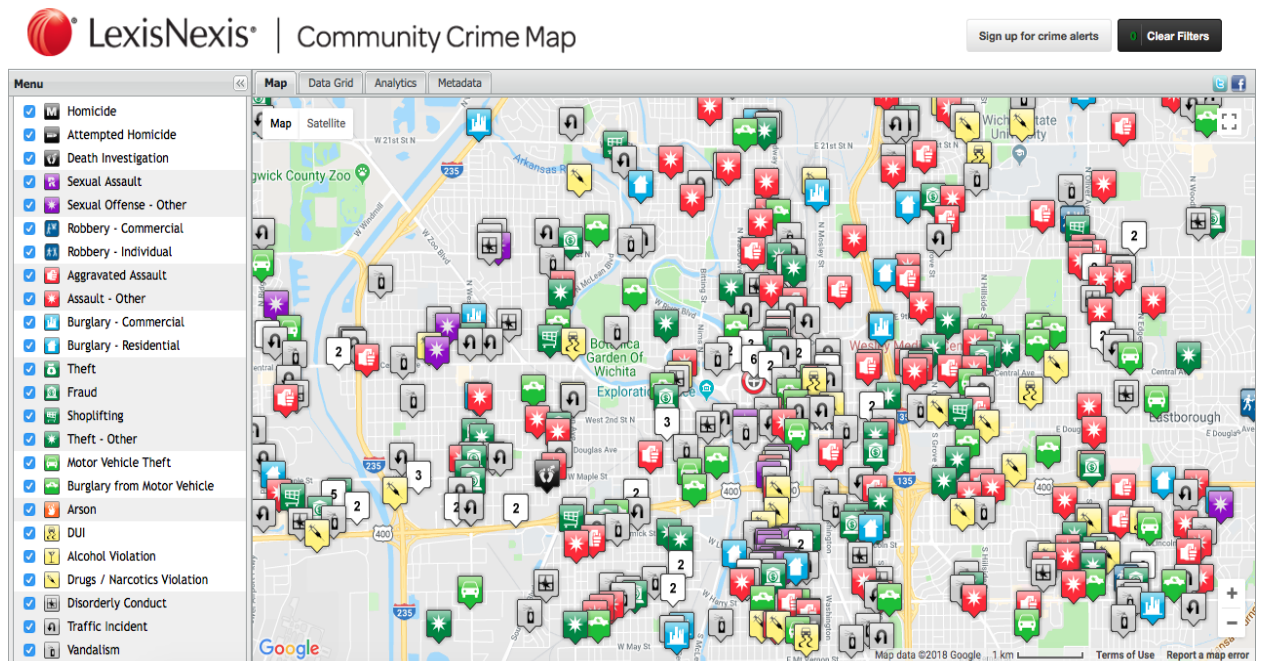


Рис.1.6. Відкрита карта правопорушень LexisNexis® [26]

Карта автоматично підтримує загальну інформацію про недавню злочинну діяльність і вдосконалює зв'язок між громадськістю і правозахисними органами і з підтримкою анонімних звернень. Проект має відкритий код на платформі HPCC Systems Big Data.

1.3. Огляд цифрових картографічних інструментів

В наші дні шанувальники карт можуть знайти необмежену кількість гарно оформлених інтерактивних і статичних цифрових карт. Існують карти, які відображають відкриті дані, такі як Карта скарг Нью-Йорка Кріса Уонга[30], яка відображає за останні сім днів 311 скарг міста. Існують карти,

які включають в себе безліч вихідних наборів даних, таких як Карта життя[27], набір карт, які відображають інформацію про діапазоні видів і списки видів для будь-якої географічної області у всьому світі. Є й веселі, і корисні карти, наприклад, Карта виживання зомбі-апокаліпсису[13], яка показує найближчі місця для продуктових магазинів, лікарень, складів і т.д.. Разом з небезпечними зомбі і зомбі. Існують інформаційні карти в режимі реального часу, такі як Tetsudo Now[35], анімована карта, яка використовує Google Maps – OpenLayers[21] і розклад руху, щоб показати токійські метро і автобуси в дорозі.

API-інтерфейси пропонують широкий спектр можливостей. Багато з цих API надають схожу функціональність, а деякі пропонують унікальні функції. Серед цих платформ і API-інтерфейсів можна знайти можливості картографування, які включають в себе інтерактивні і статичні карти, анімовані карти, геокодування, напрямки руху і транзиту, дані про висоту, потік трафіку і багато іншого. API також добре зарекомендували себе по відношенню до різноманітних критеріїв, таких як популярність, потенціал, документація, простота використання, функціональність.

Напротязі 14 років Карты Google (Рис.1.7.) як і раніше залишається безперечним лідером в області цифрових карт. Додаток для смартфонів на Google Картах відвідує в середньому понад 79 мільйонів унікальних користувачів на місяць[36], і це тільки додаток для смартфонів. Існує також офіційний додаток Карты Google для персональних комп'ютерів, не кажучи вже про мільйони сторонніх додатків, що використовують API Google Maps. На насправді існує безліч API Google Maps, в тому числі Embed API, Maps Image APIs, Places API, Web Services API та Google Maps API for Work Google надає дуже докладну документацію по API, а також приклади коду, бібліотеки, SDK і інші інструменти цифрового картування. Існує також засіб вибору API, яке розробники можуть використовувати, щоб знайти правильний API зіставлення для своїх проектів. Google постійно додає нові

функції і поліпшення в Карти Google, що допомагає компанії залишатися лідером цифрових карт в майбутньому.

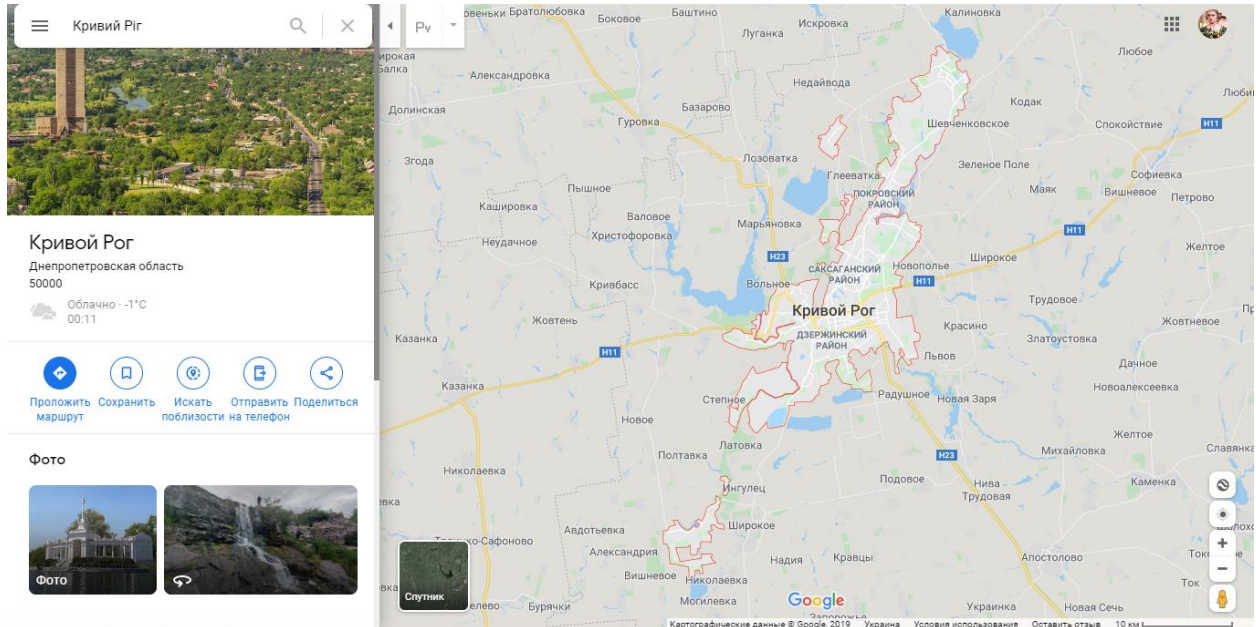


Рис.1.7. Приклад використання Карти Google для пошуку міста Кривий Ріг

Microsoft Bing Maps (Рис.1.8.) є дуже популярною картографічною платформою, хоча вона менш популярна як Google Maps. Microsoft постійно працює над додаванням нових функцій і поліпшує платформу Bing Maps [31]. Нещодавно компанія додала багато нових міст Стрітсайда і аерофотознімків з високою роздільною здатністю на картах Bing, а також нових тривимірних міст в додатку Bing Maps Preview. Документація Microsoft Bing Maps API дуже ретельно і докладно створена, хоча за нею досить важко слідувати. Є хороший AJAX Interactive SDK, який надає зразки функцій і фрагменти коду для розробників, що створюють додатки JavaScript. У міру того, як Microsoft додає більше функцій і можливостей в платформу Bing Maps, популярність її використання серед розробників і загальна популярність серед користувачів Microsoft Bing Maps постійно зростає.

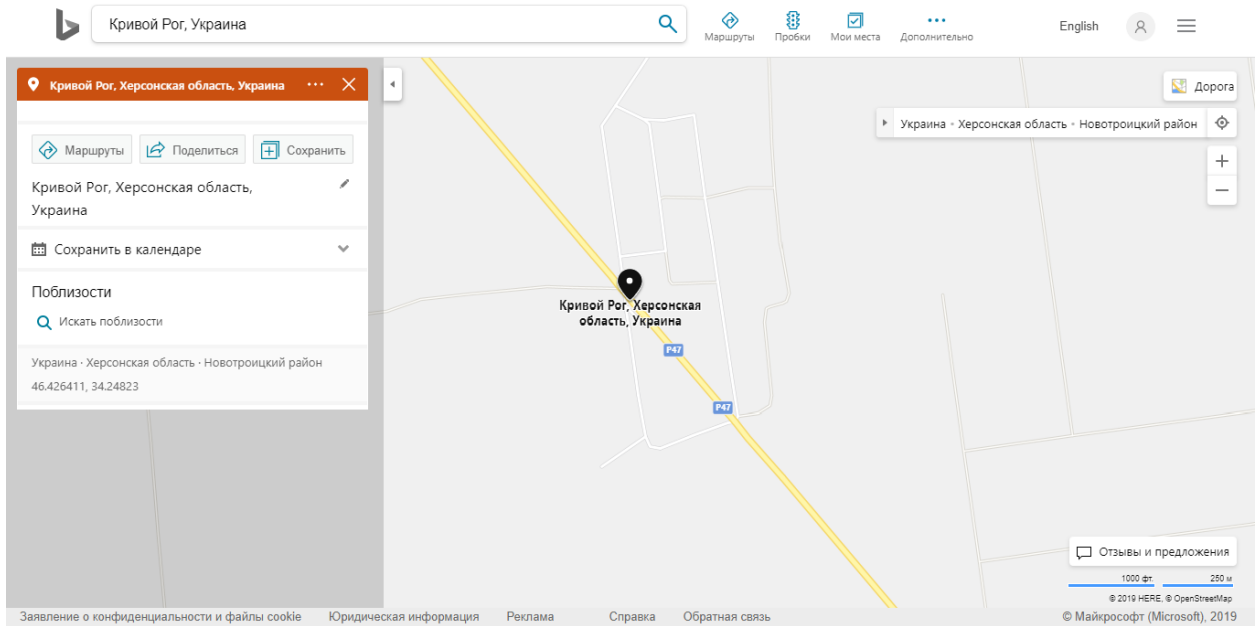


Рис.1.8. Приклад використання Microsoft Bing Maps для пошуку міста Кривий Ріг

OpenStreetMap (Рис.1.9.) – це проект, підтримуваний великою групою добровольців, які безкоштовно створюють і поширюють географічні дані по всьому світу. Багато додатків, такі як Watercolor by Stamen, HitchWiki і OpenWeatherMap, використовують дані OpenStreetMap. API OpenStreetMap відслідковує цей API, не вбуовують карту у веб-сторінку; це дозволяє отримувати необроблені геодані та зберігати їх у базі даних OpenStreetMap. Документація API досить інформативна, хоча інформація міститься на вікі-сторінці. OpenStreetMap представляє фізичні особливості, такі як комерційні будівлі та залізниці, використовуючи теги, і кожен тег описує географічний атрибут. OpenStreetMap має безліч функцій карти, включаючи зручності, будівництво, автошляхи, місця, магазини та водні шляхи.

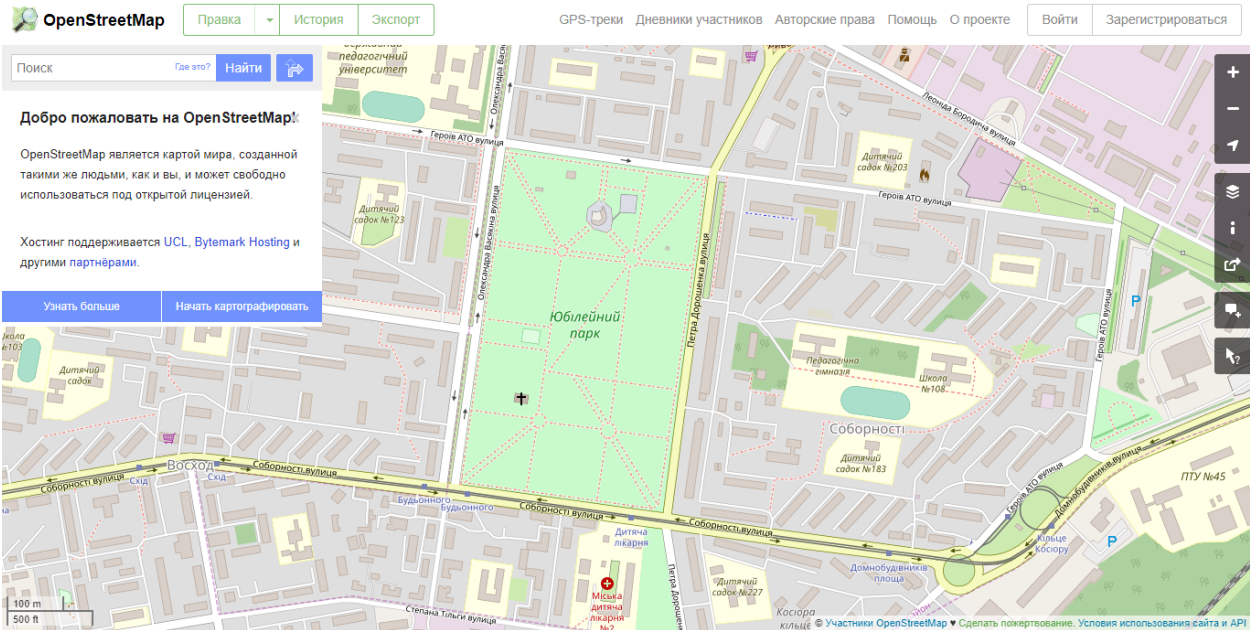


Рис.1.9. Початок роботи з OpenStreetMap

Google Map (Рис.1.10.) пропонує хороший вибір продуктів для цифрового картографування, включаючи веб-сайт Google Map.com, Google Map Mobile, Google Map Local, Google Map Enterprise і Google Map Developers. Існує дві версії платформи Google Map: ліцензійна і відкрита. Платформа Licensed Data доступна з ліцензією Google Map Enterprise Edition, а платформа Open Data доступна з безкоштовною і відкритою ліцензією. Слід зазначити, що OpenStreetMap є основним джерелом даних для відкритої платформи даних Google Map. Google Map API включає в себе добре розроблену документацію, будівник карт, майстер статичних карт, планувальник маршрутів і інші інструменти розробника. Крім того, Google Map надає JavaScript Maps API, набір Open Web Services, плагіни Open Leaflet і багато іншого. Google Map був заснований ще в 1967 році, задовго до того, як з'явився Google. Google Map почала пропонувати продукти цифрового картографування протягом деякого часу, перш ніж Google запустив Google Maps в 2005 році. Платформа Google Map добре зарекомендувала себе і надійна, і вона обслуговує мільйони відвідувачів Google Map.com і користувачів додатків Google Map.

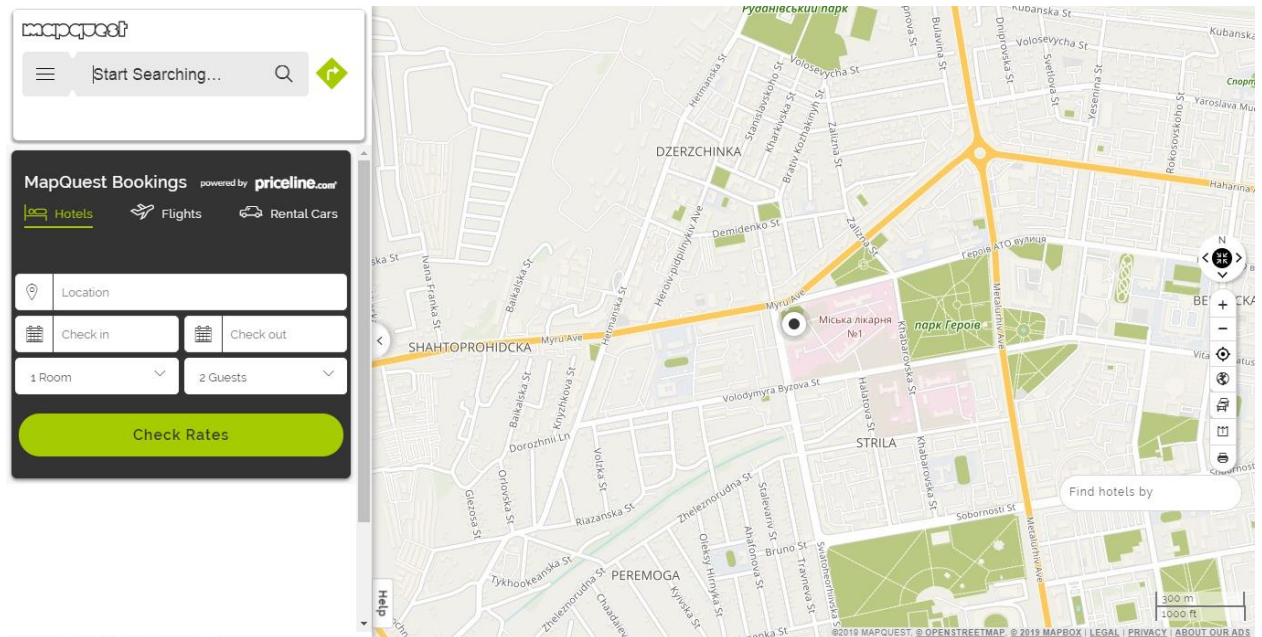


Рис.1.10. Приклад використання Google Мар для пошуку міста Кривий Ріг

Марбох (Рис.1.11.) надає набір інструментів цифрового картографування, які дозволяють швидко та легко додавати власні карти до програм. Платформа Марбох є відкритим кодом і містить текстури, ілюстрації, спеціальні маркери, векторні плитки, статичні карти, геокодування та інше. Марбох пропонує п'ять планів, починаючи від безкоштовного стартового плану до великого обсягу корпоративного плану. Марбох надає API, SDK та інші інструменти, які розробники можуть використовувати для додавання ди ічних карт та технології Марбох у свої програми. На платформі також представлений редактор Марбох, онлайн-редактор, який дозволяє користувачам швидко створювати власні карти за допомогою власних даних. Дані можна імпортувати з файлу електронних таблиць (CSV), GeoJSON, KML або GPX. Документація API добре розроблена і легко піддається дотриманню. Марбох також пропонує безліч демонстрацій та зразків коду, які розробники можуть використовувати як вихідну точку для своїх додатків. Хоча Марбох порівняно новий у порівнянні з Google Maps (Марбох був заснований у 2010 році), компанія наполегливо

працює над додаванням нових функціональних можливостей та залучення функцій, щоб допомогти відрізнити його від інших цифрових карт.

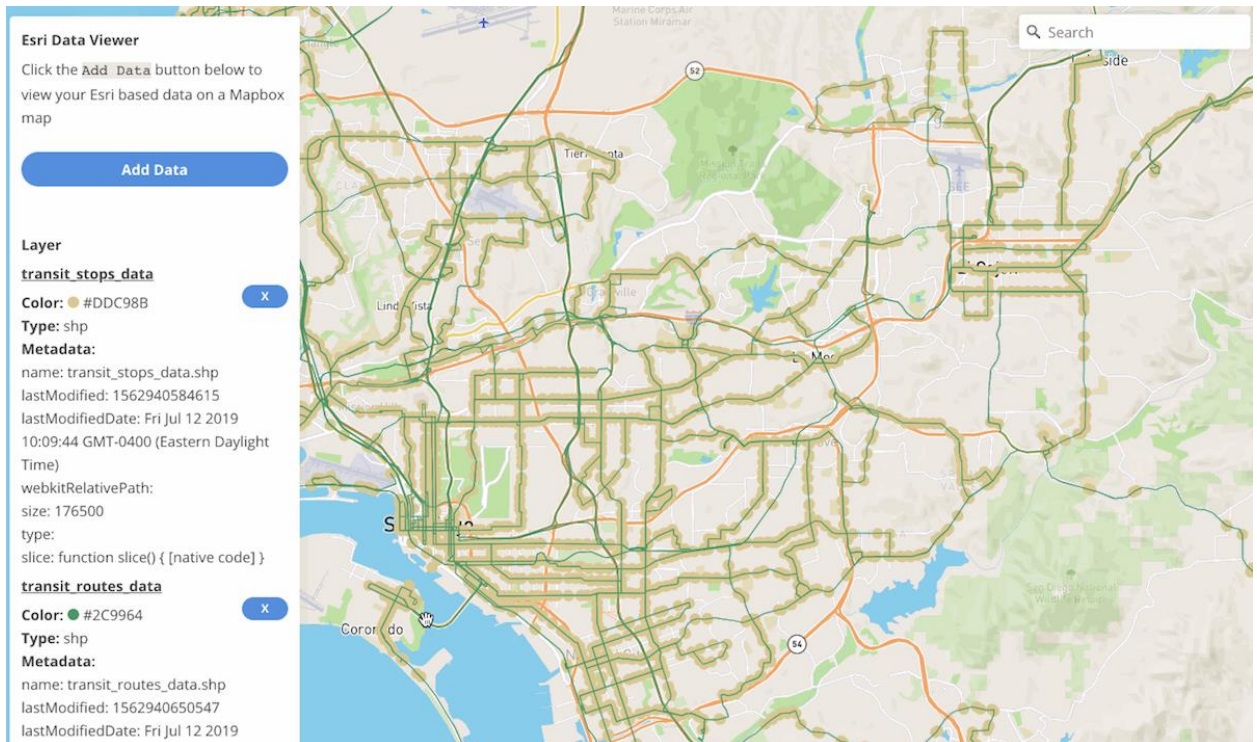


Рис.1.11. Приклад роботи з Mapbox

CartoDB (Рис.1.12.) – це механізм картографування, аналізу та візуалізації з відкритим кодом, який дозволяє розробникам створювати веб-та мобільні геопросторові програми. CartoDB був запущений у квітні 2012 року і з тих пір здобув цілу кількість гучних користувачів, зокрема National Geographic, NASA, The Guardian UK та Twitter. CartoDB пропонує декілька планів, що варіюються від безкоштовного базового плану до пакету функціональних можливостей підприємства. CartoDB надає бібліотеку JavaScript (CartoDB.js) та декілька API, які розробники можуть використовувати для додавання в свої програми з картами та геопросторових візуалізацій. CartoDB також надає редактору інтуїтивно зрозумілий інтерфейс перетягування, який дозволяє користувачам швидко створювати карти та візуалізувати дані. Документація як для платформи CartoDB, так і для редактора добре організована і легко піддається розумінню.

Популярність CartoDB зростає дуже швидко, незважаючи на відносно новизну платформи. Однією з найпопулярніших особливостей платформи CartoDB є Torque – анімована візуалізація, яка показує зміни в даних про локації з часом. Такі функції, як візуалізація крутного моменту, допомагають відрізнити CartoDB від інших цифрових карт та геопросторових платформ.

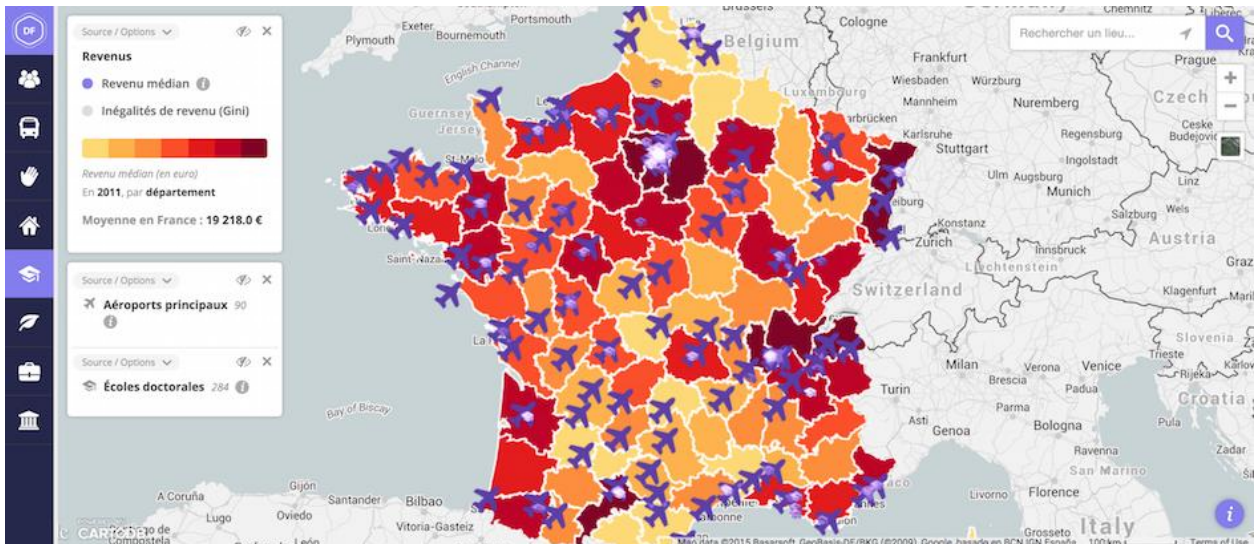


Рис.1.12. Приклад використання CartoDB для відслідковування польотів на території Іспанії

Esri ArcGIS (Рис.1.13.) – проект Есрі (Науково-дослідний інститут екологічних систем), який був заснований ще в 1969 році і спочатку був дослідницькою групою, яка аналізувала географічну інформацію для планування землекористування. У 1996 році Esri випустила MapObjects, свою першу платформу для публікації карт в Інтернеті. Сьогодні Esri зосереджується в основному на підприємствах і пропонує повну GIS-платформу, яка включає в себе хмарну платформу картографування ArcGIS, ArcGIS for Desktop, Esri CityEngine та Esri MapStudio. Хоча Esri зосереджується на комерційному ринку ГІС, компанія пропонує споживачам безкоштовний ГІС-переглядач під назвою ArcGIS Explorer Desktop. Esri має дійсно приємний портал для розробників, який добре продуманий і зручний у використанні. Портал розробників включає доступ до API ArcGIS та SDK,

хорошу документацію, живі демонстрації, зразки коду та інше. Esri пропонує окремі ціни для розробників, які використовують кредитну систему (для розробників, які не мають онлайн-підписки на ArcGIS). Структура кредитних цін дещо заплутана; простіший метод ціноутворення буде набагато кращим. Відповідно до недавньої статті Fortune.com[28], технологію Esri використовують понад 350 000 організацій по всьому світу.

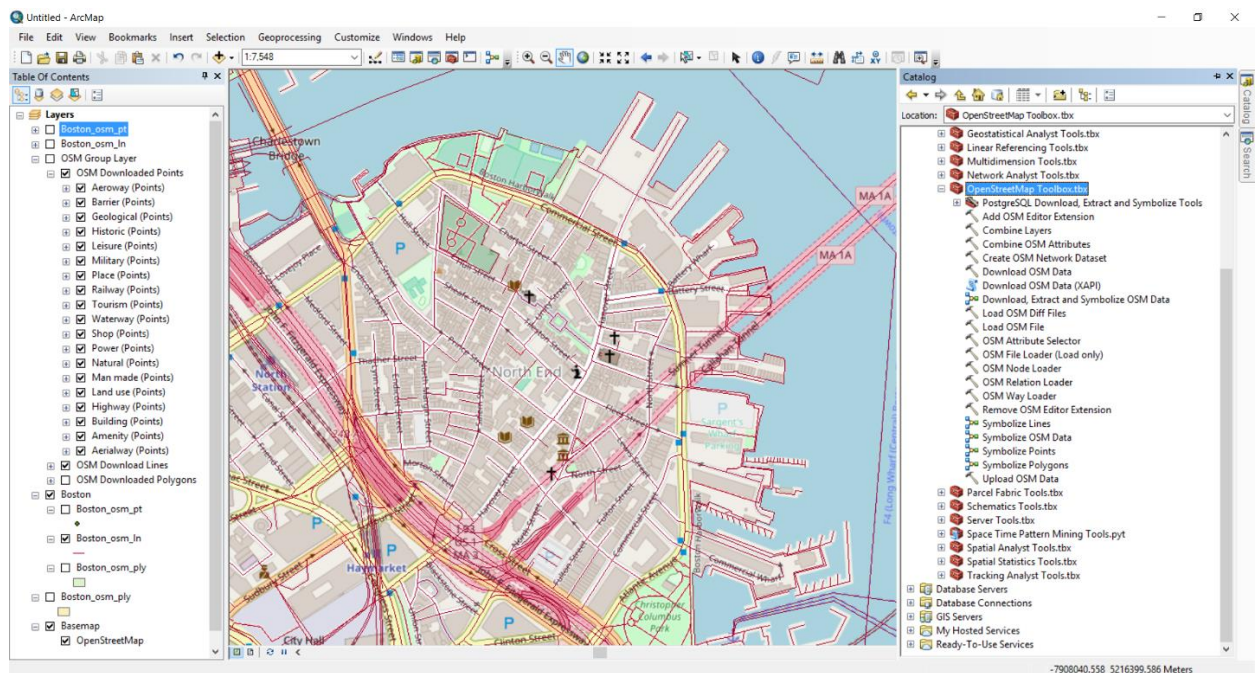


Рис.1.13. Приклад використання програмного забезпечення Esri ArcGIS

OpenLayers (Рис.1.14.) – це бібліотека JavaScript з відкритим кодом, яка використовує функції WebGL, Canvas 2D та HTML5 для візуалізації карт у сучасних веб-браузерах. OpenLayers здатний використовувати функції з OpenStreetMap, Bing, Google Map, Stamen та багатьох інших джерел відображення. OpenLayers також здатний надавати векторні дані з GeoJSON, TopoJSON, KML, GML та інших форматів географічних даних. Документація для OpenLayers добре організована і надає багато інформації для розробників. Галерея на веб-сайті OpenLayers містить великий вибір живих демонстрацій, і на GitHub доступно досить багато зразків коду. OpenLayers Track API – популярна бібліотека картографувань, багато в чому завдяки

його ліцензії на відкритий код та здатності витягувати плитки з інших платформ картографування.



Рис.1.14. Приклад використання бібліотеки OpenLayers

1.4. Постановка задачі

Необхідно створити сайт для соціального моніторингу правопорушень міста Кривий Ріг. Він являє собою Google Map з відмітками, коментарями та фото правопорушень, які залишають зареєстровані користувачі цього сайту, інші користувачі та гості сайту можуть бачити ці відмітки. Також в хедері знаходиться рядок пошуку за вулицями міста, для того, щоб швидше знайти необхідне місце.

ВИСНОВКИ ДО РОЗДІЛУ 1

Результати проведеного дослідження дають підставу зробити такі висновки:

Соціальна відповідальність ґрунтується на соціальній природі поведінки людини. Неможливо жити в суспільстві й бути вільним від суспільства. Взаємні права, обов'язки та соціальні норми забезпечують порядок і спокій в суспільстві. Для цього існує правова система, і завдяки сучасним технологіям є можливість набагато легше регулювати такі питання. Для таких цілей вже майже десятиліття існує багато інформаційних систем і програм для забезпечення моніторингу правопорушень, які допомагають працівникам правоохоронних органів та громадяни різних держав. Особливої уваги заслуговує наглядність та легкість сприйняття картографування правопорушень, яка передбачає візуальне представлення скупчень злочинності та їх характеристик і служить цінним інструментом в розробці й реалізації стратегій боротьби зі злочинністю.

Аналіз деяких проектів моніторингу правопорушень дозволив зробити такі висновки: майже усі проекти створені у вигляді сайту, де головну роль відіграють карти з місцевістю, з якою працює цей ресурс. Усі вони підтримують пошук за містами, а саме районами, вулицями міст. Деякі з них мають застарілі данні, а деякі оновлюються постійно. Декілька проектів підтримують соціальну ініціативу, але в загалом уся інформація надатися від правоохоронних органів. Існують карти правопорушень двох видів – теплові або з відмітками. Теплові карти позначають інтенсивність скоєних правопорушень за деякий період в зазначеному місці, а відмітки означають місце правопорушення та його вид в обраний час. Деякі проекти також підтримують повіомлення на e-mail або у вигляді SMS, якщо трапляється правопорушення в місці, яке обере користувач для моніторингу.

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ПРАВОПОРУШЕНЬ

2.1. Логічна структура сайту

На початковому етапі визначається тип сайту, створюється логічна структура сайту та його макет.

Вірний вибір логічної структури дозволить оптимально відобразити потреби відвідувачів, стане гарантією високої якості інтерфейсу користувача, тому важливість структурної розробки не можна недооцінювати. Структури інформаційних сайтів не такі розгалужені, як структури інтернет-магазинів і включають в себе розділи, присвячені тому чи іншому напрямку. Розділ відрізняється від сторінки, наприклад, інтернет-магазину тим, що часто є лістинг всіх документів на тему розділу без подальшої структуризації на більш вузькі напрямки. Також відмінною рисою є те, що зв'язки вибудовуються не тільки від розділу до рубрики, але і до кожної статті, що дає можливість вибудувати загальну зручну структуру всередині всього розділу.

Інформаційні сайти можна розділити на два види, від яких буде залежати їх структура:

- сайти-портали, присвячені якійсь широкій проблемі (наприклад материнство, мода, домашні тварини, тощо);
- сайти з вузькою тематикою, нішеві сайти (наприклад сайт присвячений фітнесу) [9].

Сайт-портал являє собою такий тип сайту, який організовує, зберігає та представляє інформацію користувачеві на певну тематику за допомогою різних інтерактивних інтернет-сервісів, які працюють в рамках цього сайту. Набір функціональних модулів та структура сайту влаштована таким чином, щоб користувачі сайту могли максимально та в повному обсязі отримувати інформацію. Зазвичай, на сайті-порталі завжди є стрічки новин і подій,

форуми, де можна обговорити різні теми та висловити свою думку. Ці сайти мають функціональність і характеристики всіх типів сайтів. Сайт-портал відрізняється від звичайних веб-сайтів. Можливість сайту-порталу зберігати великий обсяг даних, в тому числі і мультимедійних, з використанням для зберігання інформації не баз даних, а сховищ і вітрин даних, таких як Data Warehouse і Data Mart [10]. Цим сайтам обов'язково потрібна ефективна навігація, що дозволяє швидко переміщатися по розділах. Для того, щоб сайт-портал був привабливим для відвідувачів, необхідно надавати можливість інтерактивної взаємодії (наприклад, додавання матеріалів користувачами).

Тип логічної структури сайту «дерево» являє собою модель організації сайту, що зустрічається дуже часто. «Дерево» – це ієрархічна структура, яка дозволяє відвідувачам сайту управляти глибиною відвідування сайту. Користувачі можуть заходити тільки на сторінки верхніх рівнів ієрархії або ж «спуститися» до нижніх рівнів. При цьому воля вибору відвідувачів залежить від «ширини дерева»[8]. Урахувавши усі аспекти розробки логічної структури, було схематично створено структуру майбутнього сайту Kryvbassocial.com (Рис. 2.1). На початку «дерева» знаходиться головна сторінка сайту, на якій є всі необхідні посилання. Посилання відправляють користувача по «гілкам» на інші сторінки.

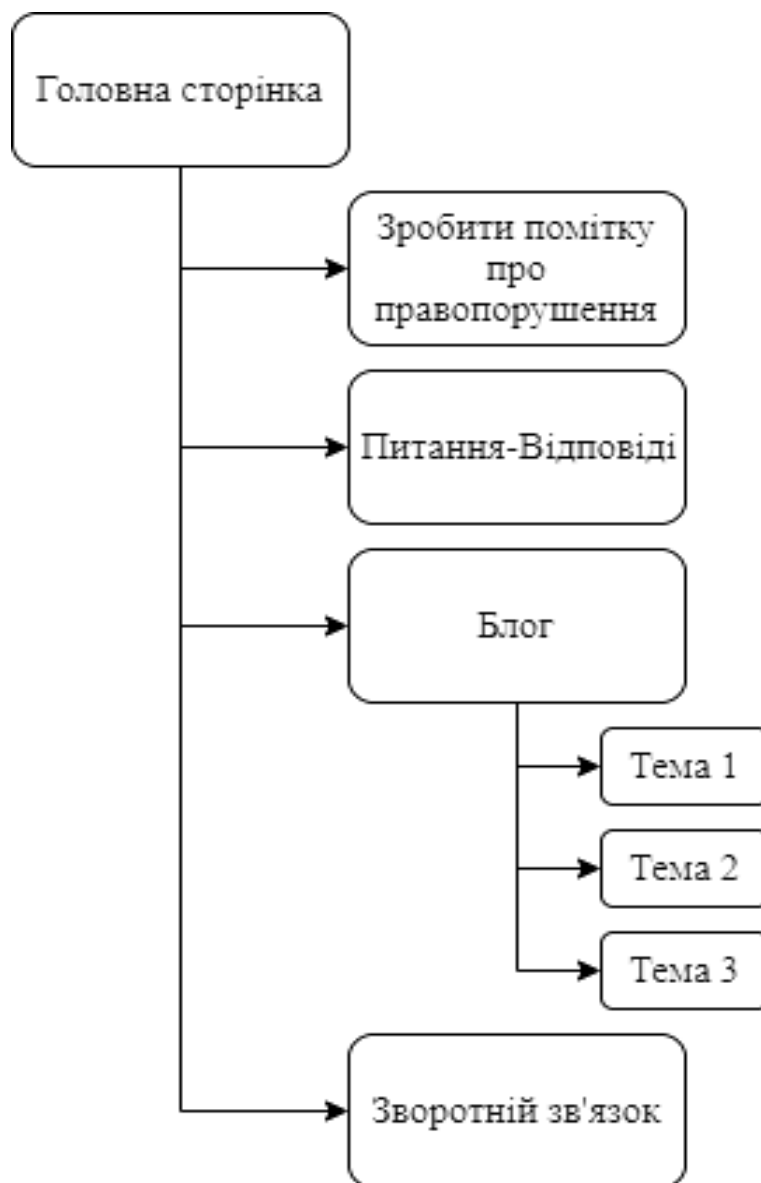


Рис.2.1. Логічна структура сайту Kryvbassocial.com

На початку розробки продуметься дизайн сайту. Розробка дизайну в загалом має три етапи. Спочатку дизайн створюється «від руки» схематично за допомогою аркушу паперу, потім використовуються графічний редактор, а вже в кінці виконується верстка сайту (Рис. 2.2.).



Рис. 2.2. Схематичне представлення етапів створення дизайну сторінки сайта

2.2. Обґрунтування вибору інструментів розробки

Зі створення функціональних макетів майбутніх сайтів починаються всі проекти веб-дизайну, всі вони створюються в Adobe Photoshop (Рис. 2.3.). Це буде прототип інтернет-ресурсу (порталу, блога, торговельного майданчика і т. д.), окремо взятої його сторінки, графічний образ з

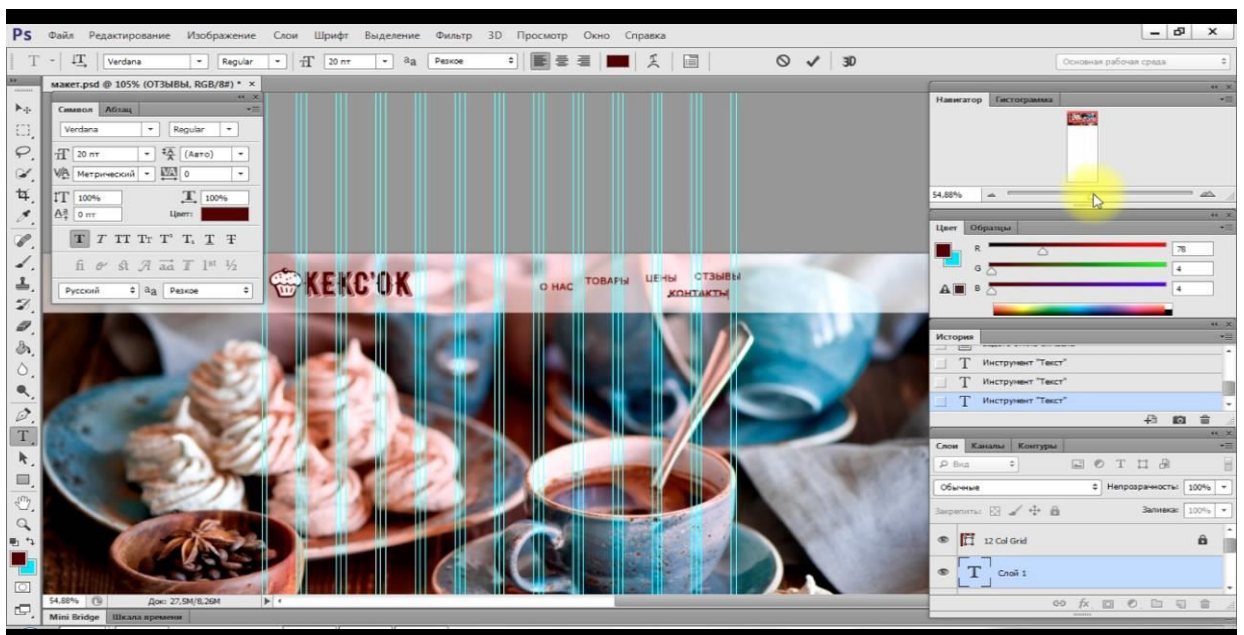


Рис. 2.3. Приклад процесу створення сторінки сайта в Adobe Photoshop

промальованими деталями різних рівнів. Картинка може бути загальною, а може досить докладно давати уявлення про все структурне наповнення сайту, навігацію й навіть інформаційний контент.

Додаток Creative Cloud Extract (Рис.2.4.) істотно змінює робочий процес переходу від композицій до коду для веб-дизайнерів і розробників. З легкістю надає необхідну інформацію про дизайн з композиції PSD, включаючи кольори, шрифти, каскадні таблиці стилів і ресурси, а потім їх можна використовувати при написанні коду для веб-дизайну. Завантажуючи макети в PSD форматі на сайт, переходячи на вкладку Extract та далі переглядаючи цей файл у вікні редагування та обираючи необхідний шар, можна отримати код CSS. Це дуже зручно, особливо для початківців.

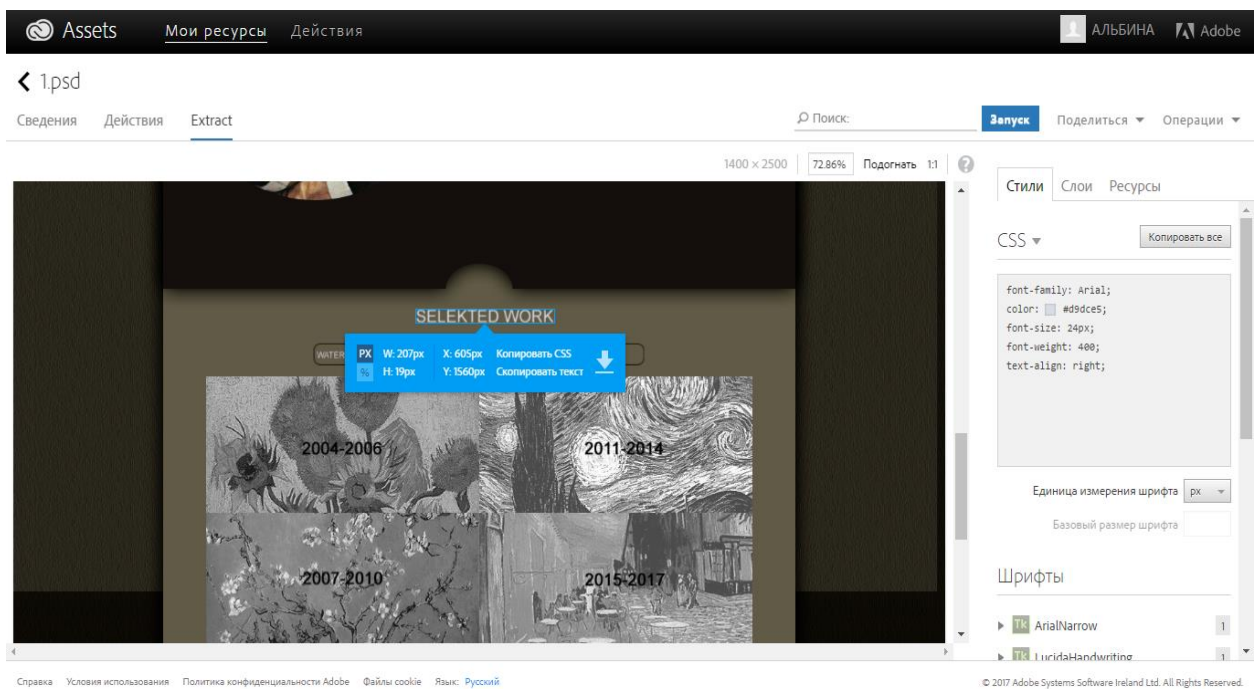


Рис. 2.4. Верстка макету сайту за допомогою додатка Creative Cloud Extract

Мова розмітки гіпертексту (HTML) – стандартна мова розмітки для документів, які призначені для відображення у веб-браузері. Загалом використовуються разом з каскадними таблицями стилів (CSS) та мовою сценаріїв, такою як JavaScript. Веб-браузери отримують HTML-документи з

веб-сервера або з локального сховища та переводять документи на мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично та спочатку містить підказки для зовнішнього вигляду документа. За допомогою HTML-конструкцій, зображення та інші об'єкти, такі як інтерактивні форми, можуть вбудовуватися у візуалізовану сторінку. HTML забезпечує засіб для створення структурованих документів, позначаючи структурну семантику тексту, таких як заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML розмежовані тегами, написаними за допомогою кутових дужок. Такі теги, як `` і `<input />` безпосередньо вводять вміст на сторінку. Інші теги, такі як `<p>`, оточують і надають інформацію про текст документа й можуть включати інші теги як під-елементи. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки [11].

JavaScript, часто скорочується як JS – це об'єктно-орієнтована мова програмування, яка відповідає специфікації ECMAScript. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок. Поряд з HTML та CSS, JavaScript є однією з основних технологій всесвітньої павутини. JavaScript включає інтерактивні веб-сторінки і є невід'ємною частиною веб-додатків. Переважна більшість веб-сайтів використовує його, а основні веб-браузери мають спеціальний механізм JavaScript для його виконання. Як мова з багатьма парадигмами, JavaScript підтримує об'єктно-орієнтований, імперативний і функціональний стилі програмування [7]. Вона має API для роботи з текстом, масивами, даними, регулярними виразами та DOM, але сама мова не має функцій вводу-виводу, мережевих засобів, сховищ чи графічних засобів. Ця мова покладається на головне середовище, в яке вона вбудована, щоб забезпечити роботу коду.

Bootstrap являє собою безкоштовний клієнтський фреймворк та набір інструментів з відкритим кодом, який містить в собі шаблони HTML та CSS,

спрямованих на розробку мобільних веб-інтерфейсів. Bootstrap використовує сучасні напрацювання в області CSS і HTML, тому необхідно бути уважним при підтримці старих браузерів.

Google Maps є веб-сервіс відображення, розроблений Google. Він пропонує супутникові знімки, аерофотознімки, карти вулиць, 360° панорамний вид вулиць (Street View), умови руху в режимі реального часу, а також планування маршрутів для подорожей пішки, на автомобілі, на велосипеді та в повітрі (в бета-версії), або на громадському транспорті. API Google Maps дозволяє розробникам інтегрувати Google Maps у свої веб-сайти (Рис. 2.5.).

Display a Google map.

English German Spanish French
Enable this content for this language

Title displayed on the site

Display the title ?

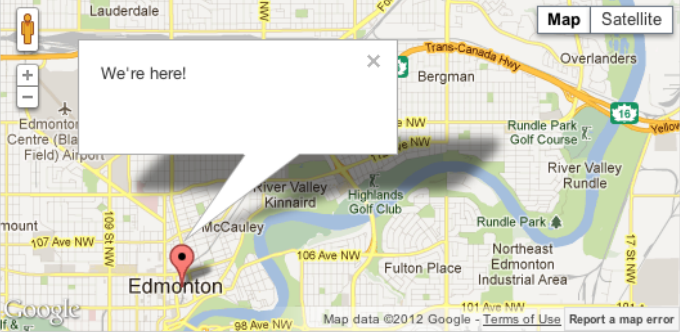
Title ?

Image Choose ?

URL link ?

Address

Map preview:



Description

Рис. 2.5. Приклад використання інтегрованих Google Maps на сайті

Для доступу до API потрібно мати ключ API, пов'язаний з обліковим записом Google Cloud з ввімкненням платежів. Використовуючи API Google

Maps, можна вставити Карти Google на зовнішній веб-сайт, на якому можуть бути накладені дані, пов'язані з сайтом. API Карт було розширено, щоб включити API для додатків Adobe Flash (але це застаріло), послугу для отримання статичних зображень карти та веб-сервіси для геокодування, генерації маршрутів руху та отримання висотних профілів. API Google Maps є безкоштовним для комерційного використання [20].

Django (Рис. 2.6.) – безкоштовний і вільний фреймворк для веб-додатків, написаний на Python. Фреймворк – це набір компонентів, які допомагають розробляти веб-сайти швидко і просто. Кожен раз при розробці веб-сайтів потрібні схожі компоненти: спосіб аутентифікації користувачів (вхід, вихід, реєстрація), панель управління сайтом, форми, інструменти для завантаження файлів і т.д.. Фреймворки існують, щоб полегшити процес розробки. Коли на сервер приходить запит, він переадресовується на Django. Для початку Цю частину процесу в Django виконує urlresolver (адреса веб-сайту називається URL – Uniform Resource). Він просто бере список шаблонів і намагається зіставити їх з URL. Django звіряє шаблони зверху вниз і, якщо щось збігається, він переправляє запит відповідної функції (яка називається view). Основна мета Django – полегшити створення складних веб-сайтів, керованих базою даних. Завдяки фреймворку є можливість повторного використання підключення компонентів, менше написання коду, швидкий розвиток і принцип «не повторювати». Python використовується повсюдно, навіть для файлів налаштувань і моделей даних. Django також надає додатковий адміністративний інтерфейс створення, читання, оновлення та видалення, який генерується за допомогою самоаналізу і налаштовується за допомогою моделей адміністратора. Деякі відомі сайти, які використовують Django, включають Службу громадського мовлення, Instagram [37], Mozilla [37], The Washington Times [37], Disqus [37], Bitbucket [37] і Nextdoor [37]. Він був використаний в Pinterest, але пізніше сайт перейшов на фреймворк, побудований над Flask [37].

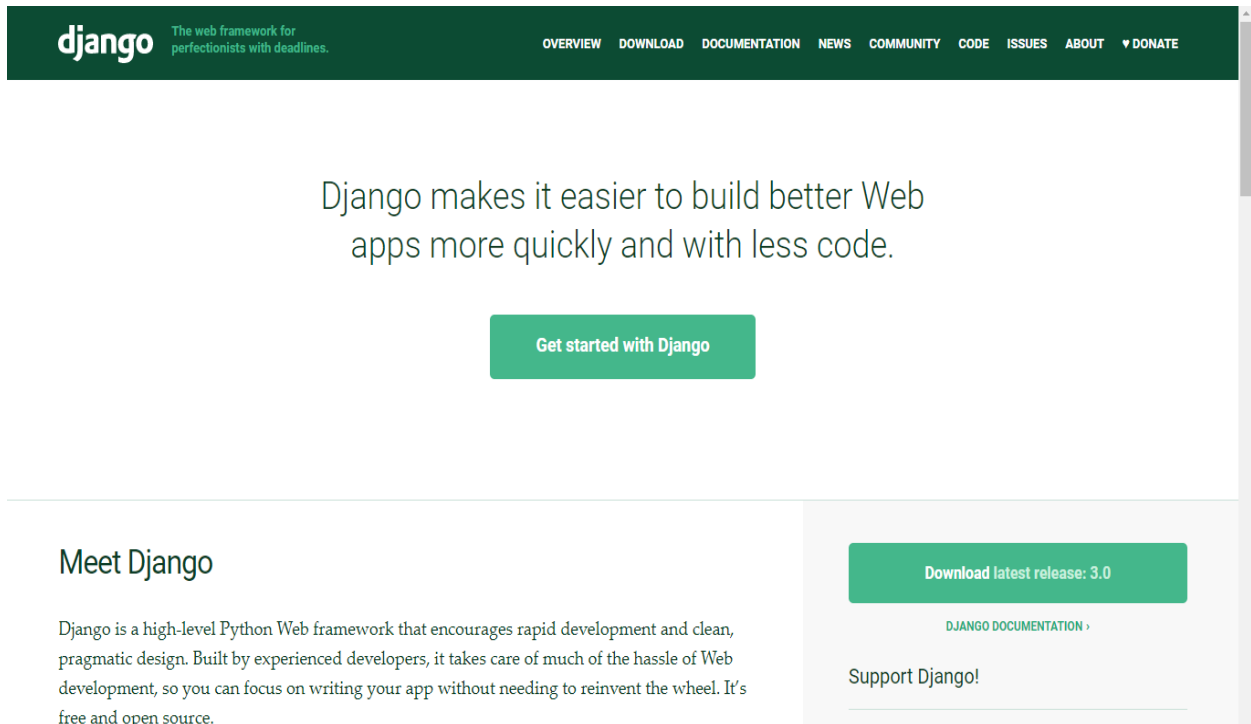


Рис. 2.6. Головна сторінка сайту Django

Для роботи з Django необхідний Python останньої версії (Рис.2.7.) – інтерпретована мова програмування високого рівня для загального програмування. Створена Гвідо ван Россумом і вперше випущена в 1991 році. Мова програмування забезпечує конструкції, які дозволяють чітко програмування на малих і великих масштабах. Python має тип динамічної системи та автоматичне керування пам'яттю. Вона підтримує декілька парадигм програмування, включаючи об'єктно-орієнтовану, імперативну, функціональну та процедурну, і має велику стандартну бібліотеку. Велика стандартна бібліотека Python, яку часто називають однією з найбільших сильних сторін, надає інструменти, що підходять для багатьох завдань. Для програм, що працюють в Інтернеті, підтримуються численні стандартні формати та протоколи, такі як MIME та HTTP [18]. Вона включає в себе модулі для створення графічних інтерфейсів користувача, підключення до реляційних баз даних, генерацію псевдовипадкових чисел, арифметику з

довільною десятковою точністю, маніпулювання регулярними виразами та тестування одиниць .

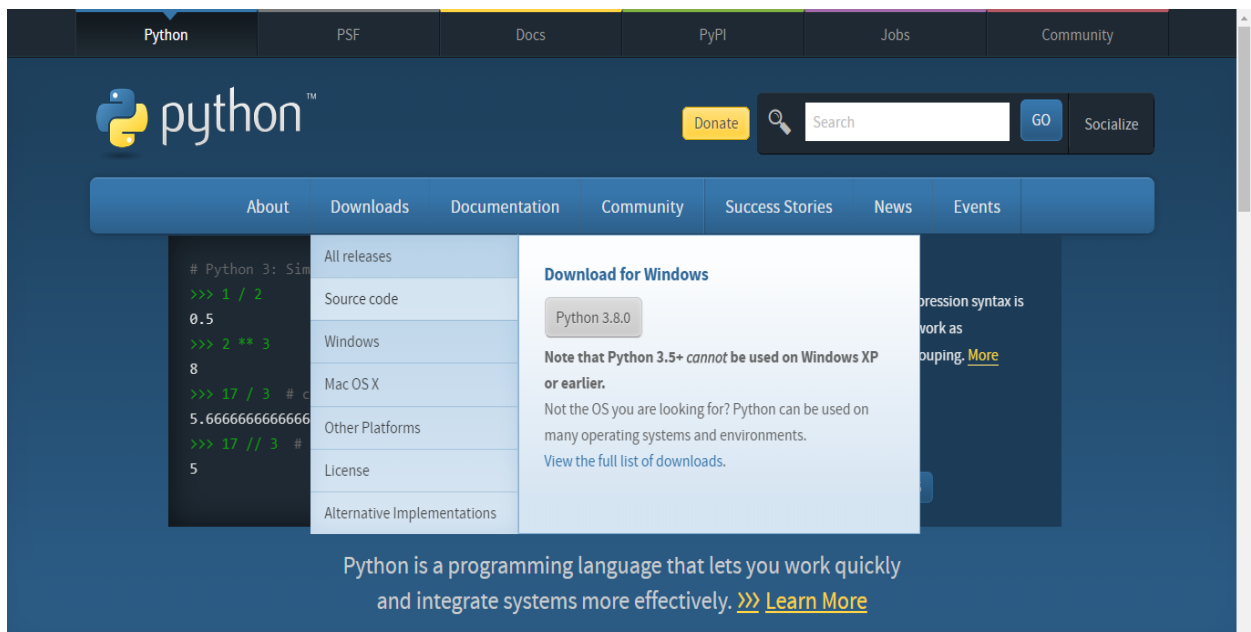


Рис.2.7. Сторінка завантаження Python версії 3.8.0

Деякі частини стандартної бібліотеки охоплюються специфікаціями (наприклад, реалізація інтерфейсу шлюзу веб-сервера (WSGI) який відповідає PEP 333). Вони вказуються за своїм кодом, внутрішньою документацією та тестовими комплектами (якщо вони надані). Однак, оскільки переважна частина стандартної бібліотеки є крос-платформенним кодом Python, для реалізації варіантів потрібно лише кілька модулів змінити або переписати.

Станом на листопад 2019 року, Python Index Package (PyPI), офіційний репозиторій програмного забезпечення сторонніх Python, містить понад 130 000 пакетів з широким спектром функціональних можливостей, в тому числі:

- графічні інтерфейси користувача;
- фреймворк;
- мультимедіа;
- бази даних;
- мережа;

- тестові рамки;
- автоматизація;
- веб-скрапінг;
- системне адміністрування;
- наукові обчислення;
- обробка тексту;
- обробка зображення.

Це багатопарадигмальна мова програмування. Об'єктно-орієнтоване програмування та структуроване програмування повністю підтримуються, і багато його функцій підтримують функціональне програмування та аспектно-орієнтоване програмування (у тому числі за допомогою метапрограмування та метаоб'єктів). Багато інших парадигм підтримуються за допомогою розширень, включаючи дизайн за контрактом та логічним програмуванням.

SQL (англ. Structured query language – мова структурованих запитів) – декларативна мова програмування, яка використовується в програмуванні і призначена для управління даними, що зберігаються в системі керування базами даних, або для обробки потоку в реляційній системі управління бази даних (СКБД)[34]. Це особливо корисно в обробці структурованих даних, тобто даних, що включають відносини між сутностями та змінними. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Стандарт SQL має функції для визначення зміни, перевірки й захисту даних. Це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також управління базами даних. MySQL зручний у випадку створення дуже великої бази даних. Використовується для створення веб-сторінок, оскільки має підтримку з боку різноманітних мов програмування, що є дуже зручним у даній роботі.

Для зручного створення бази даних було використано dbForge Studio for MySQL (Рис. 2.6.), який є універсальним інструментом для роботи з

MySQL сервером, який дозволяє розробникам MySQL і адміністраторам баз даних MySQL створювати і виконувати запити, розробляти і налагоджувати процедури і функції, а також автоматизувати управління об'єктами баз даних MySQL в зручному інтерфейсі.

Особливості dbForge Studio for MySQL:

- наявність засобів для централізованого адміністрування;
- інструменти для порівняння БД;
- візуальний профілювальник запитів;
- можливість управляти привілегами користувачів;
- наявність Дизайнера БД, який дозволяє будувати візуальні діаграми

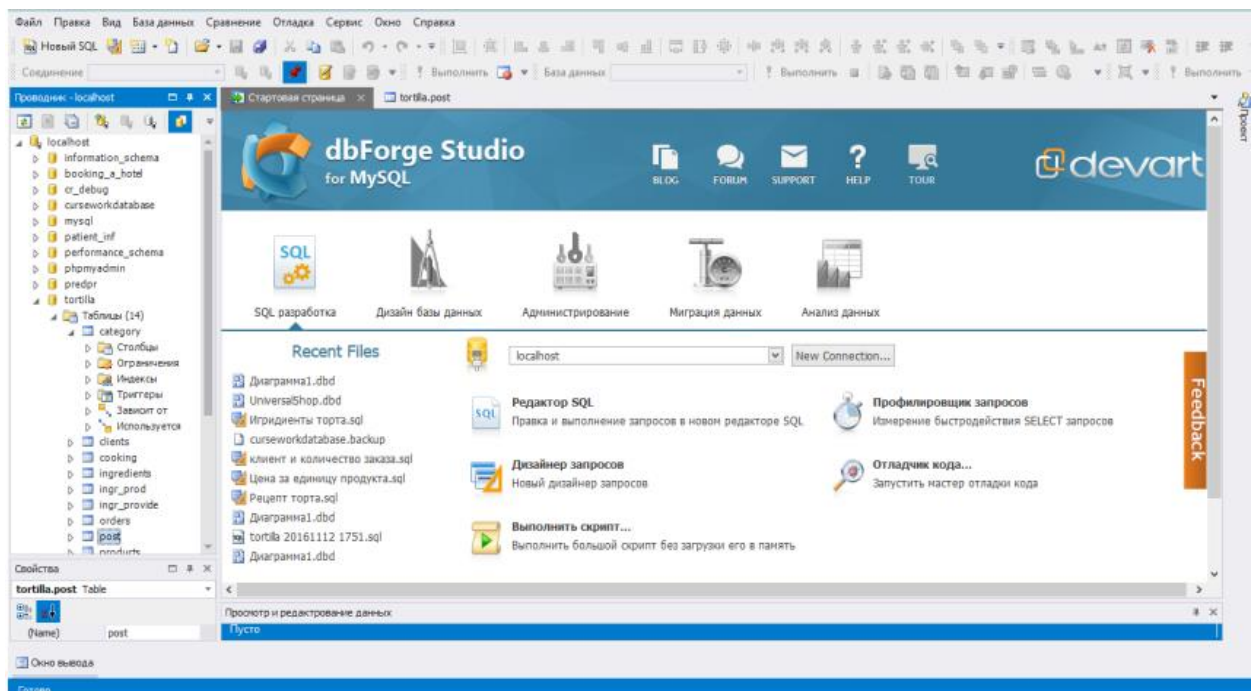


Рис. 2.6. Стартова сторінка програми dbForge Studio for MySQL

Open Server (Рис. 2.7.) – це портативна WAMP (Windows, Apache, MySQL і PHP) платформа, потужний програмний комплекс серверного програмного забезпечення, створений з участю бажаючих багатьох веб-розробників. Основним передбачуваним даним додатком є розробка,

тестування та відвідання практично будь-яких веб-пропозицій, а також для представлення веб-сервісів у локальних сетях.



Рис. 2.7. Офіційний сайт Open Server

Повна збірка серверної платформи з Apache, Nginx, PHP, MySQL включає в себе портативні програми для роботи з графікою, офісними додатками та різними системними програмами.

3.3. Проектування бази даних

Для створення бази даних системи моніторингу правопорушень використовувалась вільна система керування реляційними базами даних MySQL за допомогою програми db Forge Studio for MySQL. Вдала розробка бази даних забезпечує простоту її підтримки.

Перша таблиця, яка була створена, називається “users”, вона має такі поля:

id_user – має тип INT(11), оскільки у цьому полі можливий лише цілий без знаку номер, тут зберігається ID користувача;

login – тип VARCHAR(255), логін для входу в систему;

name – тип VARCHAR(255), оскільки у тут зберігаються дані у вигляді рядків, повне ім'я та прізвище користувача;

phonenumber – тип INT(13), номер мобільного телефону користувача;

email – VARCHAR(50), email клієнта, працює як нік при реєстрації;

password – VARCHAR(50), пароль для авторизації на сайті.

“marks” є таблицею поміток на карті та інформацією про ці помітки, які надають користувачі сайту:

id_mark– тип INT(11), ID помітки на карті;

map_positon– VARCHAR(255), позиція помітки;

text – VARCHAR(255), опис місця помітки;

time – тип DATETIME, час, коли була зроблена помітка;

id_user – тип INT(11), одиниці виміру, які беруться з таблиці “ users”.

Таблиця “admin” немає ніяких звязків, але вона необхідна для створення адміністративної частини сайту:

id_admin – INT(11) , ID категорії продукту;

login – VARCHAR(255), назва категорії.

password – VARCHAR(50), пароль для авторизації на адміністративній панелі.

Таблиці та зв'язки в даній базі даних можливо зручно та наглядно візуалізувати за допомогою діаграми (Рис. 2.8.).

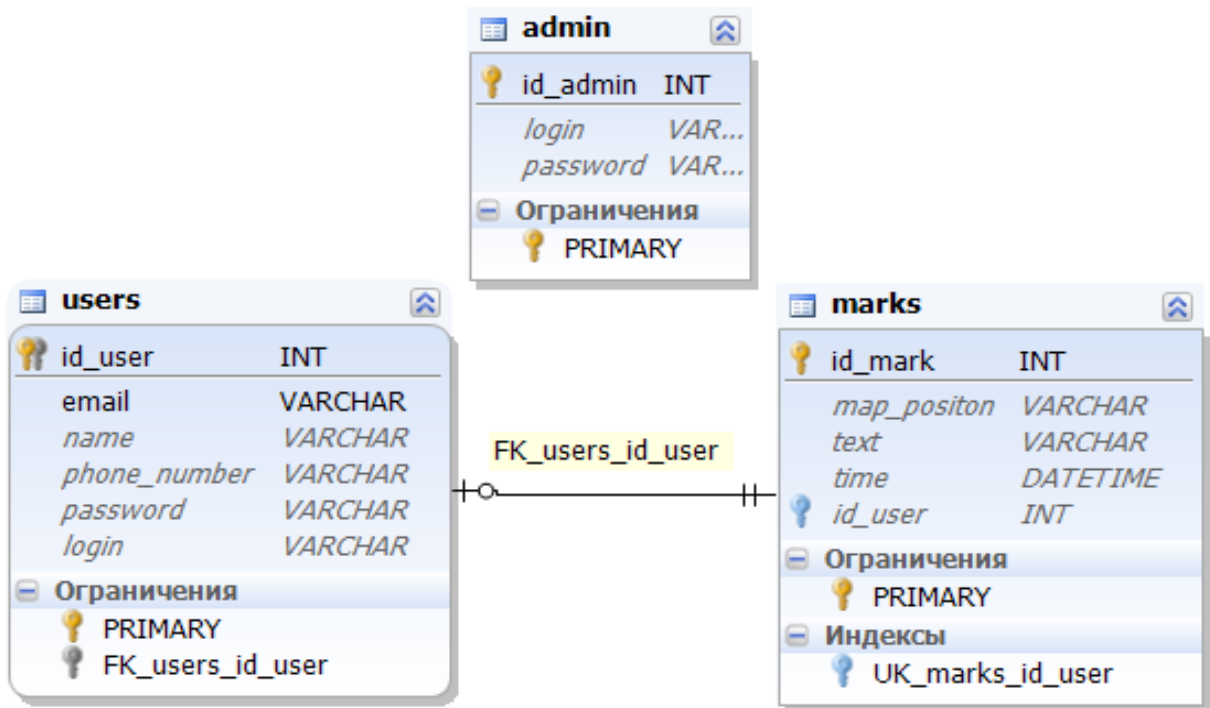


Рис. 2.8. Діаграма бази даних системи моніторингу правопорушень

ВИСНОВКИ ДО РОЗДІЛУ 2

Результати проведеного дослідження дають підставу зробити такі висновки: веб-розробка потребує визначення конкретних цілей та підбір інструментів, які зручні розробнику. Тому для створення системи моніторингу правопорушень обрано такі інструменти:

Adobe Photoshop – графічний реактор для створення дизайну сайту.

Creative Cloud Extract – веб-додаток для верстки сайту з PSD, а саме кольори, шрифти, каскадні таблиці стилів і ресурси, а потім їх можна використовувати при написанні коду для веб-дизайну.

HTML – мова розмітки гіпертексту, є стандартною мовою розмітки для документів, які призначені для відображення у веб-браузері.

Bootstrap – клієнтський фреймворк та набір інструментів з відкритим кодом, який містить в собі шаблони HTML та CSS.

JavaScript – вбудована мова для програмного доступу до об'єктів додатків.

API Google Maps – API, за допомогою якого можна вставити Карти Google на зовнішній веб-сайт, на якому можуть бути накладені дані, пов'язані з сайтом

Django – безкоштовний і вільний фреймворк, тобто набір компонентів, які допомагають розробляти веб-сайти для веб-додатків, написаний на Python.

Python – для роботи з Django, а саме серверно частини сайту для взаємодії сайту з базами даних, являє собою інтерпретовану мову програмування високого рівня.

SQL – мова структурованих запитів, декларативна мова програмування, яка використовується в програмуванні і призначена для управління даними, що зберігаються в системі керування базами даних, або для обробки потоку в реляційній системі управління бази даних (СКБД).

dbForge Studio for MySQL – програмне забезпечення, універсальний інструмент для роботи з MySQL сервером, який дозволяє створювати і

виконувати запити MySQL, розробляти і налагоджувати процедури і функції, а також автоматизувати управління об'єктами баз даних MySQL в зручному інтерфейсі.

Open Server – це портативна WAMP (Windows, Apache, MySQL і PHP) платформа, потужний програмний комплекс серверного програмного забезпечення, для коректного тестування веб-сайтів.

Був обраний тип сайту, а саме сайт-портал, який містить в собі багато функцій та підходить для сайтів з соціальною направленістю. Була побудована логічна структура сайту, яка дозволить оптимально відобразити потреби відвідувачів. Тип логічної структури сайту являє собою «дерево» – це ієрархічна структура, яка дозволяє відвідувачам сайту управляти глибиною відвідування сайту. Користувачі можуть заходити тільки на сторінки верхніх рівнів ієрархії або ж «спуститися» до нижніх рівнів.

Для ефективного зберігання та обробки інформації обрано реляційну базу даних через те, що така база даних може містити в собі елементи даних, які організовані у вигляді набору описаних таблиць, з яких дані можуть бути доступними або повторно зібрані багатьма різними способами без необхідності реорганізації таблиць бази даних. Структура бази даних включає в себе три таблиці, а саме таблиця з реєстраційними даними адміністраторів, таблиця з даними користувачів сайту та пов'язана з цією таблицею відміток на карті, які створюють користувачі.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Налаштування фреймворку Django

Для початку необхідно встановити останню версію Python. На даний момент це версія Python 3.8.0. Також необхідно надати їй права адміністратора для подальшого встановлення необхідних бібліотек. Python 3.8.0 знаходиться у вільному доступі на сайті розробника.

На відміну від більшості систем та сервісів Unix, система Windows не включає встановлену системою установку Python. Щоб зробити Python доступним, команда CPython склала інсталятори Windows (пакетів MSI) з кожним релізом протягом багатьох років. Ці встановлювачі призначені, перш за все, для додавання інсталяції Python для кожного користувача, при цьому основний інтерпретатор і бібліотека використовуються одним користувачем. Інсталятор можна також встановлювати для всіх користувачів окремої машини, а окремий ZIP-файл доступний для локальних дистрибутивів додатків.

Як зазначено в PEP 11, випуск Python підтримує лише платформу Windows, тоді як корпорація Майкрософт розглядає цю платформу за посиленою підтримкою. Це означає, що Python 3.6 підтримує Windows Vista та новіші версії.

Чотири встановлювача Python 3.8.0 доступні для завантаження – по два для 32-розрядних та 64-розрядних версій інтерпретатора. Веб – установник невеликий початкового завантаження, і він буде автоматично завантажувати необхідні компоненти по мірі необхідності. Інсталятор включає компоненти, необхідні для установки за замовчуванням і вимагає тільки підключення до Інтернету для додаткових функцій. Після запуску програми встановлення можна вибрати один із двох варіантів установки (Рис. 3.1.).



Рис. 3.1. Програма інсталятора Python 3.8.0

При виборі «Встановити зараз» будуть встановлені такі функції:

- Python буде встановлено у каталог користувача;
- Програма Python Launcher для Windows буде встановлена відповідно до опції в нижній частині вікна інсталятора;
- Буде встановлено стандартну бібліотеку, тестовий набір, пускову установку та pip;
- Якщо це буде обрано, каталог встановлення буде додано до PATH;
- Ярлики будуть доступні лише для поточного користувача.

Щоб запустити Python зручно з командного рядка необхідно змінити деякі змінні середовища за замовчуванням у Windows. Хоча інсталятор надає можливість налаштувати змінні PATH та PATHNEXT, це надійно для єдиної загальносистемної установки.

Windows дозволяє змінювати змінні середовища як на рівні користувача, так і на рівні системи або тимчасово в командному рядку.

Щоб тимчасово встановити змінні середовища, потрібно відкритите командний рядок і необхідно скористатися командою set :

```
C:\>set PATH=C:\Program Files\Python 3.6.0;%PATH%
C:\>set PYTHONPATH=%PYTHONPATH%;C:\My_python_lib
C:\>python
```

Ці зміни застосовуватимуться до будь-яких подальших команд, виконаних на цій консолі, і будуть успадковані будь-якими додатками, запущеними з консолі.

Включення імені змінної в процентні знаки розшириться до існуючого значення, що дозволить вам додати ваше нове значення в початковому або кінцевому вигляді. Модифікація PATH додавання каталогу, що містить python.exe до початку, є поширеним способом забезпечення правильної версії Python.

Щоб остаточно змінити змінні середовища за замовчуванням, натисніть кнопку Пуск і знайдіть «Змінити змінні середовища» або потрібно відкритите «Властивості системи», Додаткові параметри системи та натисніть кнопку Змінні середовища. У цьому діалоговому вікні ви можете додати або змінити користувацькі та системні змінні. Щоб змінити Системні змінні, потрібно мати необмежений доступ до вашої машини (тобто права адміністратора).

Windows буде об'єднувати змінні користувача після системних змін, які можуть призвести до неочікуваних результатів при змін PATH.

The PYTHONPATH змінна використовується всіма версіями Python 2 і Python 3, тому не треба постійно налаштовувати цю змінну, якщо вона не включає тільки код, сумісний з усіма встановленими версіями Python.

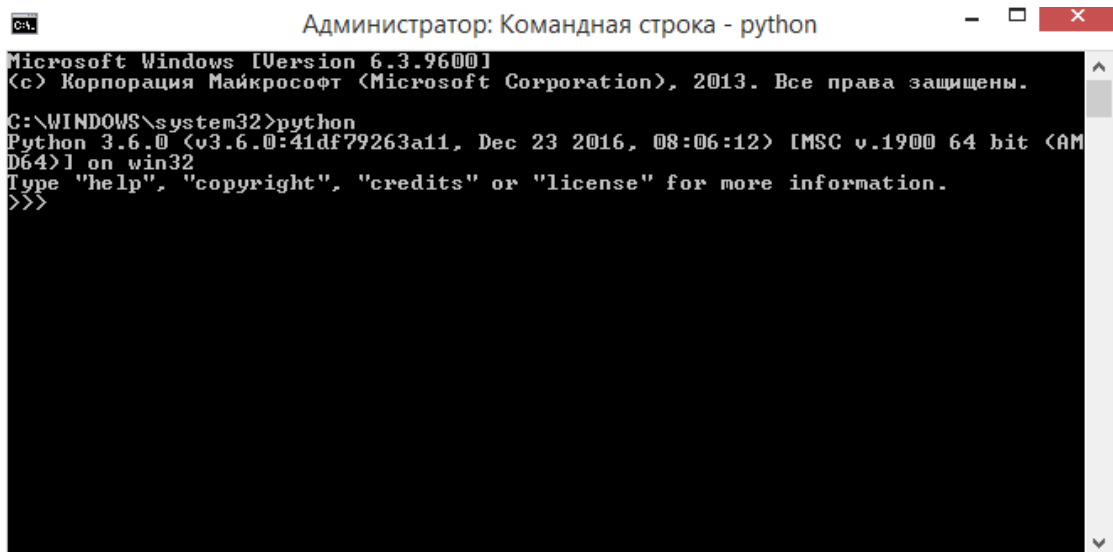
Окрім використання автоматично створеного пункту меню для інтерпретатора Python буде необхідність запуску Python у командному рядку.

На першій сторінці встановлювача може бути обраний варіант, що має назву «Додавання Python до PATH», щоб інсталятор додавав місце установки в PATH. Місце розташування Scripts\папки також додано. Це дає змогу вводити python для запуску інтерпретатора та pip для встановлення пакета. Таким чином, ви також можете виконувати сценарії за допомогою параметрів командного рядка.

Якщо не включати цей параметр під час встановлення, завжди можна запуснути програму встановлення, вибрати «Змінити та увімкнути його». Крім того, можна вручну змінити PATH використовуючи директиви в Excursus: Налаштування змінних середовища.

```
C:\WINDOWS\system32;C:\WINDOWS;C:\Program  
Files\Python 3.8.0
```

Тепер, коли встановлено Python, необхідно відкрити командний рядок і переконатися, що Python доступний, набравши (Рис. 3.3.). Потім виконати наступне, щоб встановити.python – version.



```
Администратор: Командная строка - python  
Microsoft Windows [Version 6.3.9600]  
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.  
C:\WINDOWS\system32>python  
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

Рис. 3.3. Перевірка доступу до Python через консоль

Далі необхідно переконатися, що є наявні останні `pip` та `wheel`. Для цього необхідно ввести цю команду:

```
python -m pip install --upgrade pip wheel
setuptools
```

Встановити `virtualenv` та `virtualenvwrapper`

`virtualenv` та `virtualenvwrapper` забезпечують спеціальні умови для кожного створеного проекту Django. Хоча це не є обов'язковим, це вважається найкращою практикою та заощадить час у майбутньому. В консолі виконати команду:

```
python -m pip install virtualenvwrapper-win
```

Потім створюється віртуальне середовище для проекту:

```
...\> mkvirtualenv myproject
```

Віртуальне середовище буде активовано автоматично, і можна обійти «(мій проект)» поруч із командним рядком. Якщо запускатися новий командний рядок, потрібно буде знову активувати середовище, використовуючи:

```
...\> workon myproject
```

Django можна легко встановити за допомогою `pip`. У командному рядку необхідно переконатися, що віртуальне середовище активне, і виконати таку команду:

```
python -m pip install Django
```

Це завантажить та встановить останню версію Django.

Після завершення інсталяції можна перевірити установку Django, виконавши командний рядок.

```
django-admin --version
```

Якщо `django-admin` відображається лише довідковий текст незалежно від того, як змінюється робочий проект, можливо, існує проблема з асоціацією файлів у Windows. Далі необхідно перевірити, чи існує декілька змінних середовища для запуску сценаріїв Python PATH. Зазвичай це відбувається, коли встановлено більше однієї версії Python.

Якщо відбувається підключення до Інтернету за допомогою проксі, можуть виникнути проблеми із запуском команди. Тоді необхідно встановити змінні середовища для конфігурації проксі в командному рядку наступним чином:

```
py -m pip install Django
```

```
...|> set http_proxy=http://username:password@proxyserver:proxyport
```

```
...|> set https_proxy=https://username:password@proxyserver:proxyport
```

3.2. Налаштування серверної частини сайту

Перший крок – створюється новий проект Django. По суті, це означає, що необхідно запустити кілька стандартних скриптів Django, які створять скелет проекту. Це просто купа каталогів і файлів, які будуть використовуватись пізніше.

Назви цих каталогів і файлів дуже важливі для Django. Їх не можна переименувати. Переміщати їх в інше місце теж не найбільш вдала ідея.

Django необхідна певна структура, щоб мати можливість знайти важливі речі [27].

Обовязково потрібно запускати всі команди в `virtualenv`. Якщо немає в командному рядку префікса (`myenv`), то необхідно активувати `virtualenv`. Для цього потрібно набирати `myenv\Scripts\activate` в Window.

`django-admin.py` – це скрипт, який створює необхідну структуру директорій і файлів. Тепер у проекті повинна бути наступна структура:

```

djangogirls
├── manage.py
├── mysite
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   └── __init__.py
└── requirements.txt

```

`manage.py` – це інший скрипт, який допомагає з управлінням сайтом. За допомогою нього існує можливість запуску веб-сервера на комп'ютері без установки додаткових програм при необхідності.

Файл `settings.py` містить налаштування для веб-сайту, а файл `urls.py` містить список шаблонів, за якими орієнтується `urlresolver`.

Далі необхідно внести зміни в `mysite/settings.py`. Також потрібно встановити коректний часовий пояс на сайті. Перейти до списку часових поясів Вікіпедії та скопіювати назву свого часового поясу (TZ) (наприклад, `Europe/Kiyev`).

У файлі `settings.py` знайти рядок, що містить `TIME_ZONE`, і зміни її відповідно до свого часового поясу:

```
mysite / settings.py
```

```
TIME_ZONE = 'Europe/Kiyev'
```

Код мови складається з скороченої назви мови, наприклад en для англійської або ua для української, і коду країни. Це знадобиться, якщо потрібно, щоб всі вбудовані кнопки і повідомлення від Django були на необхідній мові. Таким чином, напис на кнопці “Cancel” буде переведена на заданий мові. Django підтримує великий набір готових перекладів .

Змінити мову, відредагувавши наступний рядок:

```
mysite / settings.py
```

```
LANGUAGE_CODE = 'ua-ua'
```

Також необхідно додати в налаштування інформацію про розташування статичних файлів (наприклад CSS). Спустися в кінець файлу і після змінної STATIC_URL додати нову - STATIC_ROOT:

```
mysite / settings.py
```

```
STATIC_URL = '/static/'
```

```
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

Коли налаштування DEBUG має значення True, а ALLOWED_HOSTS порожня, ім'я хосту веб-сайту звіряється зі списком ['localhost', '127.0.0.1', ':::1']. Жодне із значень не збігається з назвою хоста на PythonAnywhere при публікації нового застосування, тому необхідно змінити таке налаштування:

```
ysite / settings.py
```

```
ALLOWED_HOSTS = ['127.0.0.1', '.pythonanywhere.com']
```

Налаштування бази даних

Існує безліч різних баз даних, які можуть зберігати дані для сайту. Ми будемо використовувати стандартну – `sqlite3`.

Вона вже обрана за замовчуванням в файлі `mysite/settings.py`:

```
mysite / settings.py
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Щоб створюється базу даних потрібно набрати в командному рядку наступне:

```
python manage.py migrate
```

Якщо все пройшло успішно, то буде наступний результат:

```
command-line
```

```
(myvenv) ~/djangogirls$ python manage.py migrate
Operations to perform:
  Apply all migrations: auth, admin, contenttypes,
sessions
Running migrations:
  Rendering model states... DONE
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
```



```
Applying admin.0002_logentry_remove_auto_add...OK
Applying
contenttypes.0002_remove_content_type_name... OK
Applying
auth.0002_alter_permission_name_max_length... OK
Applying
auth.0003_alter_user_email_max_length...OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null...OK
Applying auth.0006_require_contenttypes_0002...OK
Applying
auth.0007_alter_validators_add_error_messages... OK
Applying sessions.0001_initial... OK
```

Запуск веб-сервера

Необхідно знаходитись в директорії, де розташований файл `manage.py`.

Запуск веб-сервер з командного рядка: `python manage.py runserver`:

command-line

```
(myvenv) ~/django$ python manage.py runserver
```

Якщо виника помилка `UnicodeDecodeError`, необхідно використати іну команду:

command-line

```
(myvenv)~/django$ python manage.py runserver 0:8000
```

Тепер необхідно перевірити, чи працює веб-сайт – відкрити браузер (Firefox, Chrome, Safari, Internet Explorer або будь-який інший). Вводиться наступна адреса: <http://127.0.0.1:8000/> Спрацювало (Рис. 3.4.)!

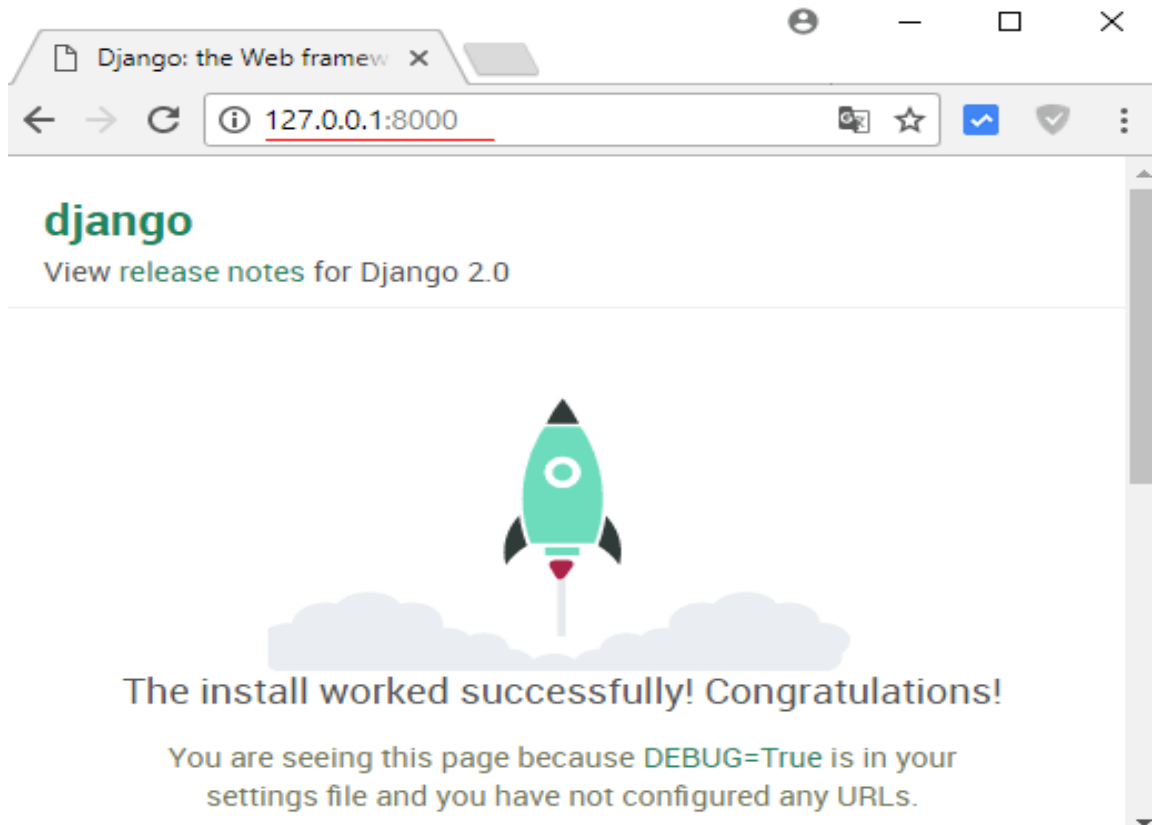


Рис. 3.4. Сторінка за адресою якого запусканий проект Django

Поки працює веб-сервер, в терміналі не буде запрошення для введення команд. Можна ввести текст, але не можна виконати ніяку іншу команду. Це відбувається тому, що сервер продовжує роботу, «слухаючи» вхідні запити.

Веб-сервер займе командний рядок, поки його не зупинити. Щоб і далі мати можливість набирати команди, потрібно потрібно відкрити ще одне вікно терміналу і активувати в ньому віртуальне оточення. Щоб зупинити веб-сервер, необхідно перейти назад у вікно, в якому він працює, і натиснути CTRL + C – кнопки Control і C разом (в Windows може знадобитися натиснути клавіші Ctrl + Break).

3.2. Реалізація клієнтської частини сайту

Однією з найпотужніших частин Django є автоматичний інтерфейс адміністратора. Він читає метадані з створених моделей, щоб забезпечити швидкий, орієнтований на модель інтерфейс, де довірені користувачі можуть керувати вмістом на сайті. Функції адміністратора обмежуються внутрішнім інструментом управління організацією.

Створюваний сайт обов'язково буде містити в собі панель управління адміністратора. Щоб додавати, редагувати і видаляти записи на сайті, буде використовуватись панель управління адміністратора Django. Відкриваємо файл `blog/admin.py` та замінити його вміст на:

```
blog / admin.py
```

```
from django.contrib import admin
from .models import Post
admin.site.register(Post)
```

Таким чином відбувся імпорт (увімкнення) моделі `Post`. Щоб модель стала доступна на сторінці адміністрування, потрібно її зареєструвати її за допомогою `admin.site.register(Post)`.

Тепер треба поглянути на модель `Post`. Обов'язково запуснути веб-сервер командою `python manage.py runserver`. Перейти в браузер і набираємо адресу <http://127.0.0.1:8000/admin/>. Створюється сторінка авторизації (Рис.3.5) [23].

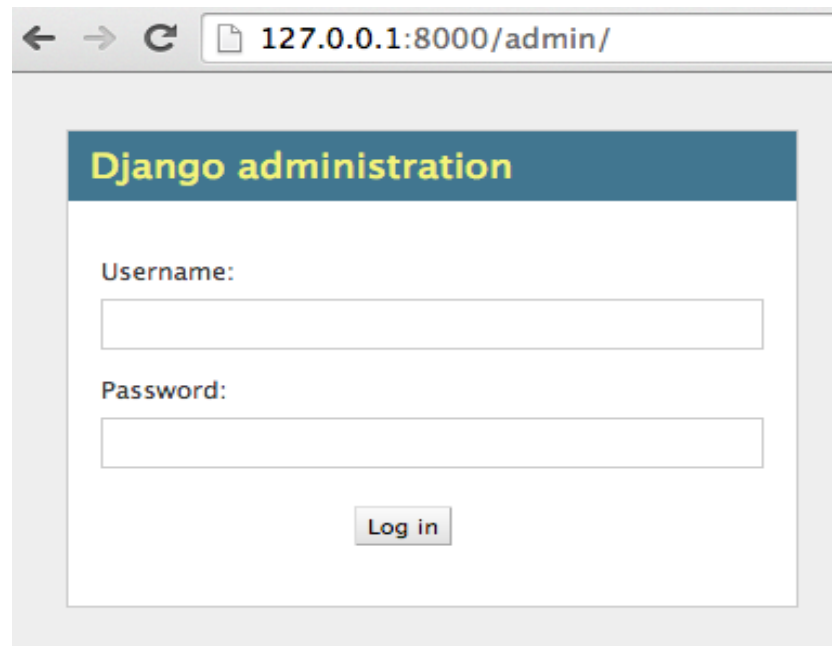


Рис.3.5. Сторінка авторизації на панелі управління адміністратора

Щоб увійти до системи, спочатку необхідно створити суперкористувача (англ. Superuser) - користувача, який має повний доступ до управління сайтом. Повертаємось до командного рядка і набираємо `python manage.py createsuperuser`, і натиснути Enter.

Щоб виконувати команди під час роботи сервера, потрібно відкрити нове вікно терміналу і активувати в ньому віртуальне оточення.

У відповідь на запит ввести ім'я користувача (малими літерами, без пробілів), адресу електронної пошти та пароль. Якщо пароль не з'являється на екрані під час введення, так і задумано. Просто надрукувати його і натиснути Enter, щоб продовжити. Результат повинен виглядати наступним чином (ім'я користувача і пошта):

command-line

```
(myvenv) ~/djangogirls$ python manage.py
createsuperuser
Username: admin
Email address: admin@admin.com
```

Password:

Password (again):

Superuser created successfully.

Повернутися в браузер і увійти в систему за допомогою імені користувача та пароля. Відбудеться вхід в панель управління Django (Рис.3.6).

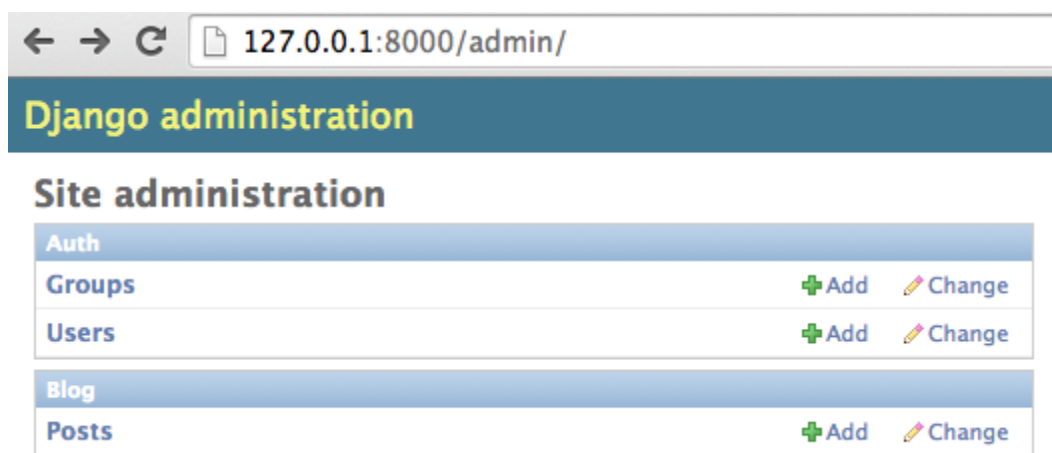


Рис.3.6. Панель управління адміністратора Django

Сайт буде містити в собі блог. Усі шаблони сайту зберігаються в директорії `blog/templates/blog`. Для початку створюється директорію `templates` всередині папки `blog`. Потім створюється інша папка `blog` всередині папки `templates`:

blog

└── templates

└── blog

Тепер створюється файл `post_list.html` усередині директорії `blog/templates/blog`.

Якщо виникла помилка `TemplateDoesNotExist`, потрібно спробувати перезапустити сервер. Перейти до командного рядка, зупини веб-сервер натисненням `Ctrl + C` (`Ctrl` і `C` одночасно) і запустити його знову командою `python manage.py runserver`

Далі додається наступний код в файл шаблону окрім стандартної «шапки»:

```
blog / templates / blog / post_list.html
```

```
<style>
```

```
body {
```

```
    min-width: auto;}
```

```
body #u94 #google-container-u94 {
```

```
    position: relative;
```

```
    width: 100%; розміри вікна в браузері
```

```
    height: 100%;
```

```
    background-color: #e7eaf0;} колір фону сторінки
```

```
</style>
```

Найбільш базовий тег, `<html>`, завжди присутній на початку веб-сторінки, а `</html>`- в кінці. Основний контент веб-сайту знаходиться між тегами `<html>` і `</html>`.

`<p>` є тегом для параграфів; `</p>`, Відповідно, закриває кожен параграф.

Кожна HTML-сторінка також ділиться на два елементи: `head` і `body` .

`Head` – це елемент, який містить інформацію про документ, яка не відображається на екрані.

`Body` – це елемент, який містить все, що буде відображено на веб-сторінці.

Використовуємо тег `<head>`, щоб повідомити браузеру про налаштування сторінки, і тег `<body>`- про її вміст.

API сторонніх організацій – це API, що надаються третіми сторонами - як правило, такими компаніями, як Facebook, Twitter або Google - щоб була можливість отримати доступ до їх функціональності через JavaScript та використовувати його на своєму сайті. Один з найбільш очевидних прикладів - використання API-карти для відображення спеціальних карт на ваших сторінках.

Далі пояснено, як автентифікувати додаток у API JavaScript Maps за допомогою власного ключа API.

Необхідно виконати наступні дії, щоб отримати ключ API:

1. Перейти на консоль Cloud Cloud Platform[20].
2. Створити або вибрати раніше створений проект.
3. Натиснути Продовжити, щоб увімкнути API та будь-які пов'язані з цим сервіси.
4. На Credentials сторінки, отримати ключ API (і встановити обмеження ключових API). Якщо є існуючий необмежений ключ API або ключ із обмеженнями браузера, можна використовувати цей ключ.

5. Щоб запобігти крадіжці квоти та захистити ключ API, необхідно його захистити. Така функція знаходиться на цьому ж сайті.
6. Увімкнути виставлення рахунків натиснувши пункт Обмеження використання.
7. Замінити значення key-параметра в URL-адресі своїм власним ключем API [20].

```
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
</script>
```

Необхідно зберігти цей файл з ім'ям, яке закінчується .html. Завантажити HTML-файл у веб-браузер, перетягнувши його зі свого робочого столу на свій браузер. Крім того, подвійне клацання файлу працює в більшості операційних систем.

API сторонніх виробників, розташовані на сторонніх серверах. Щоб отримати доступ до них з JavaScript, спочатку потрібно підключитися до функціональності API та зробити доступним на вашій сторінці. Зазвичай це перше посилання на бібліотеку JavaScript, доступну на сервері через <script>елемент:

```
<script src="https://api.mqcdn.com/sdk/googlemap-
js/v1.3.2/googlemap.js"></script>
```

```
<link type="text/css" rel="stylesheet"
href="https://api.google.com/sdk/googlemap-
js/v1.3.2/googlemap.css"/>
```


Потім можна почати використовувати об'єкти, наявні в цій бібліотеці:

```
let L;  
  
let map = L.googlemap.map('map', {  
  
  center: [53.480759, -2.242631],  
  
  layers: L.Google Map.tileLayer('map'),  
  
  zoom: 12});
```

Тут створюється змінна для зберігання інформації карти, потім створюється нова карта за допомогою `googlemap.map()` методу, який приймає за свої параметри ідентифікатор `<div>` елемента, в якому потрібно відобразити карту ('map'), і об'єкт параметрів, що містить деталі конкретної карти, яку ми хочемо показати. У цьому випадку ми визначаємо координати центру карти, тип картографічного шару карти для відображення (створений `Google Map.tileLayer()` методом) та рівень масштабування за замовчуванням.

Це вся інформація, яка потрібна API Google Map для побудови простої карти. Сервер, до якого підключається, обробляє всі складні речі, як-от відображення правильних плиток карти для області, яка відображається, тощо.

Деякі API-файли обробляють доступ до своєї функціональності дещо по-різному, вимагаючи від розробника запит HTTP на певний шаблон URL-адреси для отримання даних. Вони називаються RESTful API .

Захист для ключа API, як правило, полягає в тому, щоб користувач знав, що відбувається на веб-сайтах, які вони відвідують, і менше шансів стати жертвою того, хто використовує API зловмисним чином.

Сторонні API мають дещо іншу систему дозволів - вони, як правило, використовують ключі розробника, щоб дозволити розробникам доступ до функцій API, що більше захищає постачальника API, ніж користувача:

```
L.googlemap.key = 'YOUR-API-KEY-HERE';
```

Цей рядок визначає API або ключ розробника, який потрібно використовувати у d програмі – розробник програми повинен подати заявку, щоб отримати ключ, а потім включити його у свій код, щоб отримати доступ до функціональності API. Потрібний ключ дозволяє постачальнику API притягувати користувачів API до відповідальності за свої дії. Коли розробник зареєструвався для ключа, він буде відомий постачальнику API, і можна вжити заходів, якщо вони почнуть робити щось шкідливе з API (наприклад, відстежувати місцезположення людей або намагатися спамувати API із завантаженням запитів на припинить його роботу, наприклад) [14].

Усі ці дії виглядають так:

```
<div id="map_canvas"></div>

<script>

function initMap() {

    var myOptions = {

        zoom: 8,

        center: {lat: 53.480759, lng: -2.242631},
```

```
mapTypeId: google.maps.MapTypeId.TERRAIN,

disableDefaultUI: true,
```

Налаштування відображени елементів керування:

```
zoomControl: true,

mapTypeControl: true,

scaleControl: true}

var map = new
google.maps.Map(document.getElementById("map_canvas"),
myOptions);
```

Додати спеціальний значок:

```
var iconBase =
'https://maps.google.com/mapfiles/kml/shapes/';

var marker = new google.maps.Marker({

position: {lat: 53.480759, lng: -2.242631},

icon: iconBase + 'flag_maps.png',

map: map });
```

Додати вікно інформації:

```
var contentString = '<div id="content"><h2
id="firstHeading" class="firstHeading">Custom info
window</h2><p>This is a cool custom info
window.</p></div>';
```

```

var infowindow = new google.maps.InfoWindow({
    content: contentString });

marker.addListener('click', function() {
    infowindow.open(map, marker);});}

```

```
</script>
```

```

<script
src="https://maps.googleapis.com/maps/api/js?key=ENTER-
API-KEY-HERE&callback=initMap"

```

```
async defer></script>
```

Додавання карти з поміченим маркером:

```
<div id="map"></div>
```

```
<script>
```

```

function initMap() {

    var uluru = {lat: -25.344, lng: 131.036};

    var map = new google.maps.Map(
        document.getElementById('map'), {zoom: 4,
center: uluru});

    var marker = new google.maps.Marker({position:
uluru, map: map});

} </script>

```

```
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=YOUR_A
PI_KEY&callback=initMap">
```

В наведеному нижче коді завантажуються API із вказаної URL-адреси:

```
<script async defer

src="https://maps.googleapis.com/maps/api/js?key=YOUR_A
PI_KEY&callback=initMap">
</script>
```

У наведеному вище коді callback параметр виконує initMap функцію після завантаження API. async атрибут дозволяє браузеру продовжувати надавати іншу частину вашої сторінки під час завантаження API. key параметр містить ключ API [29].

Код нижче містить initMap функцію, яка ініціалізує та додає карту під час завантаження веб-сторінки. Використовуйте scriptтег, щоб включити власний JavaScript, який містить initMap функцію.

```
function initMap()
```

Далі необхідно додати getElementById() функцію пошуку карти на веб-сторінці.

Код нижче створює новий об'єкт карт Google і додає на карту властивості, включаючи центр та рівень масштабування.

```
{ var uluru = {lat: -25.344, lng: 131.036};
```

```

var map = new
google.maps.Map(document.getElementById('map'), {

    zoom: 4,

    center: uluru });}

```

У наведеному вище коді `new google.maps.Map()` створюється новий об'єкт карт Google. `Center` властивість вказує API де знаходиться карта на сторінці сайту.

Координати карти встановлюються в порядку: широта , довгота .

The `zoom` властивість задає масштаб для карти. Масштаб: 0 - найменший масштаб і відображає всю землю.

Код нижче ставить маркер на карті. `position` властивість визначає положення маркерів.

```

var marker = new google.maps.Marker({

    position: uluru,

    map: map});}

```

Layer Google Maps Data надає контейнер для довільних просторових даних (в тому числі GeoJSON). Якщо ваші дані знаходяться у файлі, розміщеному на тому самому домені, що і додаток API API Maps, можна завантажити його за допомогою `map.data.loadGeoJson()` методу. Файл повинен бути на одному домені, але його не можливо розмістити в іншому піддомені.

```

map.data.loadGeoJson('data.json');

```

Запит JSONP. Для використання цієї методики цільовий домен повинен підтримувати запити для JSONP. Робити запити JSONP від доменів, які не знаходяться під контролем адміністратора, дуже ризиковано. Оскільки браузер завантажує будь-який код, який повертається як сценарій, слід запитувати JSONP лише з перевіреного домену. Взагалі JSONP замінюється CORS; останнє є набагато безпечнішими і повинно бути кращим вибором, якщо вони є в наявності.

Щоб подати запит на JSONP, використовується, createElement() щоб додати scriptтег до голови документа.

```
var script = document.createElement('script');
script.src =
'http://earthquake.usgs.gov/earthquakes/feed/v1.0/summa
ry/2.5_week.geojsonp';
document.getElementsByTagName('head')[0].appendChild(sc
ript);
```

Коли сценарій запускається, цільовий домен передає дані в якості аргументу іншому сценарію, зазвичай іменованому callback(). Цільовий домен визначає ім'я сценарію зворотного дзвінка, яке є іменем на сторінці, коли ви завантажуєте цільову URL-адресу в браузері.

addGeoJson() метод для розміщення проаналізованих даних GeoJSON на карті. Необхідно визначити сценарій зворотного дзвінка у своєму кодї:

```
function eqfeed_callback(response) {
    map.data.addGeoJson(response); }
```

Локалізацію карти можна налаштувати для конкретної країни чи регіону такими способами:

- Змінити налаштування мови за замовчуванням.
- Вказати код регіону, який змінює поведінку карти на основі певної країни чи території.

За замовчуванням API JavaScript Maps використовує налаштування користувачевої мови, визначеного в браузері, під час відображення текстової інформації, такої як назви елементів керування, повідомлення про авторські права, вказівки щодо руху та мітки на картах. У більшості випадків бажано дотримуватися налаштувань браузера.

Якщо необхідно, щоб API JavaScript Maps ігнорував налаштування мови браузера, його змусити відображати інформацію певною мовою, додавши languageпараметр до <script>тегу під час завантаження API API Maps Maps. Language параметр впливає на імена елементів управління, повідомлення про авторські права, напрямки руху і контролю маркерів, а також відповіді на запити обслуговування. Наприклад, коли геокодування адрес вулиці повертається, назва країни повертається мовою, яку запитували.

```
<script  
src="https://maps.googleapis.com/maps/api/js?key=YOUR_A  
PI_KEY&language=ja&region=JP">  
</script>
```

3.3. Інструкція користувача

Першим що бачать відвідувачі сайту kryvbassocial.com (Рис. 3.7.) – це карта міста Кривого Рогу та відмітки на ній. Також в хедері знаходиться назва сайту та динамічне меню.

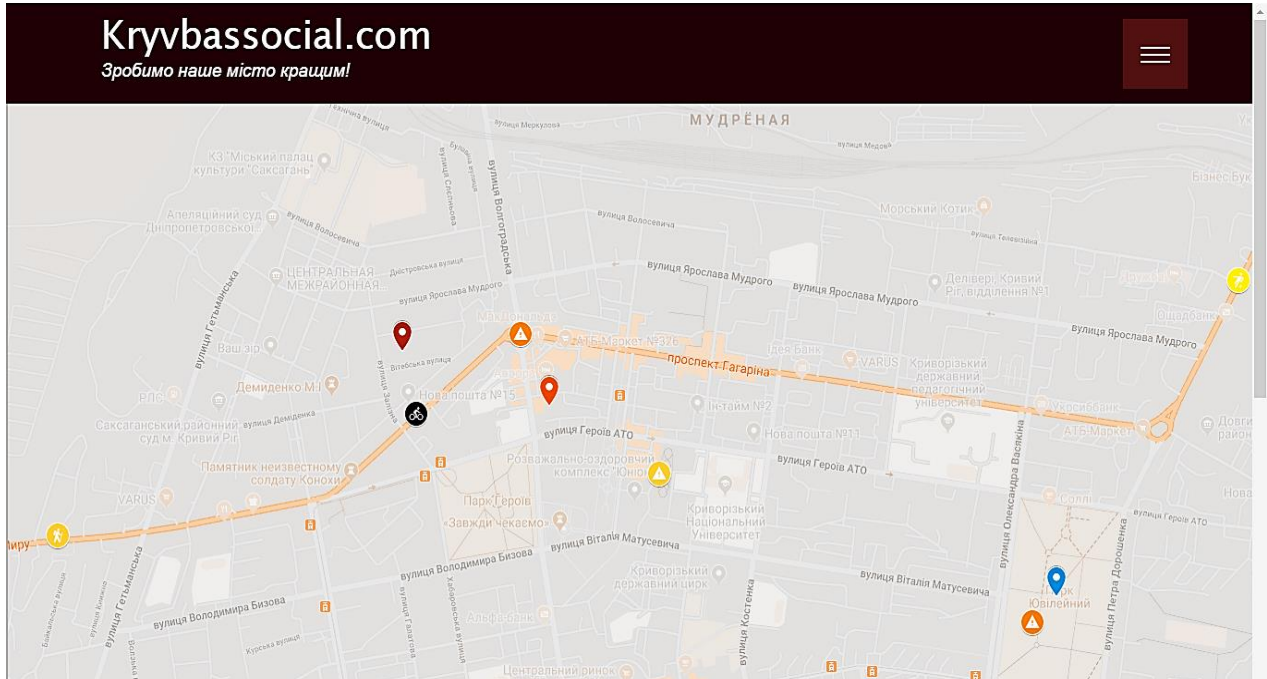


Рис. 3.7. Головна сторінка сайту kryvbassocial.com

Розгорнувши меню сайту (Рис. 3.8.), можна побачити такі пункти як «реєстрація», «вхід», «зробити помітку про правопорушення», «блог» та «зворотній з'язок».

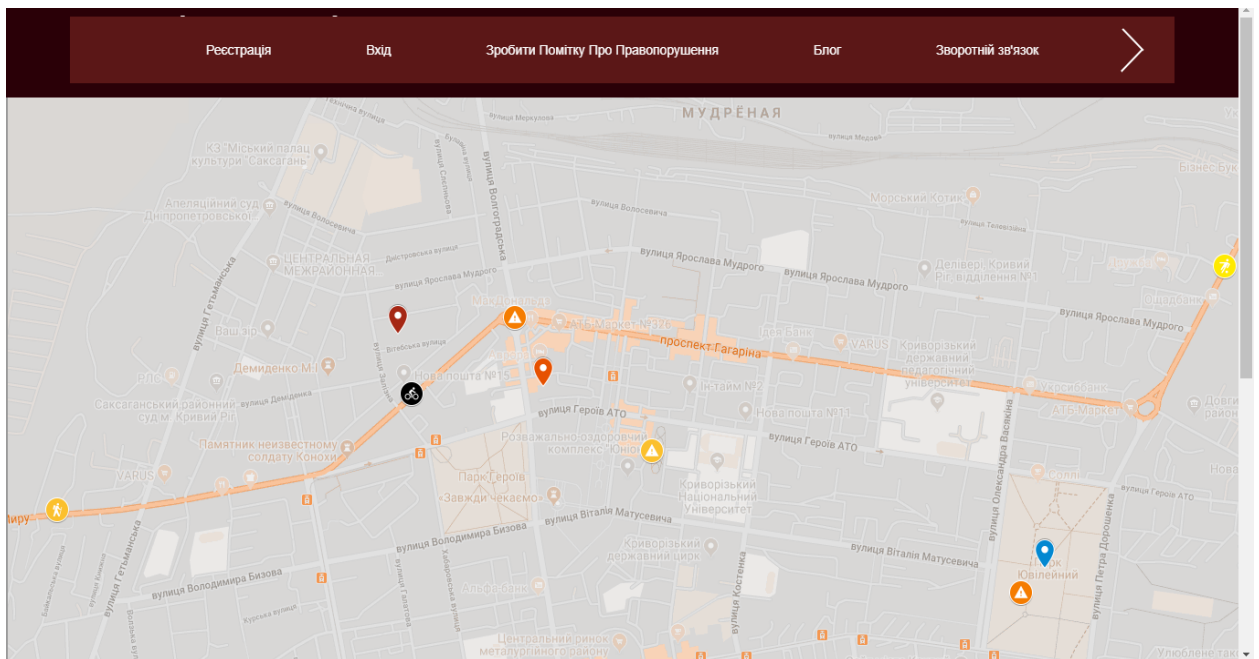


Рис. 3.8. Розгорнуте меню сайту

Натиснувши пункт «вхід» (Рис. 3.9.) користувач може заповнити рядки пошти та пароллю увійти в систему сайта, щоб мати можливість робити відмітки про правопорушення на карті.

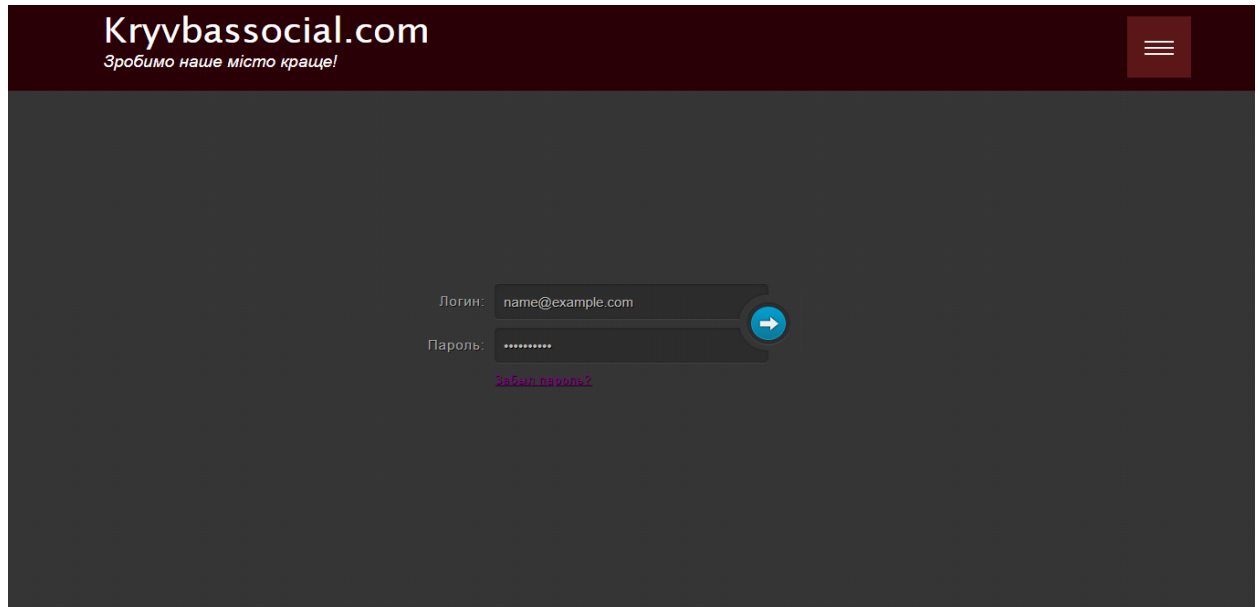


Рис. 3.8. Сторінка авторизації користувача

Після входу в систему є можливість робити відмітки на карті. Для цього потрібно натиснути на обране місце на карті і тоді з'явиться вікно (Рис. 3.9.) для надання інформації про це місце. Інформація складається з назви відмітки, тексту та фото.

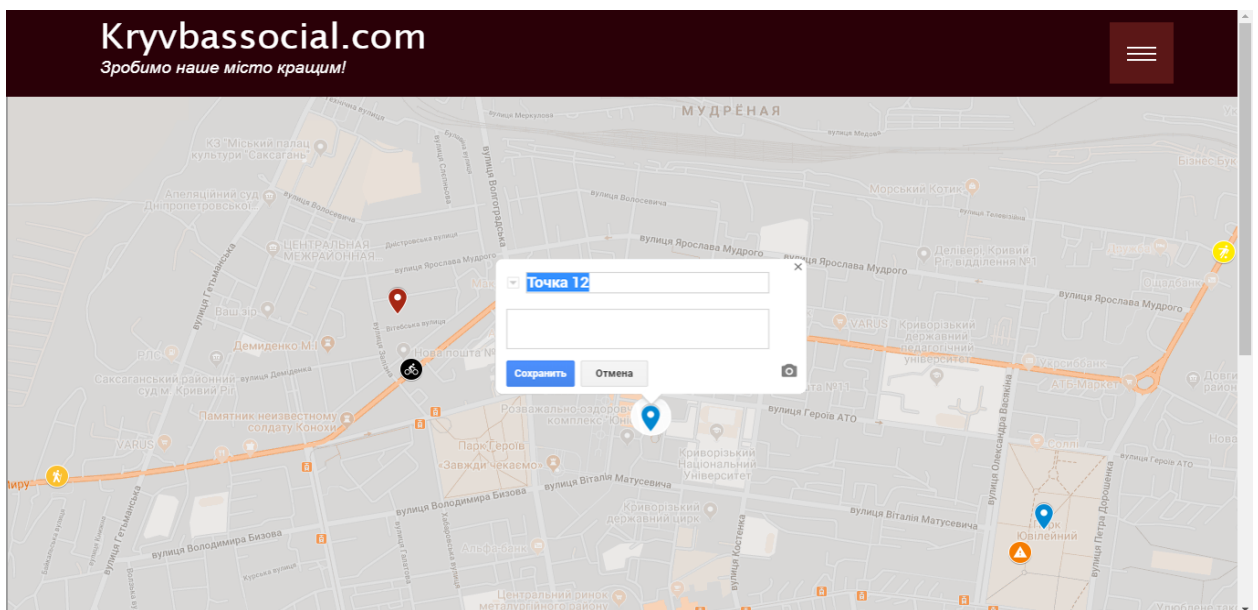


Рис. 3.9. Вімітка та вікно редагування інформації про обране місце

Після заповнення текстових полів та їх збереження за допомогою кнопки «зберігти» є можливість змінити зображення мітки та його колір (Рис.3.10.).

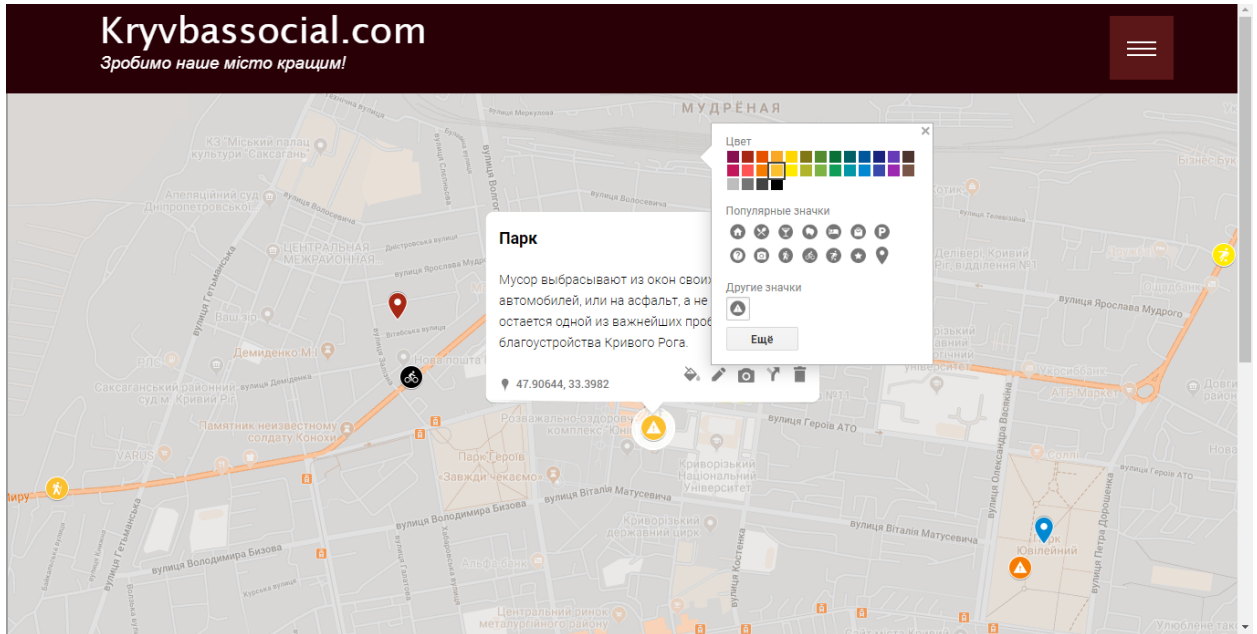


Рис. 3.10. Редагування зображення та кольору відмітки

Натиснувши на «додати зображення» з'явиться вікно форми (Рис.3.10.), за допомогою якого можна завантажити фото з пристрою, за допомогою веб-посилання або веб-камери. Також можна перенести фото у відповідне вікно.

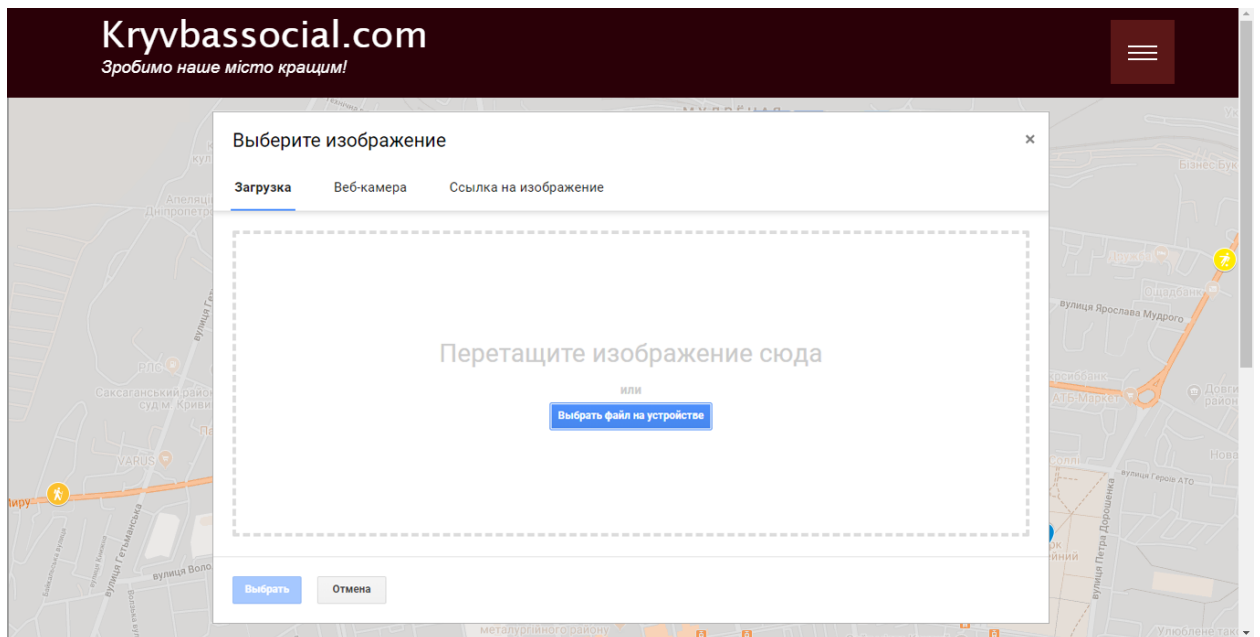


Рис. 3.10. Вікно для завантаження фото

Після повного заповнення інформацією вікна форми (Рис.3.11.) необхідно натиснути кнопку «зберегти» або кнопку «відмінити», щоб не додавати інформацію про мітку (тоді сама мітка не збережеться).

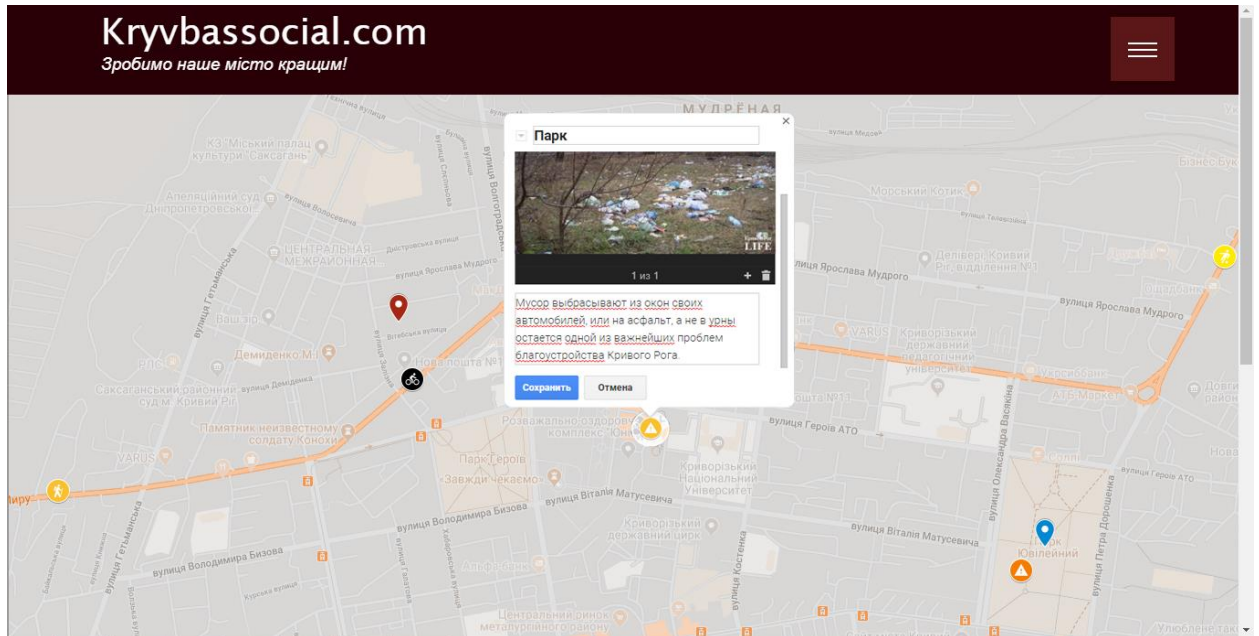


Рис.3.11. Повністю заповнене вікно форми редагування інформації про відмітку

Після зберігання відмітки з'явиться на сайті (Рис. 3.12.). Інформацію про цю відмітку можна переглянути, натиснувши на неї на карті.

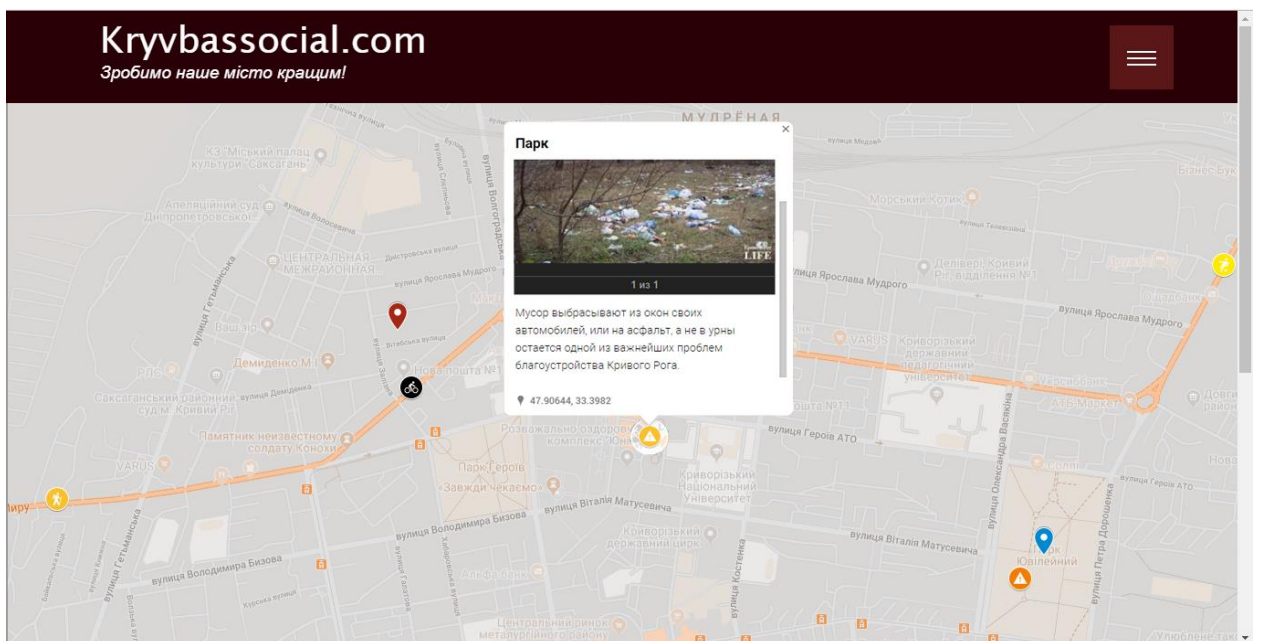


Рис. 3.12. Перегляд інформації створеної відмітки на карті

ВИСНОВКИ ДО РОЗДІЛУ 3

Отримані результати після створення системи моніторингу правопорушень дають підставу зробити такі висновки:

- Було реалізовано системи моніторингу правопорушень у вигляді сайту за допомогою фреймворку Django. Карта була створена на базі Google Maps, за допомогою API Google Maps який надає доступ до функцій розробника Google Maps.
- В ході тестування, що проводилося за допомогою локального серверу Open Server та Django, було виявлено помилки та усунено їх.
- Одним із недоліків є те, що існує небезпека поширення спаму від користувачів і це потребує більш ретельної розробки інформаційної безпеки сайту.
- Також до незручностей можна віднести те, що не було можливості правильно організувати пошук по карті, щоб користувач мав змогу швидше знайти необхідне йому місце. Цей недолік буде усуватис майбутньому.

ВИСНОВКИ

Результати проведеної роботи дають підставу зробити такі висновки:

Було проаналізовано таке явище, як соціальна відповідальність, взаємодія індивіа з суспільством, державою, оточенням, тощо. Було виявлено причини правопорушень в суспільстві та розглянуто один із інструментів в допомозі в боротьбі з ними, а саме соціальні геоінформаційні системи, які виступають інструментом комунікації населення, адміністрації, громадських діячів для запобігання негативних тенденцій розвитку суспільства на певній території.

Оцінено переваги та недоліки різних геоінформаційних систем, досвід їх використання та актуальність. Загалом такі системи популярні в США, а в Україні такі проекти не розвивалися в достатній мірі. Також було проаналізовано та виконано порівняння цифрових картографічних інструментів, які можуть бути використані для створення таких систем.

Була виконана постановка задачі, яка полягала в створенні системи моніторингу правопорушень міста Кривий Ріг у вигляді сайту-порталу.

Обрано інструменти розробки сайту, спираючись на їх актуальність та зручність для розробника.

Створені макети сайту та виконана його верстка, налаштований сервер та адміністративна панель, створений інтерфейс користувача та налаштована карта для додавання відміток про правопорушення, розроблена база даних для зберігання інформації про користувачів та мітки на карті, які вони залишають.

Проведено тестування створеного сайту, виявлено помилки та усунуено їх, оцінено неоліки та шляхи удосконалення інформаційної системи у майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гоббс Т. Левіафан, або Суть, будова і повноваження держави церковної та цивільної / Томас Гоббс. – Київ: Дух і Літера, 2000. – 600 с. – Назва з екрана.
2. Карта злочинності Львова [Електронний ресурс]. – 2015. – Режим доступу: <https://zaxid.net/projects/crime/?type=killing>. – Назва з екрана.
3. Кобзін Д. "Теорія розбитих вікон" під іншим кутом [Електронний ресурс] / Денис Кобзін. – 2018. – Режим доступу: https://dt.ua/SOCIUM/teoriya-rozbitih-vikon-pid-inshim-kutom-295016_.html. – Назва з екрана.
4. Матузов Н. И. Социальная ответственность: понятие и виды [Електронний ресурс] / Н. И. Матузов, А. В. Малько // Теория государства и права: Учебник. – 2004. – Режим доступу: <https://jurisprudence.club/teoriya-gosudarstva-prava-uchebnik/sotsialnaya-otvetstvennost-ponyatie.html>. – Назва з екрана.
5. Мінченко О. Онлайн-карта криміногенної ситуації в Україні виграла «SocialBoost: Open Data» [Електронний ресурс] / Ольга Мінченко. – 2013. – Режим доступу: <http://watcher.com.ua/2013/04/16/onlayn-karta-kryminohennoyi-sytuatsiyi-v-ukrayini-vyhrala-socialboost-open-data/>. – Назва з екрана.
6. Онлайн-карта криміногенності Києва [Електронний ресурс] – Режим доступу: zloch.in.ua. – Назва з екрана.
7. Основи JavaScript [Електронний ресурс]. – 2019. – Режим доступу: https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics. – Назва з екрана.

8. Петренко А. Розповідаємо, як створити структуру сайту для успішного просування [Електронний ресурс] / Андрій Петренко. – 2016. – Режим доступу: <https://uagolos.com/rozpovidajemo-iaak-stvoriti-stryktyru-saity-dlia-yspishnogo-prosyvannia-saity/>. – Назва з екрана.
9. Способи використання веб-технологій сучасними підприємствами [Електронний ресурс]. – 2014. – Режим доступу: <http://uastudent.com/sposoby-vykorystannja-veb-tehnologij-suchasnymu-ridpryjemstvu/>. – Назва з екрана.
10. Створення веб порталу [Електронний ресурс] – Режим доступу: <https://www.wdesign.net.ua/ua/web-portal.html>. – Назва з екрана.
11. Що таке HTML? [Електронний ресурс]. – 2019. – Режим доступу: https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/HTML_basics. – Назва з екрана.
12. Adding a Google Map with a Marker to Your Website [Електронний ресурс] – Режим доступу: <https://developers.google.com/maps/documentation/javascript/adding-a-google-map>. – Назва з екрана.
13. Apocalypse Survival Map [Електронний ресурс] – Режим доступу: <https://mapofthedead.com/v1/>. – Назва з екрана.
14. Client-side web APIs [Електронний ресурс] // Learn web development. – 2019. – Режим доступу: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs. – Назва з екрана.
15. Compstat and Organizational Change: A National Assessment [Електронний ресурс] – Режим доступу: <https://www.policefoundation.org/projects/compstat-and-organizational-change-a-national-assessment/>. – Назва з екрана.
16. Crimereports [Електронний ресурс]. – 2019. – Режим доступу: <https://www.crimereports.com/>. – Назва з екрана.

17. Dealing with files [Электронный ресурс] // Learn web development. – 2019. – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Dealing_with_files. – Назва з екрана.
18. Documentation. Python Setup and Usage. Using Python on Windows [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3/using/windows.html>. – Назва з екрана.
19. DOMINGUEZ J. Greensboro police, apartment manager thwart repeat armed robbery [Электронный ресурс] / JOE DOMINGUEZ. – 2015. – Режим доступа: <https://myfox8.com/2015/10/22/greensboro-police-apartment-manager-thwart-repeat-armed-robbery/>.
20. Google Cloud Platform Console. [Электронный ресурс] – Режим доступа: <https://console.cloud.google.com/google/maps-api/overview>. – Назва з екрана.
21. Google Maps – OpenLayers [Электронный ресурс] – Режим доступа: <https://github.com/mapgears/ol3-google-maps>. – Назва з екрана.
22. Importing Data into Maps [Электронный ресурс] – Режим доступа: https://developers.google.com/maps/documentation/javascript/importing_data. – Назва з екрана.
23. Introducing the Django Admin [Электронный ресурс] – Режим доступа: <https://docs.djangoproject.com/en/3.0/intro/tutorial02/#introducing-the-django-admin>. – Назва з екрана.
24. Jogy J. How I Understood: Building Interactive Maps using Python, Leaflet.js and Folium [Электронный ресурс] / Joel Jogy. – 2019. – Режим доступа: <https://codeburst.io/how-i-understood-displaying-interactive-maps-using-python-leaflet-js-and-folium-bd9b98c26e0e>. – Назва з екрана.

25. Karolczak L. Custom HTML Markers with Google Maps Lee Karolczak [Электронный ресурс] / Lee Karolczak. – 2014. – Режим доступа: <https://humaan.com/blog/custom-html-markers-google-maps/>.– Назва з екрана.
26. LexisNexis [Электронный ресурс] – Режим доступа: <https://www.lexisnexis.com/en-us/gateway.page>.
27. Map of Life [Электронный ресурс] – Режим доступа: <https://mol.org/>.– Назва з екрана.
28. Map: Social Media Monitoring by Police Departments, Cities, and Counties [Электронный ресурс] // Brennan Center for Justice. – 2019. – Режим доступа: <https://www.brennancenter.org/our-work/research-reports/map-social-media-monitoring-police-departments-cities-and-counties>. – Назва з екрана.
29. Marks S. Adding Numbers or Letters to Google Maps API Markers [Электронный ресурс] / Steve Marks. – 2011. – Режим доступа: <http://biostall.com/adding-number-or-letters-to-google-maps-api-markers/>.– Назва з екрана.
30. New York City Complaints Map [Электронный ресурс] – Режим доступа: <http://chriswhong.github.io/soda-leaflet/>.– Назва з екрана.
31. Over 100 New Streetside and 3D Cities Go Live on Bing Maps [Электронный ресурс]. – 2014. – Режим доступа: <https://blogs.bing.com/maps/2014/08/20/over-100-new-streetside-and-3d-cities-go-live-on-bing-maps/>.– Назва з екрана.
32. Predictive policing: the brave new age of law enforcement analytics [Электронный ресурс]. – 2014. – Режим доступа: <https://sctr7.com/2014/09/05/predictive-policing-the-brave-new-age-of-law-enforcement-analytics/>.– Назва з екрана.
33. SpotCrime.com [Электронный ресурс] – Режим доступа: <http://spotcrime.com/>.– Назва з екрана.

34. SQL [Электронный ресурс] – Режим доступа:
<http://progopedia.ru/language/sql/>.– Назва з екрана.

35. Tetsudo Now [Электронный ресурс] – Режим доступа:
<http://www.demap.info/tetsudonow/>.– Назва з екрана.

36. TOPS OF 2014: DIGITAL [Электронный ресурс] // Nielsen. –
2014. – Режим доступа:
<https://www.nielsen.com/us/en/insights/article/2014/tops-of-2014-digital/>.–
Назва з екрана.

37. Writing your first Django app, part 2 [Электронный ресурс]. –
2019. – Режим доступа:
<https://docs.djangoproject.com/en/3.0/intro/tutorial02/>.– Назва з екрана.