Final Project – Technical Report

Group 6 - CrimeStats.LA

Allen Ho
allho@iu.edu

Ashwin Venkatakrishnan
ashvenk@iu.edu

Blake Farmer
blfarmer@iu.edu

Application URL

We have hosted our web application on Vercel (PaaS) which allows continuous deployment by integrating with a GitHub repository. Once we push our commits, Vercel will build and deploy our application, providing free hosting forever.

Website: https://crimestats-la.vercel.app/

LAPD Login Credentials

Our website includes an internal dashboard that is meant to be used by LAPD staff to read, update, and delete incident reports. To access this page, you can use the following credentials that we have setup for testing:

- Username: csla.admin
- Password: cslaAdmin123!

GitHub URL

If you are facing permission-related issues with our repository, please contact us as soon as possible.

Repository: https://github.iu.edu/D532-Group6/crimestats-la

Project Summary

Our web application focuses on using a crime dataset from Los Angeles Open Data that contains details of reported crime incidents from various areas dating back to 2020, our web application is the one location for visitors and residents of Los Angeles (LA) to view insights into the crime statistics of different areas of LA that can help them make informed decisions about visiting, buying homes, opening businesses, finding good school districts, and more in LA.

Project Objectives and Usefulness

Our primary objectives with this project are to:

1. Provide the latest crime statistics for LA county using data from a reliable source on a user-friendly website.
2. Allow civilians and police officers to report incidents from anywhere using our website.

3. Provide an easy-to-use dashboard for LAPD staff to manage incident records.
4. Use a modern technology stack that is easy to maintain and scale.

One of the objectives we originally planned to achieve over this semester with this website was to overlay the crime data on a map of LA county using Google Maps API to provide an intuitive user experience. Using location attributes from the dataset and functions provided by MongoDB, features such as heatmaps would have been a great visualization for users. These features could not be developed in time, but they demonstrate the future scope of this project.

Technical Description

Tools

- Database: MongoDB (Atlas)
- Backend: MongoDB Atlas Functions and HTTPS Endpoints (JavaScript)
- Frontend: Next.js (JavaScript)
- Build, Deployment and Hosting: Vercel
- Collaboration: GitHub (IU), Microsoft 365 (IU)

Data

Leveraging an existing crime dataset published by the Los Angeles Police Department (LAPD) on the Open Data platform as our source of truth, we built our website using MongoDB, Atlas Functions and HTTPS Endpoints, and Next.js.

Dataset: https://data.lacity.org/Public-Safety/Crime-Data-from-2020-to-Present/2nrs-mtv8/about_data

MongoDB Atlas is a great platform for a project like ours as it provides flexibility and enables rapid prototyping during the initial stages to experiment with our database design and API functions. We began our project by using a Jupyter notebook on Google Colaboratory to define a schema using Mongoengine, a popular Python Object-Document Mapping (ODM) library. Mongoengine allowed us to turn the JSON schema we designed into Python code using the Schema and Model classes. The dataset, in CSV files, was then loaded using Pandas before using the Mongoengine models to create objects and store in MongoDB. The complete dataset contains over 915,000 documents. We truncated this to 2000 documents and created our database. We had to truncate because of the time it took to load the documents. Given our normalized data model, each document required multiple iterations to find the respective referenced documents from other collections before it could be saved. Our Jupyter notebook took approximately 45 minutes to load the truncated 2000 documents.

The following table provides an overview of the collections and fields stored in our database. Here, the incident numbers are a 9-digit code assigned by LAPD, while the date reported for an incident is automatically saved when a user submits the form on our website. Area, Modus Operandi, Premise, Weapon, Status and Crime IDs are 3-digit codes based on an existing list of codes maintained by the LAPD. As the public is often unfamiliar with the codes used by the LAPD, new incident reports will only require the user to choose the description from a drop-down menu while the respective code will be assigned by our website by referring to the respective collection in our database.

Table 1. Variable Descriptions and Constraints

	Variable	Description	Constraint
Area Information	**area**	**LAPD area codes**	**Int, Required**
	area_name	**Area name**	**String, Required**
Modus Operandi Information	**code**	**LAPD MO Codes**	**Int, Required**
	desc	**MO description**	**String, Required**
Premise Information	**code**	**LAPD Premise codes**	**Int, Required**
	desc	**Premise description**	**String, Required**
Status Information	**code**	**LAPD Incident Status codes**	**String, Required**
	desc	**Status description**	**String, Required**
Weapon Information	**code**	**LAPD Weapon codes**	**Int, Required**
	desc	**Weapon description**	**String, Required**
Incident Information	DR_NO	Incident Number (Primary Key)	9- Digit Code, Required
	Date_Rptd	Date Reported	DateTime, Required
	Date OCC	Date Occurred	DateTime, Required
	Time OCC	Time Occurred	Int, Required
	Location	Generated Location ID based on Area	Location Document Reference
	Victim	Generated Victim Code	Victim Document Reference
	Crime	Generated Crime Code	Crime Document Reference
	Premise		Premise Document Reference
	Weapon		Weapon Document Reference
	Status		Status Document Reference
	Modus Operandi		Modus Operandi Document Reference
Victim Information	Age	Victim Age	3 – Digit Age Limit
	Sex	Victim Sex	M, F, or X Drop Down
	Descent	Victim Descent	A,B,H,O,W, or X
Location Information	Location	Street Address	Address Format
	Area	Area Information	Area Document Reference
	Cross Street	Cross Street Address	Address Format
	Lat	Latitude	Geographical Latitude
	Lon	Longitude	Geographical Longitude
Crime Information	code	LAPD Crime Codes	Int
	desc	Crime description	String Text

The next step was to develop a way to access these documents from our front-end service. Instead of following the traditional approach of developing an API/back-end service for our website, we decided to explore MongoDB Atlas Functions and HTTPS endpoints. Atlas functions are written in JavaScript where any Node.js dependency may be imported and used, just as you would in a local development environment. For our project, we did not require any additional dependencies to be imported in our Atlas

environment. We developed 17 functions on Atlas with an HTTPS endpoint for each of them that works for the HTTP methods - GET, POST, PATCH and DELETE to enable CRUD operations.

In user interface submitted data, we have expanded the information provided to the dataset. At a minimum an incident date, information surrounding the victim, an area, and the crime type must be submitted. Age will be a free form submission. Victim sex and descent information will be selected from drop down menus. For gender, the user can submit Male (M), Female (F), or Unknown (X). For descent, the existing database used Asian (A), Black (B), Hispanic (H), Other (O), White (W), or Non-Binary (X). However, we have expanded descent to include a more diverse range.

Location information will be user submitted as free text. An optional field for cross street and address can be submitted as well as free form text. A pre-determined list of area and location premises can be submitted from a dropdown menu. Based on the user submitted address, geographical location will be queried for longitude and latitude. Crime details can be submitted by the user through various dropdown menus. The crime type, modus operandi (MO), and if a weapon was used can be selected. The user can also look to submit the initial status of the investigation.

Functionalities

Create

- Users (civilians and/or officers) can report an incident with some mandatory fields such as date & time occurred, area, and victim details.

Read

- Users can click on the following preset visualizations, which are based on the number of incident reports, for a better view and understanding of the crime statistics:
    - Top 5 Areas
    - Top 5 Crimes
    - Top 5 Premises
    - Top 5 Weapons
    - Victim Gender Distribution
    - Victim Descent Distribution

Update

- LAPD officials will be able to view incident reports and update any field except the date when the incident was reported. Our current approach to updating data is very flexible since constraints can be added as required. Using LAPD login credentials, an officer can manage all documents in the database. To update an incident report, a “secret” is required to be sent in the request, without which the operation would not proceed. This is to ensure that such requests originate from authorized users only.

Delete

- LAPD users can also reject/delete incident reports from civilians or delete existing records as well. This is very flexible now as well, but additional constraints can be added as required. To delete an incident report, a “secret” is required to be sent in the request, without which the operation would not proceed. This is to ensure that such requests originate from authorized users only.

Issues

Due to the normalized nature of our NoSQL schema, the initial step of loading data took a very long time in the first try for our whole dataset of 915,000 documents. After 4 hours, we decided to stop the data loading process and updated our notebook to truncate to 2000 records instead. The normalization also made it challenging to work with the data in our web application. We store document references in almost every document and as such, when we had to fetch an incident document to our web application with all the incident details populated, we had to write an aggregation pipeline with lookups, unwinds, grouping and projections. In hindsight, some amount of denormalization would have made it easier for us to work with the fetched data in the front-end.

Group Contribution

Name	Tasks	Average Time Spent
Allen Ho	Finding a dataset Queries for preset visuals Endpoint integration Project demo video editing Project Report	12 hours each
Ashwin Venkatakrishnan	Database design Web App Development Atlas Functions and Endpoints Web App Build and Deployment Project Report	
Blake Farmer	Database design Queries for preset visuals Endpoint integration Project Report	