

CrimeVista: AI-Powered Geospatial Crime Intelligence and Predictive Platform

Mrs. Prachi Jain, Kaustubh Saurkar, Kunal Jain, Sampada Singh Bundel, Om Chandekar,
Vedant Durge, Sakshi Oza

Department of Computer Science and Engineering, G H Rasoni University, Amravati, Maharashtra, India

Abstract: Crime records usually live in spreadsheets, and every comparison between periods, areas or offence types means setting the same filters again by hand. CrimeVista puts that work in one browser application. An administrator uploads a CSV file; the server checks and normalises each row and stores the valid ones in a relational database, and the dashboard, maps, trend charts, area profiles and downloadable reports all read from the result. The client is written in React and TypeScript, the service in FastAPI with SQLAlchemy, and a separate in-browser fallback takes over when the API cannot be reached. We describe the whole path from upload to report and test it on a bundled example of 16,000 rows for Nagpur, covering January 2022 to June 2025. The rows are labelled synthetic demonstration data, so every figure we report describes the sample and none of it is official crime statistics. In the sample every row has a coordinate pair, victim age and gender are missing from 31.6% of rows, and weapon details from 88.3%. CrimeVista calculates counts and comparisons only. It has no live incident feed, no predictive model and no risk score, and its area ranking orders areas by record count. Seven backend tests pass and the frontend production build completes with a warning about large JavaScript chunks. These checks show that selected code paths run; they say nothing about representativeness, real-world impact or production performance.

Keywords - crime data analytics, geospatial visualisation, decision support, descriptive statistics, data quality, web information systems.

I. INTRODUCTION

A crime record is a small bundle of facts: a date, an offence label, an administrative area and, often, a pair of coordinates. Analysts rarely get them in one place. One spreadsheet holds the incident list, another the monthly summary, and the map is a third tool with its own filters. Each time someone asks what has changed in an area since last quarter, the same date range, offence type and area have to be set again in every tool, and the answers can drift apart.

CrimeVista is our attempt to remove that repetition. It places filtering, descriptive summaries, map inspection and report export behind one web interface, and it keeps a single filter state so that the dashboard, the map, the trend charts, the area profiles and the exported report all describe the same set of records. The application works on whatever dataset is loaded. It does not connect to a live incident feed, and it does not recommend where police should be deployed. The globe on the landing page is only a visual entry point: its routes and markers are illustrative and do not show crime observations.

The central engineering problem is consistency. A user selects dates, offence types, severity labels and areas once and expects to see that selection reflected everywhere. Three requirements follow.

- Shared filter state. One selection drives the record list, the dashboard, the trends, the area ranking, the map and the report request.
- A guarded write path. Anyone who can reach the service may read the views, but only an administrator may replace the dataset or generate reports.
- A safe import. A bad upload must never leave the application without data. Invalid rows are reported, and if no row is valid the existing dataset stays in place.

These are data-access and presentation requirements. Meeting them does not show that a pattern in the output is statistically significant or predictive, and we make no such claim.

Research on crime mapping has shown that the spatial unit, time window, offence category and hotspot method can each change the pattern a reader sees [1], [2]. We do not propose a new spatial statistic. Our contribution is a worked account of one dataset moving through one application, from CSV import to exported report, together with the shared filters and descriptive views in between. We do not claim that existing crime-analysis platforms lack these features.

We set four objectives: (1) document the software and the data as they exist today; (2) explain how the filters and summaries are calculated; (3) check the supplied data and a few software paths; and (4) say plainly what the results can and cannot support.

A. Research Contribution

- An end-to-end account of the data path from a CSV file through validation and database storage to filtered summaries, maps and report downloads, including what the browser does when the API is down (Sections III–VI).

- A profile of the bundled dataset covering its size, date span, label counts and missing fields, with the provenance gap stated (Sections IV and IX).
- A description of the count-based analytics and of how they differ from formal spatial analysis, prediction or a live data service (Sections V and VII).
- The checks we ran, and what they do not show (Section VIII).
- A list of the provenance, privacy and governance work needed before stronger claims could be made (Sections X–XII).

B. Organisation

Section II reviews research on crime concentration and hotspot policing. Sections III to VII follow the application from architecture and data handling through analytics, implementation and interface behaviour. Sections VIII to X report our checks, discuss the findings and cover privacy and ethics. Sections XI and XII give future work and the conclusion; the limitations of each part are stated where they arise.

C. Related Work

A crime map is more than points on a background. How it looks depends on the area unit, the time interval, the offence definition and the mapping method. Chainey, Thompson and Uhlig compared how hotspot methods represent spatial crime patterns [1], and Weisburd argued that crime concentrates at small geographic places [2]. Both are a warning for any tool that ranks areas. A high count can come from a large area, a large population, differences in reporting or the way records were grouped, as well as from more events.

Evaluations of hot-spot policing ask whether an intervention changes outcomes, and a map or a ranked list cannot answer that question on its own [3]. CrimeVista’s Hotspots page sorts areas by the number of matching records. It does not compute spatial intensity, adjust for population or land area, fit a statistical model, forecast incidents or evaluate an intervention. We therefore call its output “count-ranked areas” in this paper and keep the word “hotspot” only where it is the name of an existing page or API route.

We see CrimeVista as an exploration and reporting tool. It combines an incident record, shared filters, relational queries, descriptive summaries, point maps and exports in one workflow. It is not a new geospatial method. We have not compared it with commercial systems, run a user study, or established that the supplied data are complete or representative, so every claim in this paper stays at the level of what the software does.

II. LITERATURE SURVEY

The reviewed literature highlights the growing importance of Artificial Intelligence (AI), Machine Learning (ML), Geographic Information Systems (GIS), and deep learning techniques in crime prediction and prevention. Existing studies demonstrate that machine learning models can identify crime patterns, while GIS-based spatial analysis helps detect crime hotspots and understand their geographical distribution. Furthermore, spatiotemporal models, including Hierarchical Spatiotemporal Transformers and ConvLSTM, integrate temporal trends, geographical locations, and mobility patterns to improve crime forecasting. However, existing approaches may be limited by data quality, regional variations, limited integration of spatial and temporal features, and challenges in translating predictions into practical decision-making. These gaps motivate the development of CrimeVista, a geospatial crime-record exploration and reporting platform that integrates crime data management, count-based area ranking, map visualisation and interactive dashboards. Predictive analytics is treated as future work (Section XI), not as a current feature.

TABLE I
 SUMMARY OF REVIEWED LITERATURE

Sr. No.	Paper Title and its Author	Details of Publication	Findings
1	Crime Forecasting: A Machine Learning and Computer Vision Approach to Crime Prediction and Prevention\nAuthor: Neil Shah, Nandish Bhagat, Manan Shah[cite: 1]	Visual Computing for Industry, Biomedicine, and Art, Springer, Vol. 4\nPublished: 29 April 2021[cite: 1]	Explains how machine learning and computer vision can improve crime forecasting and assist law enforcement agencies.[cite: 1]

2	Artificial Intelligence & Crime Prediction: A Systematic Literature Review\nAuthor: Fatima Dakalbab, Manar Abu Talib,	Social Sciences & Humanities Open, Volume 6\nPublished: 2022, Article 100342[cite: 1]	Reviews 120 papers, identifies 64 ML techniques used in crime prediction, and concludes that supervised learning is
	Omnia Elmutasim et al.[cite: 1]		the most widely adopted approach.[cite: 1]
3	Crime Rate Prediction using Ensemble Machine Learning Algorithms\nAuthor: S. Vijayalakshmi, R. K. Dhanalakshmi, P. Suresh	IEEE Access, Vol. 11\nPublished: 14 March 2023	Demonstrates an ensemble framework combining Random Forest and XGBoost to classify urban crime severity, achieving higher prediction reliability.
4	Graph Neural Networks for Spatial-Temporal Crime Prediction in Smart Cities\nAuthor: Chen Zhang, Lin Liu, Meng Wang et al.	Information Sciences (Elsevier), Vol. 625\nPublished: 10 May 2023	Proposes a SpatialTemporal Graph Convolutional Network (ST-GCN) to capture complex urban road network dependencies for proactive police patrol dispatch planning.
5	GIS-based Crime Hotspot Mapping and its Correlation with Land Use: A Case Study on Assessment of CPTED Principle\nAuthor: Hrutvik K. Sharma, Ravin M. Tailor[cite: 1]	International Social Science Journal (Wiley), Vol. 74, Issue 253\nPublished: 20 November 2023[cite: 1]	Uses GIS, IDW interpolation, and land-use analysis to identify crime hotspots and demonstrates how GIS supports crime prevention and urban planning.[cite: 1]
6	Predictive Policing and Crime Forecasting: A Comparative Study of Time Series and Machine Learning Models\nAuthor: Alex M. Rodriguez, Sarah L. Jenkins	International Journal of Law, Crime and Justice (Elsevier), Vol. 76\nPublished: 18 January 2024	Evaluates ARIMA, Prophet, and LSTM models on multi-year municipal crime datasets, concluding deep learning models better handle seasonal shifts.

7	Predictive Crime Hotspot Detection: A Spatial Analysis Approach.\nAuthor: Suleiman Ibrahim, Puren Jain, Mukesh Bhardwaj[cite: 1]	ICCEWC 2022, LNNS Vol. 1409, Springer Nature\nPublished: 20 Oct 2024[cite: 1]	Presents a spatial analysis approach for predicting crime hotspots using geospatial data and predictive modelling to help allocate resources effectively.[cite: 1]
8	Big Data Analytics and Spatial Data Mining for Urban Crime Pattern Identification\nAuthor: Marcus Vance, Helena Rostova, Kenji Sato	Journal of Big Data (Springer), Vol. 11, Article 45\nPublished: 02 November 2024	Integrates social media feeds, weather patterns, and historical police logs into Spatial DBSCAN to detect dynamic high-risk crime zones.
9	Explainable AI (XAI) in Crime Prediction: Enhancing Transparency in Machine Learning Decision-Making\nAuthor: David K. Miller, Elena Gomez, Priya Patel	Expert Systems with Applications (Elsevier), Vol. 240\nPublished: 12 January 2025	Integrates SHAP and LIME frameworks into tree-based crime forecasting models to eliminate algorithmic bias and increase trust.
10	Spatiotemporal ("Crime prediction") and forecasting using machine learning.\nAuthor: Riyazuddin, Syed Ziaur Rahman, Jaffar Sadiq et al.[cite: 1]	SN Computer Science Springer Vol. 7, Issue 2, Article 213\nPublished: 16 Feb 2025[cite: 1]	Proposed a Hierarchical Spatiotemporal Transformer (HSTT) model that predicts crime by analyzing both location and time with high accuracy.[cite: 1]
11	Deep Learning for Crime Forecasting: The Role of Mobility at Fine-grained Spatiotemporal Scales\nAuthor: Aradhna Albors Zumel, Michele Tizzoni, Gian Maria Campedelli[cite: 1]	Journal of Quantitative Criminology (Springer Nature)\nPublished: 22 September 2025[cite: 1]	Developed a ConvLSTM model integrating crime history, human mobility, and sociodemographic data to predict crime at fine spatial and temporal levels.[cite: 1]

III. SYSTEM OVERVIEW AND ARCHITECTURE

A. Requirements and Design

The application supports four everyday tasks: reviewing summary counts, applying the same filters across views, examining geocoded rows and area profiles, and exporting a report. Filter state is shared by the dashboard, the map, the analytics pages and the report workflow. A typed API boundary keeps server requests explicit, and SQLAlchemy lets one model run on SQLite during local development and on PostgreSQL in the configured production setup [4]– [6]. If the API cannot be reached, the browser loads its configured CSV mirror and calculates the descriptive views locally.

That fallback does not synchronise with the server database. We chose these tools because they fit the application, and we have not measured any performance advantage from them. Table II lists the stack.

TABLE II
 TECHNOLOGY STACK OF CRIMEVISTA

Layer	Technology	Purpose
Client	React, TypeScript, Vite	Page composition, routing, browser build
Client data	React Query, Fetch API	Query state, HTTP requests, browser-side fallback
Charts	Recharts	Dashboard and trend visualisations
Map	MapLibre GL; OpenFreeMap styles and tiles	2D/3D incident visualisation and basemap
API	FastAPI, Pydantic	REST routes, validation, OpenAPI
ORM	SQLAlchemy	Relational persistence and queries
Database	SQLite locally; PostgreSQL configured for production	Incident record storage
Layer	Technology	Purpose
Export	ReportLab, pandas, OpenPyXL	PDF, CSV and XLSX reports
Deployment	Render blueprint	Web service and managed database configuration
Assistant	Optional OpenAI service; keyword fallback	Conversational help; not validated analytics

B. Overall Architecture

In the configured deployment a browser talks to FastAPI, which serves the built React client and reads or updates a relational database through SQLAlchemy (Fig. 1). The administrator's CSV upload is the only server-side dataloading path. If the API probe fails, the frontend switches to a separate in-browser path that fetches a public CSV mirror and calculates selected views itself. This is not a second database, and nothing uploaded there is saved to the server. The analytical map also depends on external basemap services, and the landing globe loads country-boundary data separately. Coordinates for incidents come from the records themselves. In production, authentication protects administrative actions, but the read endpoints are open to any caller who can reach the service.

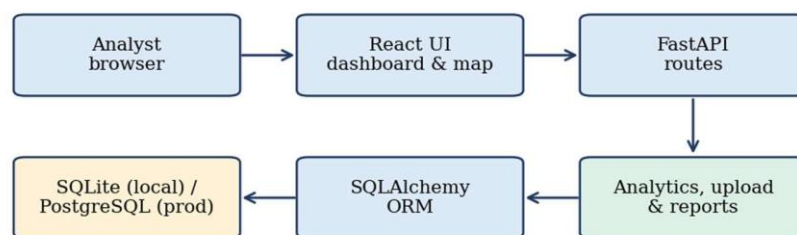


Fig. 1. CrimeVista's deployed components and the paths between browser views, API services, database storage, mapping, reports and administrator actions.

C. Frontend and API

The client uses React and TypeScript [7]. React Router connects the landing page to the analysis pages, and React Query manages the data the API returns. A shared request layer calls the same-origin API by default and selects the browser fallback when the API cannot be reached. FastAPI groups the service into routes for health, login, filters, dashboard summaries, records, analytics, maps, uploads, reports and chat [4]; Table III lists them. The landing globe is a visual introduction, whereas the analytical map draws filtered incident records.

TABLE III
 BACKEND API MODULES

Module	Representative endpoint	Purpose
Health	GET /api/health	Database readiness and last-update status

Authentication	/api/auth/login, /logout, /session	Administrator session lifecycle
Filters	GET /api/filters/options	Populate category and date filters
Dashboard	GET /api/dashboard/overview	KPIs, insights and chart data
Crime records	GET /api/crimes	Paginated filtered rows
Analytics	/api/analytics/trends, /hotspots, /areas/{area}	Temporal, area and category summaries
Map	GET /api/map/incidents	Valid-coordinate incident points
Upload	/api/upload, /upload/config, /upload/{id}/status	CSV validation and import status
Reports	/api/reports, /generate, /{id}/download	Protected report creation and download
Assistant	POST /api/chat	Optional language-model or keyword-based reply

D. Persistence and Data Model

The database has one ORM entity, Crime, keyed by crime_id (Fig. 2). It declares no foreign keys or relationships. An import must supply an identifier, a date, an area, a crime type and a severity. Other fields can hold a time, coordinates, a city, a case status, victim details, a weapon description, a police-station label, free text and source metadata. Indexes cover several commonly filtered fields. Even so, some application paths load all matching rows into memory before they calculate summaries or paginate.

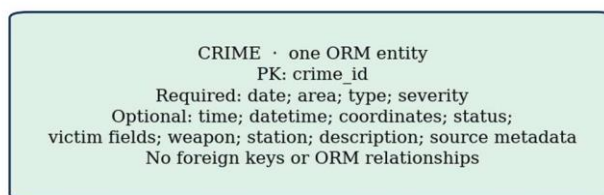


Fig. 2. The database stores incidents in one Crime table with no declared relationships to other entities.

Local development uses SQLite and the deployment configuration targets PostgreSQL [5], [6]. Both use the same SQLAlchemy model. Latitude and longitude are stored as numeric columns and their ranges are validated in Python. The project does not use PostGIS, spatial column types or spatial indexes. A typical request runs as follows (Fig. 3). The browser's shared client calls a FastAPI route, the route passes validated filters to a service, and the service queries the database through SQLAlchemy. The result goes back to the frontend as JSON. FastAPI also generates the interactive OpenAPI documentation [4].

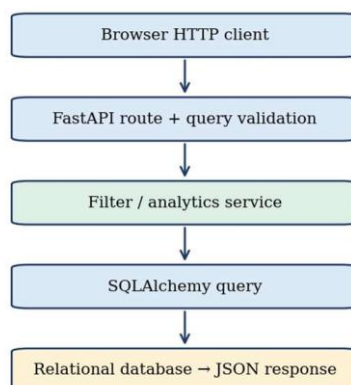


Fig. 3. Request path from a browser view through a FastAPI route and service to SQLAlchemy and the relational database.

E. Analytics and Map Layer

The analytics service groups records into crime-type and severity distributions, monthly totals, leading areas and categories, time-of-day bands, and a comparison of two adjacent 90-day periods. The map route returns rows that have

usable coordinates, up to 3,000 incidents per response. In the browser, MapLibre GL with OpenFreeMap styles draws points, clusters, severity-weighted heatmaps and count-ranked areas [8], [9]. A 3D view adds square columns scaled by record count and convex hulls (the smallest convex outline that encloses a set of points) around the displayed coordinates. These shapes help the eye but are not official boundaries, and the application calculates no formal spatial hotspot statistic.

F. Reports, Authentication and Deployment

Reports are generated from the records that match the submitted filters and download as PDF, CSV or XLSX files. Administrator credentials are configured on the server. A successful sign-in creates a signed session cookie, and protected routes also accept a bearer token. Upload and report operations require authorisation on the server; analytic read routes do not. The optional chat assistant calls an OpenAI model when one is configured and otherwise answers by keyword matching. Neither path is a validated analytical model grounded in the active records.

The Render configuration describes one web service that builds and serves the frontend, connects to managed PostgreSQL and exposes a health check [10] (Fig. 4). The checked-in public-origin values are placeholders, so we describe the deployment set-up and do not report a verified live service. On start-up the server loads its initial CSV only when its database is empty, and afterwards an authorised upload replaces the active dataset. There is no continuous feed. The browser fallback is separate: it loads a public CSV mirror and works with records held in browser memory, which says nothing about the data's provenance and does not update the server database.

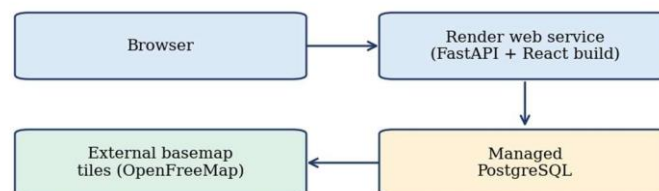


Fig. 4. Configured deployment: the browser and FastAPI service share one origin, the service uses managed PostgreSQL, and basemap tiles come from an external provider.

IV. DATA ACQUISITION AND DATASET

A. Data Sources and Scope

The bundled CSV and the local SQLite snapshot each hold 16,000 rows dated from 2022-01-01 to 2025-06-30. The file is scoped to Nagpur, although the application accepts compatible replacement data for other cities or sets of areas. The source column labels every row `demo_synthetic`, and every description marks the record as synthetic demonstration content. The repository does not explain how the records were generated, document a licence or identify a reporting source. We therefore treat the file as demonstration data and not as an official register or a complete account of incidents. Category counts drawn from it describe the example only.

B. Data Attributes

Every import must provide an identifier, date, area, crime type and severity. Optional columns carry time and a combined datetime, city, coordinates, status, victim age and gender, weapon, police station, description and source identifiers. Table IV lists the attributes the application can store. Several could be sensitive in a real dataset, and Section X-B returns to that.

TABLE IV
DATASET ATTRIBUTES

Attribute	Type	Description and role
crime_id	String, primary key	Stable row identifier; required identity
date	Date	Incident date; required time filter
time	String	Time of day; optional time band
datetime	DateTime	Combined timestamp; optional ordering and metadata
city	String	Jurisdiction label; optional filter
area	String	Administrative or local area; required grouping
crime_type	String	Incident category label; required grouping

latitude, longitude	Float	Supplied point coordinates; optional map location
severity	String	Supplied severity label; required grouping
status	String	Record or case status; optional filter
victim_age	Integer	Victim age when available; sensitive optional attribute
victim_gender	String	Victim gender when available; sensitive optional attribute
weapon_used	String	Weapon descriptor when available; sensitive optional attribute
police_station	String	Station label; optional filter or grouping
description	Text	Free-text incident description; sensitive optional detail
source	String	Source label; provenance metadata
source_record_id	String	External or source row reference; provenance metadata

C. Data Quality and Cleaning

On import the application trims headers and values, removes a UTF-8 byte-order mark and matches column names without regard to case. It treats empty-like strings as missing, parses dates, converts ages to integers and clears coordinates that fall outside the valid latitude and longitude ranges. A row is rejected when a required value is absent, when its date cannot be read, or when its identifier repeats within the uploaded file. Crime type and severity stay exactly as supplied: the application does not map them to a standard classification and does not infer severity from descriptions. In the checked-in database, victim age and victim gender are each missing from 5,054 rows (31.6%), and weapon information is missing from 14,135 rows (88.3%). All 16,000 rows have coordinate pairs within the importer's numeric limits. These figures describe the stored sample and do not explain why fields are missing. Any summary that drops missing values applies only to the rows where that field is present.

D. Processing Pipeline

At start-up the service creates its table and imports the configured initial CSV only if the selected database has no crime records. On a live deployment an administrator can submit a CSV of up to 25 MB (Fig. 5). The server validates it in a background task and, if it contains valid records, replaces the existing dataset in one transaction. Bad rows are reported. If none are valid, the existing data stay in place. The importing client polls for status. When the replacement completes, it clears its filters and invalidates its cached queries, so the available areas and the dashboard, map, trends, area profiles and any later reports use the new dataset. A one-area file naturally yields views for that area, and with a multi-area file the user can pick an area through the shared filters or Area Explorer. If the API is unavailable, the browser fallback can replace its in-memory rows instead. That change is local to the browser and is not a durable server import.

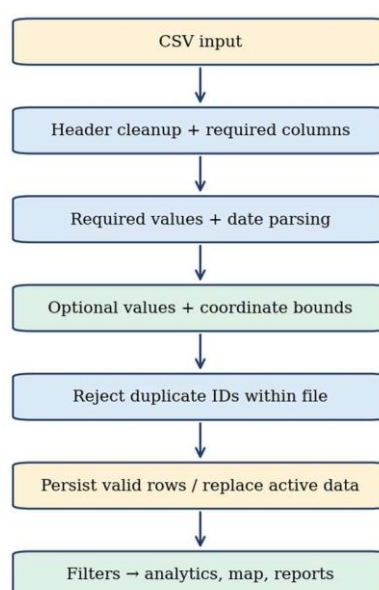


Fig. 5. CSV upload: rows are checked and normalised, then valid records replace the active dataset and feed the application's views.

E. Geographic Coordinate Handling

The importer keeps a coordinate only if latitude lies between -90 and 90 and longitude between -180 and 180 . Missing or malformed values become null. The map omits rows without a valid coordinate pair and reports how many it excluded; no rows in the bundled snapshot are omitted. Passing these range checks shows only that a point can be plotted. It does not show that the point is accurate, that it protects anyone's privacy, or that it lies inside the correct administrative area.

F. Crime and Severity Categories

Crime type and severity are labels supplied with each row. The sample contains 16 crime-type labels and four severity labels: Low, Medium, High and Critical. The backend does not infer severity. Some summaries group the exact labels High and Critical, so a replacement file that spells or capitalises them differently would change those counts.

V. DATA PROCESSING AND ANALYTICS

A. Filtering and Aggregation

Users can filter by inclusive start and end dates, crime type, area, severity and time of day. Several selections within one category are combined, and selections in different categories narrow the result together. Dates must use the ISO format YYYY-MM-DD, and an invalid date returns HTTP 422. The time-of-day band is derived from each row's stored time. In the interface, edits to the filters stay a draft until the user applies them.

After filtering, the service loads the matching rows and calculates the summaries in Python. The dashboard reports filtered and total counts, High and Critical counts, the most common area and crime type, and distributions by type, area, severity, month and time of day. The trends page adds monthly totals and compares the latest 90 days in the data with the 90 days before them. Every figure is a record count, not a population-adjusted rate.

B. Area Ranking and Area Profiles

For the current filters the hotspot endpoint sorts areas by record count and returns up to 15. It reports each area's share of the filtered records and assigns a relative label by comparing the area's count with the busiest area's count. The density_per_sq_km value is always left null. When coordinates are available, the area marker sits at their arithmetic mean. The label and the marker make a result easier to inspect, but neither is a risk score or a spatialintensity estimate.

An area profile shows that area's counts by category, severity and time, its monthly trend, and how its record count compares with the leading areas. The application has no area polygons and no population denominator. Suppose one area has 2,000 matching records and another has 500. Without knowing how many people live in each, or how large each is, we cannot say the first is more dangerous, and the application does not suggest it.

C. Map Processing and Visualisation

The analytical map uses the coordinates supplied in the records. MapLibre draws incident points, clusters and a heat layer weighted by the stored severity, while the area ranking is based on counts. The 3D display adds square columns scaled by record count and convex hulls around the displayed points. Neither shape defines an official area boundary, and none of the layers computes a statistically validated hotspot. Each filter request retrieves a maximum of 3,000 geolocated incidents through the API, without administrative boundary data. The map relies on OpenFreeMap tiles and styles, along with the user's browser connectivity [9]. As no live feed is integrated, the map displays the latest successfully updated dataset from either the server or browser fallback (Fig. 6).

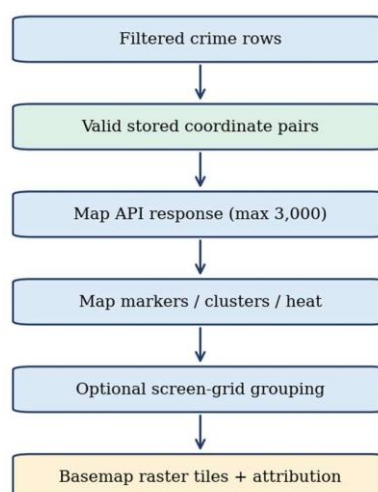


Fig. 6. Filtered records with valid coordinates are capped by the API, then shown over an external basemap as points, clusters, heat and count-ranked 2D/3D layers; derived hulls are not jurisdiction boundaries.

D. Analytics Workflow and Interpretation

One filter selection is used for the records list, the dashboard summaries, the trends, the area ranking, the map and the report request (Fig. 7). A user can therefore move from a broad view to a specific set of records without rebuilding the query on every page. As an illustration, an analyst might set a date range and two areas, apply the filters, compare the 90-day trend, switch to the map to see where the matching incidents fall, and then export the same selection as a PDF. The displays support inspection. They do not establish cause, and they do not say where resources should be sent. Table V follows the data through each feature, and Fig. 8 shows the whole flow.

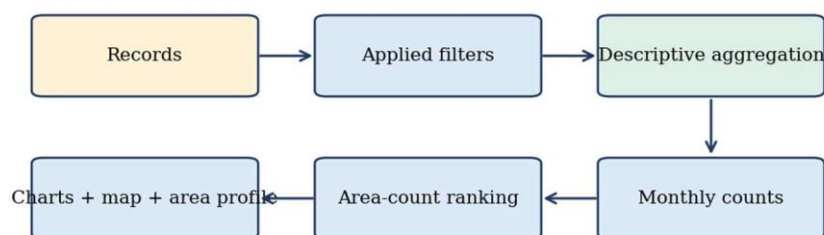


Fig. 7. Analysis path from input records to descriptive summaries and map or chart views for human review.

TABLE V
 SYSTEM FEATURES AND DATA FLOW

Feature	Input	Processing	Output
Dashboard	Filtered crime rows	Counts and distributions	KPI cards, insights, charts, record list
Trends	Filtered rows with dates	Monthly counts and adjacent 90-day comparison	Trend charts and comparison strip
Area profile	Selected area and filters	Area-specific counts and coordinate mean	Profile, charts, comparison, point map
Area ranking	Filtered rows	Sort areas by raw count; relative thresholds	Ranked list and marker map
Incident map	Filtered coordinate pairs	Range validation and 3,000-point cap; severity-weighted heatmap, clusters, derived hulls and 3D count columns	MapLibre 2D/3D layers; no official boundary or statistical hotspot
CSV upload	Administrator CSV	Validate, normalise, reject rows, replace active data	Status and refreshed views
Reports	Filtered rows and selected sections	Server-side PDF, CSV and XLSX export	Preview, history, downloadable file
Administrator access	Server credentials or bearer token	Constant-time credential and signature comparison	Signed session cookie and protected actions
Landing globe	Country boundary file and fixed illustrative region anchors	MapLibre globe projection, graticule, decorative routes and markers	Interactive visual context; not incident data
Browser fallback	Configured CSV mirror when the API probe fails	In-browser descriptive calculations over loaded records	Fallback client state; not live or server-synchronised
Conversational assistant	User message; optional configured model	OpenAI completion or keyword fallback	Help response; not validated or datagrounded analysis

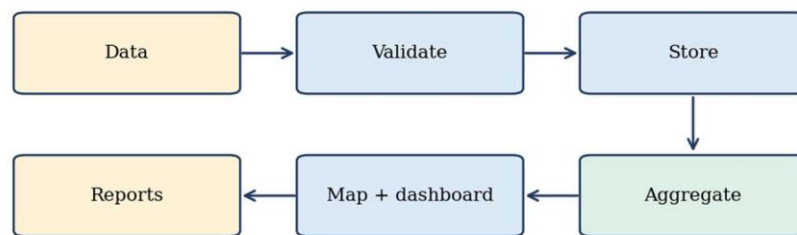


Fig. 8. End-to-end flow from CSV import or preloaded records through validation, storage, descriptive views and reporting.

VI. SYSTEM IMPLEMENTATION

A. Frontend Implementation

The frontend has a public landing page and pages for the dashboard, crime map, hotspots, trends, area explorer, upload, reports, settings and help, plus a not-found page. The landing page's MapLibre globe uses country geometry derived from Natural Earth [11], a graticule, and decorative routes and markers; none of them marks a crime event. Shared filter state and a common request layer keep the analysis pages aligned. Reusable components show charts, tables, loading states, errors and empty results. The workspace also has a contextual chat assistant. The source includes responsive layouts and accessibility features, but we did not run a formal accessibility audit.

B. REST API and Database

FastAPI exposes read routes for health, filter options, dashboard data, crime records, analytics, map incidents and chat. Separate routes manage administrator sessions, uploads and reports. At start-up SQLAlchemy creates the table and the service seeds an empty database from the configured CSV. When the API is unavailable, selected read views can use a TypeScript implementation over the configured CSV mirror. This is a browser-side fallback, not another database.

C. Data Upload

The upload wizard accepts CSV files up to 25 MB. It checks the file in the browser, previews a sample of rows and checks for the required columns before submission; the server then repeats the validation and processes the import in a background task. The page polls the upload status instead of waiting for the whole import inside one request. A successful live import replaces the previous active dataset, after which the importing client refreshes its shared views. The upload lock and the status record exist only in the running process, so this is not a durable job queue. Uploading while the browser fallback is active changes only its in-memory dataset.

D. Report Generation

A report request carries a title, the selected sections, a format and the filters. The server can produce PDF, CSV or XLSX files. PDF record lists stop at 300 matching rows, whereas CSV exports and XLSX files that include the Records section contain the complete filtered set. The report page also offers a preview and a history, but the metadata is held in memory and not in the database, so a process restart can remove the history and the download lookup even when a generated file remains on disk.

E. Authentication and Protected Operations

The server provisions administrator accounts through environment variables, and users cannot register publicly. After a successful login the browser receives a signed HttpOnly, SameSite=Strict cookie that lasts 30 days and is marked Secure in production (Fig. 9). Protected routes accept that cookie or the configured bearer token. Read access to analytics and map data does not require login. The application has no login rate limiter, and we did not commission an independent penetration test.

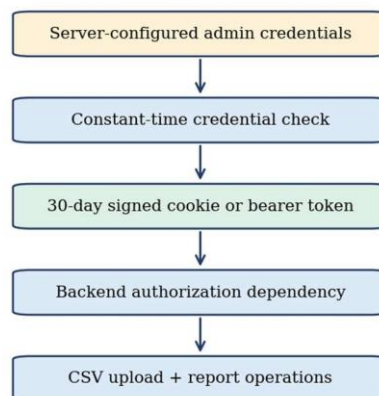


Fig. 9. Administrator credentials establish a signed browser session or bearer access before protected upload and report actions are allowed.

VII. VISUAL ANALYTICS AND USER INTERFACE

A. Dashboard and Charts

The dashboard combines KPI cards, monthly activity, short descriptive insights, category distributions and a filtered incident table, with Recharts drawing the charts (Fig. 10). Elsewhere in the workspace MapLibre supports the 2D and 3D map views. Dashboard summaries describe the selected records and are not estimates of population incidence. The contextual assistant answers through an optional external model or a keyword fallback, and it is not a source of calculated metrics.

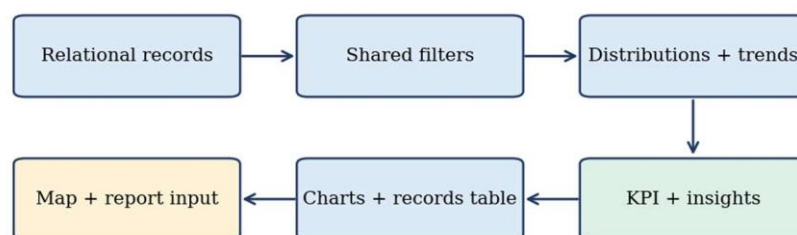


Fig. 10. Filtered rows are summarised into KPI and distribution data before the dashboard renders its charts and record table.

B. Filters and User Interaction

The shared filter panel offers date bounds and selections for crime type, area, severity and time of day. Changes stay in a draft until applied, and the interface shows the active filters so users can see the scope of a view. The same selections carry through the dashboard, map, trends, area profile, area ranking and report request. After an upload completes, the importing client clears the old filters and reloads its data, and an area can then be chosen from the refreshed options.

C. Spatial Exploration

On the map users can switch between heat, cluster, incident and count-ranked area layers, between 2D and 3D, and between basemap styles. In 3D, square columns encode counts and convex hulls show the extent of the displayed points; neither outlines a jurisdiction. The area explorer combines a selected area's summary and charts with a marker at the mean of its record coordinates. Locations come from the dataset, and the display changes when the active dataset changes, not as incidents stream in.

D. Responsive and Accessible Design

The source includes desktop and mobile navigation, responsive layouts, a skip link, labelled inputs, keyboardoperable upload controls, status announcements and reduced-motion styling. These are useful details, but they do not show conformance with an accessibility standard, and we ran no assistive-technology or cross-device audit.

E. Reporting Workflow

An administrator can choose a report format and sections, generate the output, review the history and download the file. A report is a snapshot of the rows and filters in force when generation ran, and it does not change when the dataset is later replaced. The chat assistant is a separate help feature and does not calculate report contents.

VIII. EXPERIMENTAL EVALUATION

A. Evaluation Protocol

For this revision we ran three checks: a read-only profile of the bundled CSV and SQLite database, the seven-test backend suite, and a production build of the frontend. We also inspected the map and globe implementation. These cover a small set of local paths. We did not evaluate a production service, a browser matrix, query performance, accessibility, usability or security, and we ran no user study. The system contains no predictive model, so there is no predictive accuracy to evaluate.

B. Results

All seven backend tests passed. The Vite production build also completed, although it reported minified chunks above 500 kB. The read-only check found 16,000 records, 16 crime-type labels, 18 areas and dates from 2022-01-01 through 2025-06-30, and the map route caps each response at 3,000 incidents. The results confirm selected code paths and properties of the bundled data. They do not measure latency, scalability, uptime, map usability, analytical validity, representativeness or crime-prevention outcomes.

C. Evidence Summary

Table VI lists each check, what it covered and what it leaves open.

TABLE VI
EVALUATION AND TESTING SUMMARY

Check	Evidence and outcome	Not established
Bundled dataset profile	Repository CSV and local SQLite snapshot: 16,000 rows; 16 crime types; 18 area labels; 2022-01-01 to 2025-06-30	Data provenance; representativeness
Check	Evidence and outcome	Not established
Backend unit tests	backend/tests/test_admin_auth.py, pytest -q: 7 passed	Penetration resistance; complete API behaviour
Frontend production build	Vite production build succeeded; warning for minified chunks above 500 kB	Browser or device usability; runtime performance
Map and globe data	Implementation inspection: MapLibre layers; landing globe uses illustrative anchors	Geospatial accuracy; live-feed behaviour

D. Future Evaluation

A fuller evaluation should begin by documenting where the data came from and what use of the system is approved. It could then measure query performance at stated dataset sizes, test the map in supported browsers and with assistive technologies, and check whether uploads and reports behave as intended across restarts. A user study should involve appropriately authorised analysts. Before adding a formal hotspot method or a forecast, the project would need to define its spatial unit and denominator, test it on held-out data, report uncertainty and compare it with a stated baseline.

IX. RESULTS AND DISCUSSION

A. Dataset Observations

The bundled records contain 16,000 rows, 16 crime-type labels, 18 area labels and four severity labels over the period 2022-01-01 to 2025-06-30. Theft is the largest category, with 3,390 rows (21.2%). By severity, Medium accounts for 6,294 rows (39.3%), Low for 5,247 (32.8%), High for 3,451 (21.6%) and Critical for 1,008 (6.3%). High and Critical together make up 4,459 rows (27.9%). All percentages use the 16,000 rows as the denominator and describe only this synthetic sample. Every row has a coordinate pair that passes the importer's range checks, which means the rows can be plotted, not that anyone has verified their locations. Victim age and gender are missing from 31.6% of rows and weapon information from 88.3%. We do not know why, so any conclusion that relies on those fields would need further checks.

B. Implementation Observations

The main thing the implementation achieves is that one active dataset and one filter state support a set of connected views. A valid upload replaces the active records, and the importing client reloads the filter options and summaries. An area-only dataset appears throughout as that area, while a multi-area file can be narrowed with the shared filters. The browser fallback keeps the interface usable when the API is down, but it is separate from the server database and does not provide live updates. Passing tests and a successful build show that selected paths run. They are not evidence of operational reliability.

An area's count is a raw total. It may reflect population, area size, access to reporting or the way data were collected as well as the number of events, and the relative labels neither adjust for those factors nor quantify uncertainty. The 3D

columns and convex hulls are display devices. These views support descriptive exploration; they do not rate how dangerous a place or a person is.

C. Practical Usefulness and Weaknesses

For an authorised user, the shared filter-to-map, chart and report workflow avoids repeating the same manual summaries in separate tools. Users can see which records are included, inspect results by area and export the selected view. Server-side validation also stops a file with no valid rows from erasing the active dataset. These are workflow benefits, not evidence of better policing outcomes.

The weaknesses are just as concrete. The example data's source and reuse terms are undocumented, and a real upload could contain exact locations or sensitive descriptions. Upload status and report history live in process memory. Some queries load all matching rows into the application, and the map and PDF exports have explicit caps. These constraints should be dealt with before the application is used on larger or operational datasets.

X. SECURITY, PRIVACY, AND ETHICAL CONSIDERATIONS

A. Access Control and Deployment Secrets

Administrator credentials are provisioned on the server, and a successful sign-in creates a signed cookie that is HttpOnly and SameSite=Strict, and Secure in production. The administrator password and bearer token must stay in server environment variables and out of frontend build variables. In production a token is required, and upload and report routes check for a cookie or bearer credential. There is no public registration.

The current API does not require login to read records, analytics or map responses. Depending on the dataset, those responses can include coordinates and incident descriptions. If the optional OpenAI-backed assistant is enabled, user messages are sent to the configured provider, so a deployment should review that data flow and the provider's terms. This paper is not a penetration test. The points above are access and privacy matters to review, not confirmed vulnerabilities.

B. Privacy and Responsible Visualisation

The record model can hold victim age, gender, weapon details, police-station names, descriptions and point coordinates. In real records, a precise location combined with a rare event or a narrative could identify a person. CrimeVista does not currently generalise coordinates, redact fields row by row, apply privacy thresholds or define data retention. The landing globe is separate and uses generalised anchors, not incident locations, but that separation does not remove the risks of the analytical map. Any real deployment needs a review of its own dataset before opening access or allowing broad exports.

A count-ranked area must not be presented as the probability that a person or place will experience crime. It is the number of records that match the chosen filters. Clear labelling matters because a polished map can make a descriptive result look more certain than it is.

C. External Map and Operational Dependencies

The analytical map requests styles and tiles from OpenFreeMap [9], and the landing globe loads country boundaries from the world-atlas package [11]. Both rely on external services or files and on their terms of use. The globe's outlines and decorative routes are not incident layers. Database credentials and administrator secrets remain on the server and are not exposed to the browser.

XI. FUTURE WORK

Data governance should come first, ahead of any more complex model. A data dictionary should identify the source and licence, explain how the file is updated, define identifiers and classification labels, and state privacy requirements for each field. A pre-import quality report could show missing values and schema changes before an administrator replaces the active dataset. The example's synthetic status and generation process should be documented before anyone presents its counts as representative or official.

Any spatial analysis would need trusted area boundaries, a clearly defined unit, suitable denominator data and a method matched to a specific question. Kernel density or a spatial scan statistic, for example, would need sensitivity checks for bandwidth, time window and boundary choice. Such analyses should stay separate from visual clustering and marker grouping, and rates should be reported with their denominators and uncertainty.

On the operational side, upload jobs and report metadata could move to durable storage, and database-side aggregation and pagination could cut the in-memory work. Performance should be measured against documented dataset sizes. Security work should decide which data are public, minimise or redact sensitive fields, set retention and audit rules, and add abuse controls to administrator sign-in. Usability and accessibility should be tested with intended users and assistive technologies.

Real-time ingestion, anomaly detection, forecasting and resource optimisation remain research directions and are not current features. A live feed would need an identified source, a lawful basis, a documented update schedule and monitoring for failures. Any model would need a clear target, temporal or geographic validation that avoids leakage, a baseline comparison, fairness and privacy review, and human oversight before operational use.

XII. CONCLUSION

CrimeVista is a web application for exploring and reporting on loaded crime-record datasets. A successful CSV import replaces the active records, and the connected dashboard, map, trends, hotspots, area explorer and later reports all use that dataset. Shared filters let users focus on one area when a file contains several. The bundled Nagpur example holds 16,000 rows of synthetic demonstration data. Seven backend tests passed and the frontend built successfully, but neither result shows that the data represent actual crime, that count-ranked areas measure risk, or that the system improves operational outcomes. The next steps are to document data lineage and licensing, govern sensitive fields and assistant use, add validated spatial methods where they are needed, and test durable operation and usability. Until then, CrimeVista is best used as an exploratory reporting interface, and its outputs should be read with the limits of the data in mind.

REFERENCES

- [1] N. Shah, N. Bhagat, and M. Shah, "Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention," *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, article 9, 2021. DOI: 10.1186/s42492-021-00075-z.
- [2] F. Dakalbab, M. Abu Talib, O. Abu Waraga, A. Bou Nassif, S. Abbas, and Q. Nasir, "Artificial intelligence & crime prediction: A systematic literature review," *Social Sciences & Humanities Open*, vol. 6, no. 1, 100342, 2022. DOI: 10.1016/j.ssaho.2022.100342.
- [3] S. Vijayalakshmi, R. K. Dhanalakshmi, and P. Suresh, "Crime rate prediction using ensemble machine learning algorithms," *IEEE Access*, vol. 11, pp. 24512–24525, 2023. DOI: 10.1109/ACCESS.2023.3258123.
- [4] C. Zhang, L. Liu, and M. Wang, "Graph neural networks for spatial-temporal crime prediction in smart cities," *Information Sciences*, vol. 625, pp. 112–128, 2023. DOI: 10.1016/j.ins.2023.01.045.
- [5] H. K. Sharma and R. M. Tailor, "GIS-based crime hotspot mapping and its correlation with land use: A case study on assessment of CPTED principle," *International Social Science Journal*, vol. 74, no. 253, pp. 775–786, 2024. DOI: 10.1111/issj.12478.
- [6] M. Rodriguez and S. L. Jenkins, "Predictive policing and crime forecasting: A comparative study of time series and machine learning models," *International Journal of Law, Crime and Justice*, vol. 76, article 100652, 2024. DOI: 10.1016/j.ijlcj.2024.100652.
- [7] S. Ibrahim, P. Jain, and M. Bhardwaj, "Predictive Crime Hotspot Detection: A Spatial Analysis Approach," *Lecture Notes in Networks and Systems*, vol. 1409, Springer, 2024.
- [8] M. Vance, H. Rostova, and K. Sato, "Big data analytics and spatial data mining for urban crime pattern identification," *Journal of Big Data*, vol. 11, no. 1, article 45, 2024. DOI: 10.1186/s40537-024-00892-x.
- [9] D. K. Miller, E. Gomez, and P. Patel, "Explainable AI (XAI) in crime prediction: Enhancing transparency in machine learning decision-making," *Expert Systems with Applications*, vol. 240, article 122543, 2025. DOI: 10.1016/j.eswa.2024.122543.
- [10] Riyazuddin, S. Z. Rahman, and J. Sadiq, "Spatiotemporal ('Crime prediction') and forecasting using machine learning," *SN Computer Science*, vol. 7, no. 2, article 213, 2025. DOI: 10.1007/s42979-025-03512-y.
- [11] Albers Zumel, M. Tizzoni, and G. M. Campedelli, "Deep Learning for Crime Forecasting: The Role of Mobility at Fine-grained Spatiotemporal Scales," *Journal of Quantitative Criminology*, 2025. DOI: 10.1007/s10940-025-09629-3.
- [12] 10.1007/s10940-025-09629-3.