HouseLens: An Interactive Web Platform for Exploring London Housing Market Dynamics

Group： PLZ push me

Group Members: Xintong Shao (25124122): Create database; User interface and interaction development; Data visualization; Website deployment; Documentation Writing.

Tianrui Min(25047434): Data collection and processing; Backend API implement; Documentation Writing.

Yewei Bian(25206465): UI design; User interface and interaction development; Documentation Writing.

Git Hub Repository: Tongtong828/HouseLens

Website link: https://10.129.111.8

Abstract

The website named HouseLens which is developed for achieve an interactive online platform for exploring housing prices in London, studying the changes of housing prices in terms of space and time. With the increasing prominence of issues such as housing affordability, regional development inequality, and uneven access to housing opportunities, this project aims to explore the price differences among different boroughs in London and the evolution of these differences over the past five years.

The data used in the project comes from the residential transaction records of London from 2021 to 2025, as compiled by the UK Land Registry. Through interactive maps, regional price overviews, and annual trend analyses, the platform presents the spatial distribution, ranges of prices, and change trajectories of housing prices, and allows users to further view individual transaction information.
This project focus on research that how did the housing prices in different boroughs of London evolve in space and time between 2021 and 2025.

Houselens shows the market trends of housing prices in London by converting large-scale and complex housing transaction data into easily understandable visual content on the website, providing a clear and user-friendly exploration tool for the housing market in London for consumers, relevant government officials, and citizens.

Background

The continuous rise and drastic fluctuations in London's housing prices over the past few years have become one of the most concerning topics for the citizens. The current situation of the London housing market, which shows a decline in affordability, increased inequality, and development disparities between regions, has put pressure on residents and policymakers (GLA, 2023; Wilson and Barton, 2022). Although the British government has published a large amount of public data on the GOV.UK website, such as the Price Paid Dataset that records all residential transactions in England and Wales, the data is voluminous, structurally scattered, and not easily readable, making it difficult for ordinary citizens to extract meaningful information from it.

The development of the website HouseLens was based on the above realistic problems. The main reason is that the data on London housing is various, but most of the dataset have the excessive amount of data and poor readability. Most platforms only provide static charts or statistical results, lacking interactive methods that allow users to explore and compare on their own. For London residents, those considering buying a house, or policymakers of the housing, there is still a lack of a tool that can intuitively understand the price changes in each borough, community differences, and overall housing trends.

Therefore, this project aims to convert the government's released property transaction data into an easily understandable and interactive map-based visualization platform for users. By integrating geographical location, borough prices, and five-years average price trend analysis, HouseLens enables users to explore the London housing market from both spatial and temporal perspectives. Users can view the price differences in each Borough, track the five-years trend, and dig into the details of each housing transaction information. Therefore, the data transparency has been enhanced and allow users to make more rational decisions and conducting research activities.

Objectives

1. **Processing Data**

   Merge the obtained data sets, clean and standardize them, including correcting inconsistent fields, removing invalid records, mapping out the missing latitude and longitude from the right side, and reorganizing the data into a format suitable for database import.

2. **Creating Database**

   Create a database through use MySQL, including a table named transaction for storing transaction information. This table will be used to store the detailed information of each housing transaction in each borough of London from 2021 to 2025.

3. **Designing Use Interface**

   Design the wireframe diagram and implement a user-friendly interface that enables users to directly explore the changes in house prices in London, including clear map navigation, filtering controls, interactive panels, and enhanced readability.

4. **Data Visualization**

   Based on the use of Google Maps, interactive visualizations have been added, including heat maps and five years' housing price trend charts, to effectively showcase the differences in housing prices among various borough and the changes in the housing market over the years.

5. **Developing Backend API**

   Develop a backend by using Node.js that exposes structured data endpoints for use by the frontend. The API returns processed data including borough-level statistics, five-year price trends and recent transaction details. This provides a reliable communication layer linking the client interface with the database.

6. **System Integration**

   Integrate database, backend, frontend, and server into a fully functioning website. This includes configuring Nginx for assigned hosting, PM2 for backend process management, and ensuring the website can be using when the local server close.

Methodology

System Architecture

The website HouseLens's architecture as Figure 1. It include a front-end presentation layer, a back-end logic layer, and a database data layer. The architecture comprises a frontend built using HTML, CSS, and JavaScript, the backend based on Node.js and a MySQL database for data storage and retrieval.

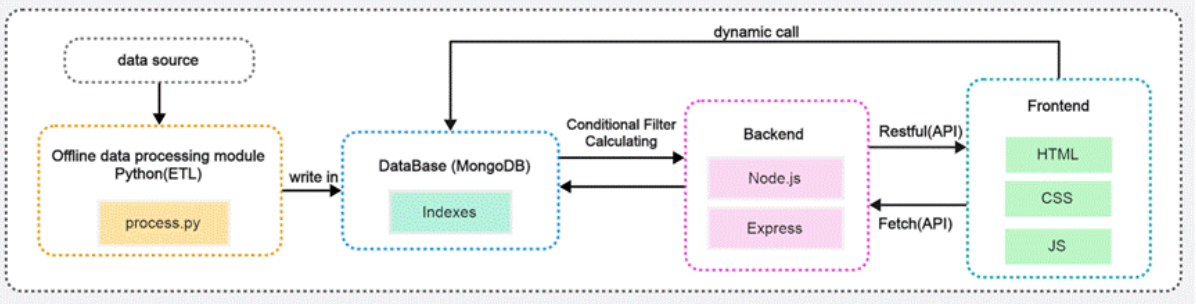

Figure 1 System architecture

The front-end layer focuses on data visualisation and user interaction. It communicates with the back-end via RESTful APIs and dynamically updates visualisation components, including heatmaps and five-year housing price line charts. The backend layer is implemented using Node.js and Express.js frameworks, handles API routing, data aggregation, and query execution. The database layer stores London's each borough housing transaction detail information.

Frontend Development and Interaction Design

The frontend integrates multiple technologies. The Bootstrap for front framework, Google Maps API provide the basic map,use the JavaScript for interaction and load london_boroughs.json which is a local GeoJSON file that defines polygon geometries for display the boundaries of each borough of London as Figure 2 shows, and Chart.js for exist trend line chart.

```
//The boundaries established in accordance with the official administrative divisions of London
map.data.loadGeoJson("./data/london_boroughs.json", null, features => {
  const boroughNames = [];
  features.forEach(f => {
    const name = f.getProperty("NAME") || f.getProperty("name");
    if (name) boroughNames.push(name);
  });
```

Figure 2 Load Borough boundaries by loading GeoJson

Data Visualisation

The frontend visualisation logic connects the map interface, statistical summaries and trend charts through asynchronous data updates, enabling users to interactively explore housing affordability.

The heatmap is the main visualisation layer. It shows the average housing prices in London boroughs by using color coded polygonal areas. This makes it easy to see where housing is affordable or expensive. Housing prices range from low to high, corresponding to colors from yellow to red on the heatmap. The heatmap allows users to see all London's housing price directly. The heatmap dynamically adjusts area fill colors by reading the latest average price data from the backend in real-time through the interface.

In order to visualise long-term housing price trends, a dynamic trend chart was implemented by using *Chart.js*. According to Figure 3, the chart is first initialised with an empty dataset and configured with smooth curves, filled areas, and currency-formatted y-axis labels. When a borough is selected, the system fetches yearly average prices from the backend and updates the chart by update the received years and values into the dataset before display as Figure 4.

```
//Initialise trend chart
trendChart = new Chart(ctx, {
  type: "line",
  data: {
    labels: [],
    datasets: [{
      label: "Average Price (£)",
      data: [],
      borderColor: "#007bff",
      backgroundColor: "rgba(0,123,255,0.2)",
      tension: 0.3,
      fill: true,
      pointRadius: 4
    }]
  },
  options: {
    responsive: true,
    plugins: { legend: { display: false } },
    scales: {
      y: { ticks: { callback: v => "£" + Number(v).toLocaleString() } }
    }
  }
});
```

Figure 3 Initialise trend chart

```
//Update Trend Chart (with data)
Qodo Gen: Test this function
function updateTrendChart(data) {
  if (!trendChart) return;
  const years = data.map(item => item.year ?? item.Year);
  const prices = data.map(item => Number(item.avg_price ?? item.average_price ?? item.price ?? 0));
  trendChart.data.labels = years;
  trendChart.data.datasets[0].data = prices;
  trendChart.update();
}
```

Figure 4 Update trend chart

Backend development

The backend of the website was implemented in server.js, which uses Node.js and Express. The db.js file connects to the database through environment variables configured in the .env file. As Figure 5 shows that through modify .env, each team member holds credentials safely connect to either local or remote databases in raspberry pi securely.

```
.env
1  DB_HOST=
2  DB_USER=
3  DB_PASS=
4  DB_NAME=houselens
```

Figure 5 .env configuration file

According to Table 1, three API has been created that use to get data from database.

API	Description
/api/borough-prices	Returns the average property price of each borough, which can be used for coloring the map.
/api/transactions/:borough	Returns the specific transaction records for the selected borough that used to display the transaction points in red on the map.
/api/borough-trend/:borough	Returns the yearly average house price data for use in the trend chart.

Table 1 API description

Website deployment

Nginx is configured as a lightweight static web server to host the frontend interface (Nginx, 2024). It serves the production-ready HTML, CSS, and JavaScript files from the */var/www/html/* directory, which allowing deploy HouseLens's frontend to the Raspberry Pi's allocated pi host. Its low resource consumption and strong stability make Nginx a suitable choice for deploy the website in raspberry pi. PM2 is used to run and manage the backend Node.js service (NPM, 2024). After set environment variables and put the frontend into nginx, using PM2 launches the Express server, ensures it remains active after SSH disconnection when putty closed. This guarantees constant availability of the backend API.

By combining Nginx and PM2, the HouseLens achieves a reliable deployment on host 10.129.111.8. This configuration enables the entire system to operate autonomously without supervision. It ensures precise communication between the frontend interface and the back-end API.

Implementation and Results

The implementation process and final outcomes of this project will be presented in two parts. Firstly, the design phase will be outlined, including website structure design, wireframes, and data collection and processing. Subsequently, specific implementation details of the development process will be demonstrated through project screenshots.

Project design

1. **Website structure design**

   The website structure diagram shown in Figure 5 clearly illustrates the relationships among the system components, helping users understand the overall system and its modules. The architecture of this website is designed to enable smooth interaction between modules. Index page serves as the entry point of the website's front end, guiding users to the main interface. Data Visualization module integrates the core functions of the site: displaying heat maps, comparing detailed property information, and visualizing historical price trends. Data Information module acts as the core of the website's back end. When users interact with the webpage, data is retrieved and transmitted to the front end through this module. About and

Contact sections provide information related to the website's development process. This layered structure maintains clarity while offering scalability for future feature expansion.

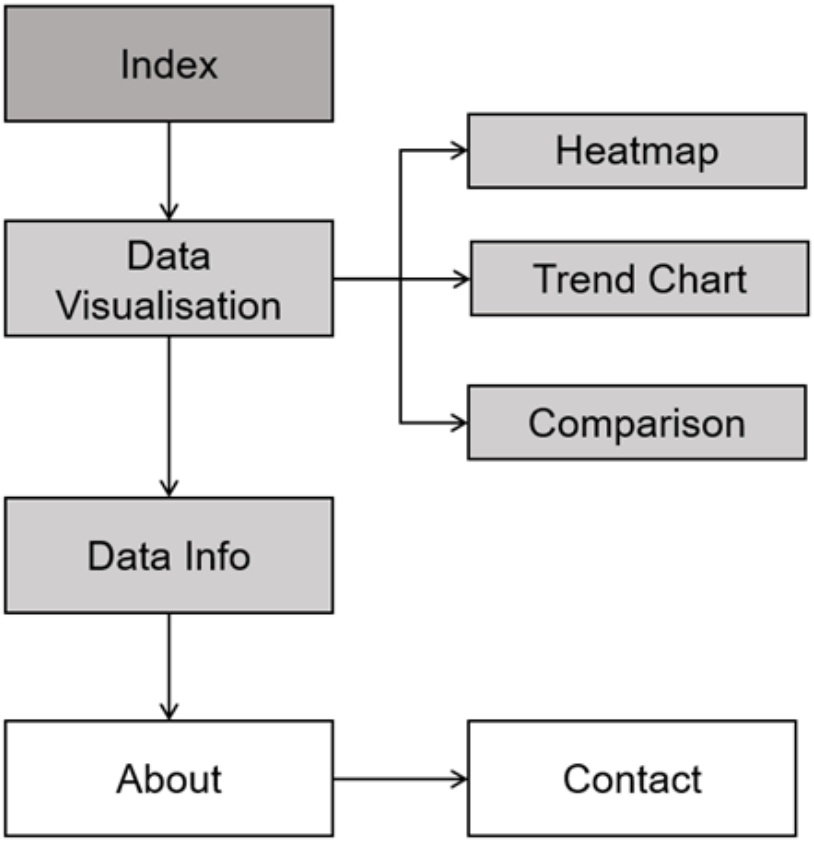

Figure 5: Website structure

2. **Wireframes**

Wireframes are utilised to illustrate the structural framework and functional layout of website pages, constituting an essential preparatory element for data visualisation. At this stage, the content information contained within each page and the precise interactive positioning of functionalities must be clearly defined. This provides developers and designers with visual reference during the coding process.

As the Figure 6, the Index Page will feature a top Navigation Bar for global site navigation. The central main area will comprise an information panel displaying the site title 'Exploring London's Housing Affordability', a brief explanatory text, and an interactive button labelled "Start". Clicking 'Start' will navigate users to the site's main interface.

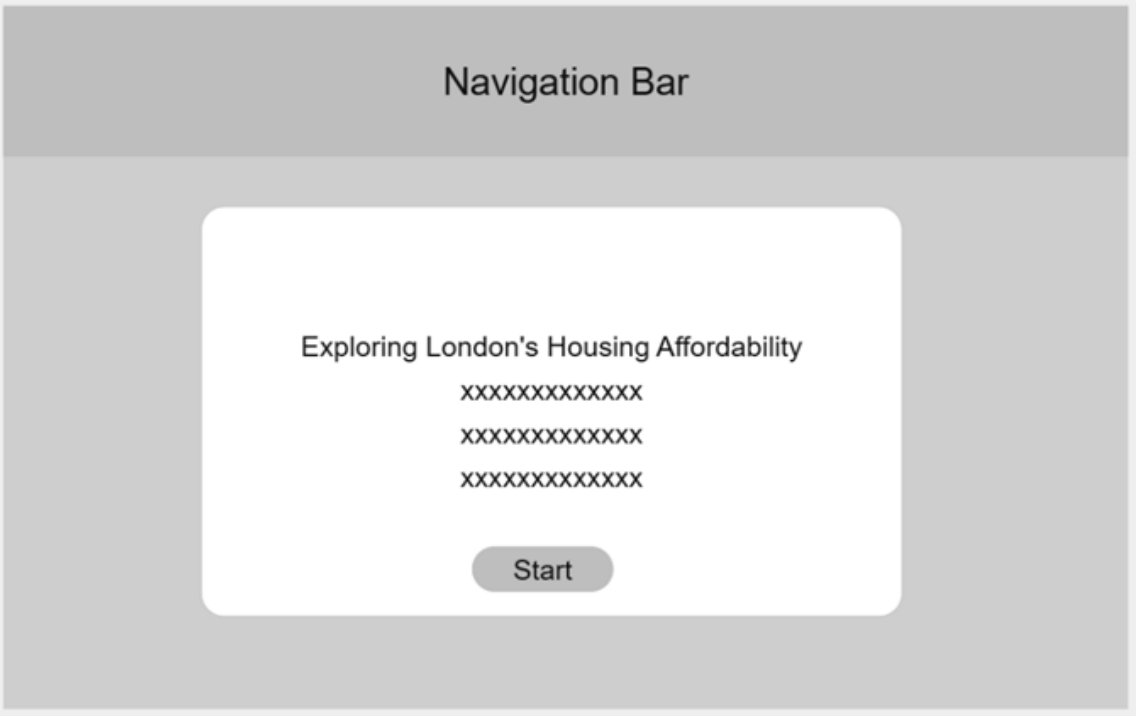

Figure 6: Index page wireframe

The website's main page features a map as its body, displaying a heatmap of housing prices across London in the central area as Figure 7 shows. This clearly illustrating the geographical distribution of London's housing market. Interactive buttons are positioned on the right-hand side. The “Full” button enables full-screen mode. The “See” button utilises Google Maps' Street View functionality, allowing users to remotely explore properties. “+ / - ” denotes the zoom function. The bottom-left corner displays information such as the average property price legend. Additionally, the main page retains the navigation bar for convenient access back to the homepage or to switch between modules.

Figure 7: Heatmap page

According to Figure 8, the navigation bar at the top of the trend analysis page now incorporates filter functionality, comprising a borough selector and a price selector. A “Reset” button has been added to the right of the navigation bar, clearing all current user interactions and returning the interface to its initial state. The left half of the page deploys a line chart

pop-up, displaying property prices at different time points to illustrate price trend changes. The right-hand section displays the currently selected area on a heatmap for spatial comparative analysis.

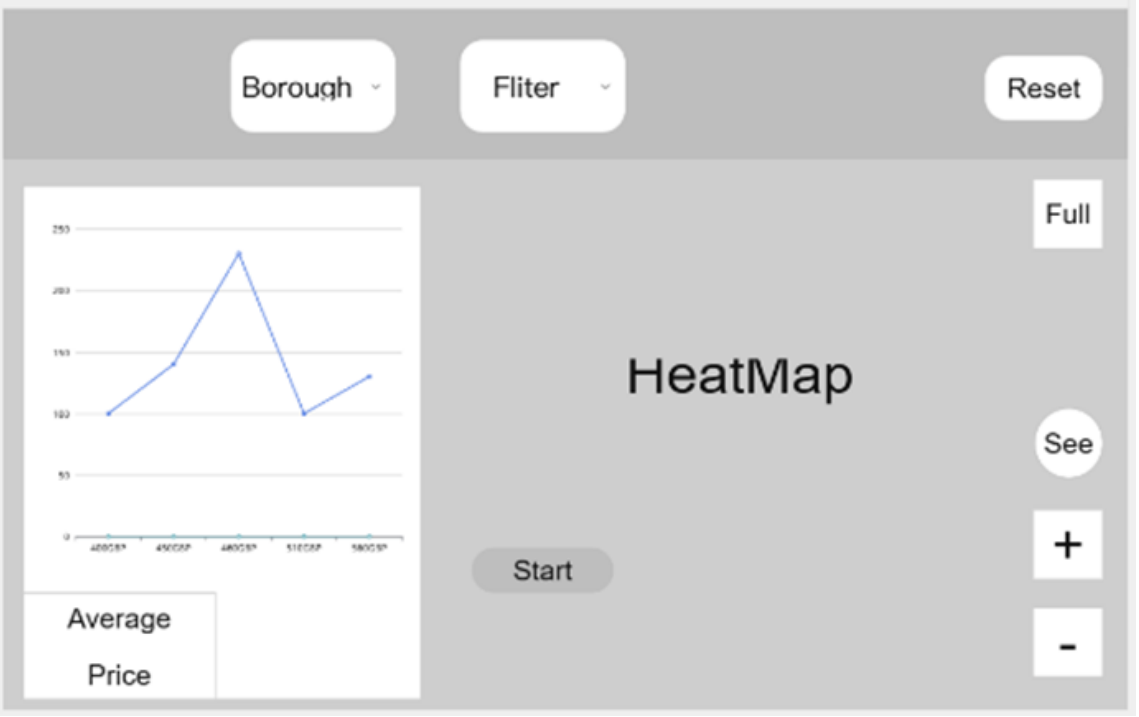

Figure 8: Trend chart page

3. **Database Construction**

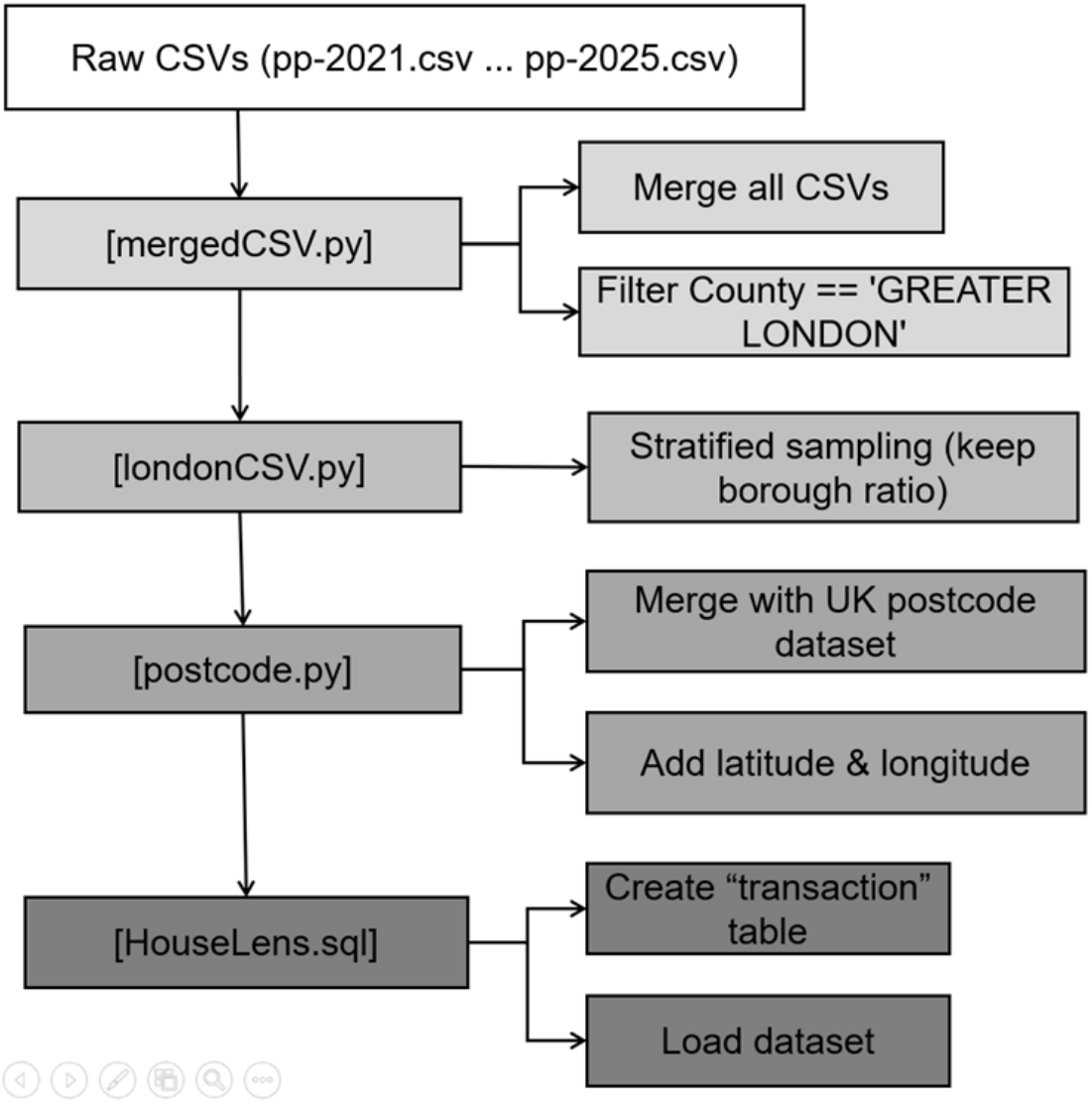

Figure 8: Build database flowchart

The process of constructing the database involves data collection, data processing and data extraction.

Step one: Obtain property sales data from the past five years via the official Price Paid Data (PPD) database of the UK Government's Land Registry, which is reliable and csv file is easy to be coded in computer editors.

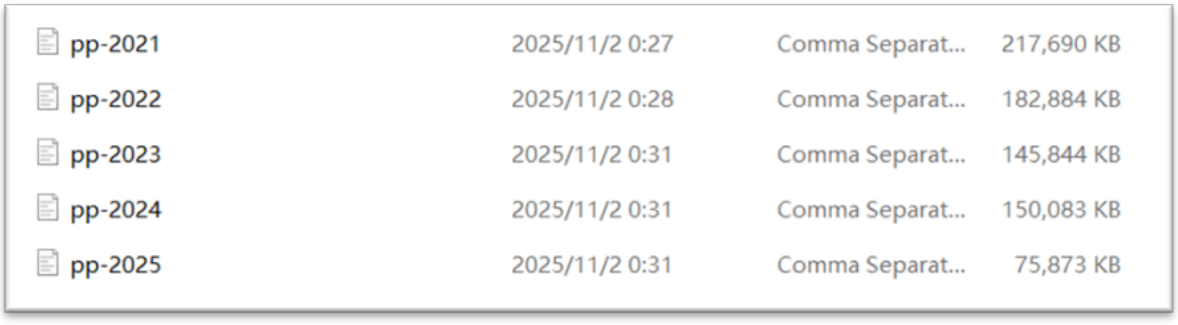

Figure 9: PPD file

Step two: Merge the five-year dataset and filter data specific to the London region. Key steps in the merging programme are illustrated in Figure 10.

```python
# Data column header name
cols = [
    'TransactionID', 'Price', 'Date', 'Postcode', 'PropertyType',
    'NewBuildFlag', 'Duration', 'PAON', 'SAON', 'Street',
    'Locality', 'Town', 'District', 'County', 'PPDCategoryType', 'RecordStatus'
]

# Merge all files
all_data = pd.concat(
    (pd.read_csv(f, header=None, names=cols) for f in csv_files),
    ignore_index=True
)

print(f"\n Merge complete, in total {len(all_data):,} records")

# Save as new file
output_path = os.path.join(folder_path, "merged_2021_2025.csv")
all_data.to_csv(output_path, index=False)

print(f"Save as:{output_path}")

# Filter data from the London area
london_data = all_data[all_data["County"] == "GREATER LONDON"]
london_data.to_csv("london_2021_2025.csv", index=False)

print(f"Save as:{london_data}")
```

Figure 10: merge csv file and get London's data

Step three: Given the excessive size of the existing dataset, sampling is required to efficiently construct the database, ensuring the proportion of data across boroughs remains unchanged before and after sampling. Figure 11 illustrates part of the sampling program. 

Figure 11: sampling function

Step four, the official PPD contains only property postcodes, yet mapping precise locations requires latitude and longitude coordinates. Therefore, the sampled PPD dataset must be merged with a postcode-to-coordinate dataset. This merged dataset forms the source for the database.

Figure 12: merge postcode with coordinate

Finally, the dataset is loaded into MySQL workbench using a .sql file to create the database. 

Figure 13： MySQL workbench transaction table

Website function implementation

1. Homepage

   Figure 14 shows the final appearance of the website's homepage. The central information panel can be divided into two sections: the upper portion provides an introduction to the website, detailing its founding motivation and future vision. The lower section highlights key features, including property price heatmaps, price trend charts, and filters. For first-time users, the "Start" button at the bottom of the information panel serves as the entry point to begin their browsing journey.

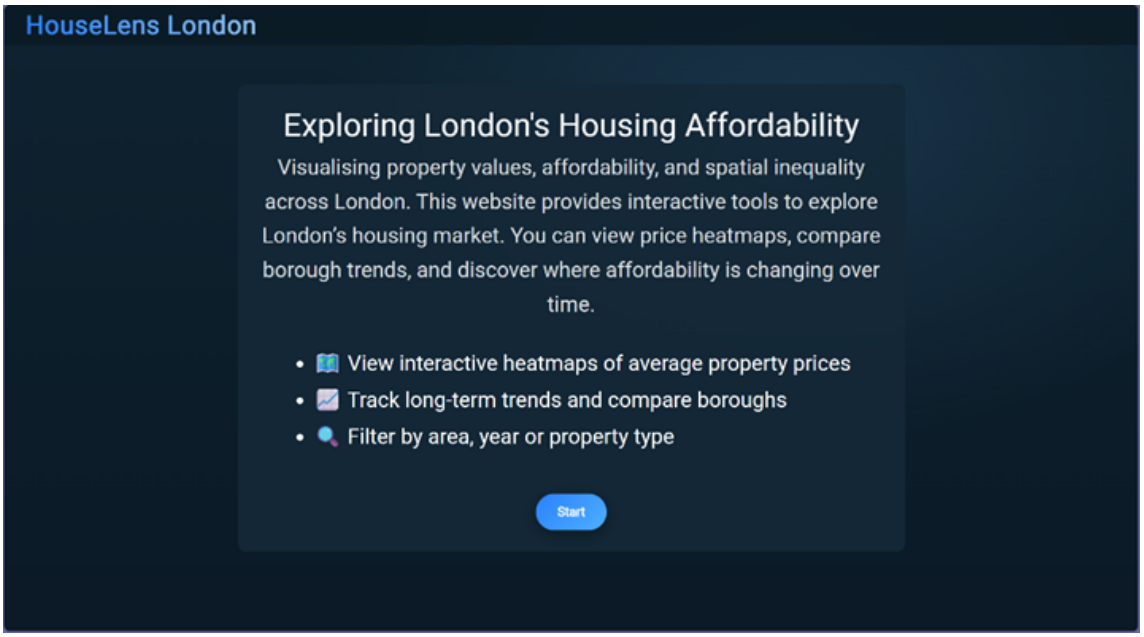

Figure 14: Homepage

2. Heatmap page

Figure 15 is the main page of the website, which uses a heatmap to visually demonstrate the price variations across London's different boroughs. Darker colours indicate higher prices, and users can also determine the average property price for a specific area via the legend in the bottom-left corner. The top menu bar contains two buttons and two dropdown menus. The "Home" button in the top-left corner returns users to the homepage, while the "Reset" button in the top-right resets the map to its initial heatmap state, overriding any user action. Both dropdown menus serve as filters: one for selecting boroughs, the other for price ranges. Notably, the site incorporates a hover information bar feature. When users hover their cursor over a region, the area's name and average property price are displayed.

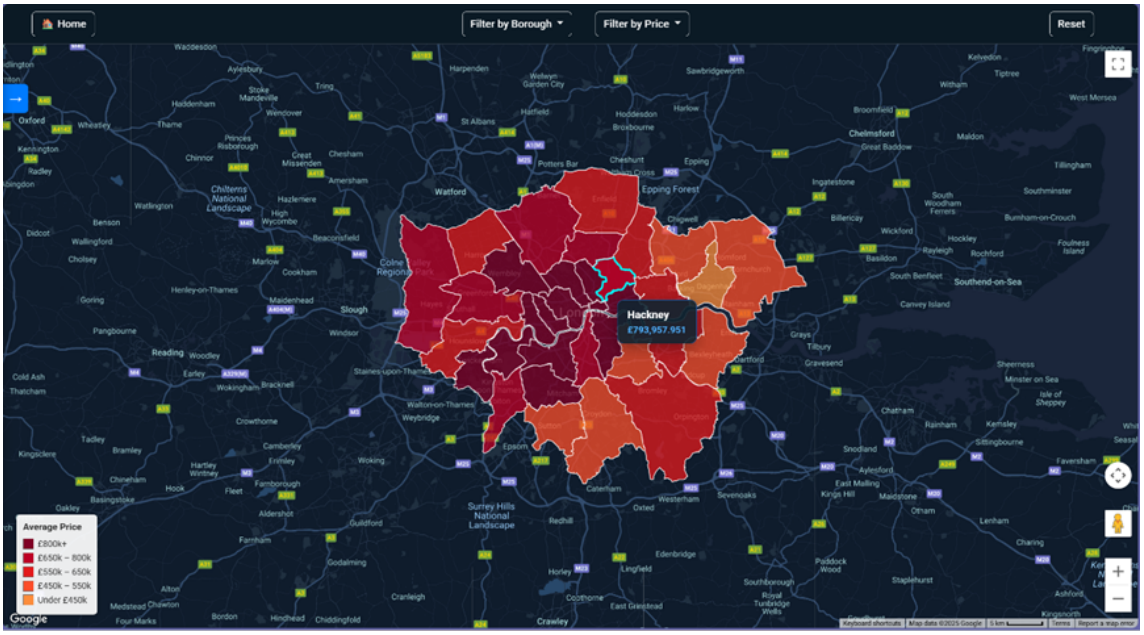

Figure 15: Heatmap page

3. Borough Detail Page

Upon selecting an area on the map as Figure 16, the page will be divided into two parts. A trend line chart pop-up window will appear on the left, with the horizontal axis representing years and the vertical axis showing prices, illustrating the average property price changes in that area over the past five years. The right-hand section will highlight and zoom in on the

selected area, displaying specific property transactions within that borough.

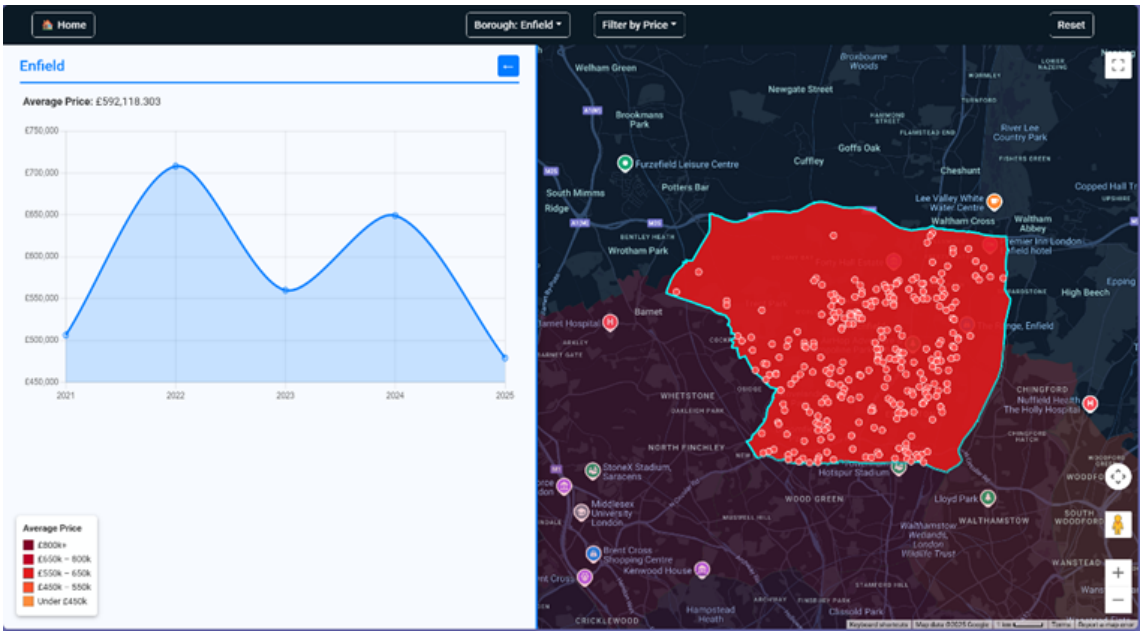

Figure 16: Borough Detail Page

4. Borough Selector

As Figure 17 shows that it is similar with Figure 16. If the user wants to explore a specific area, they can quickly select it via the Borough Selector in the menu bar.

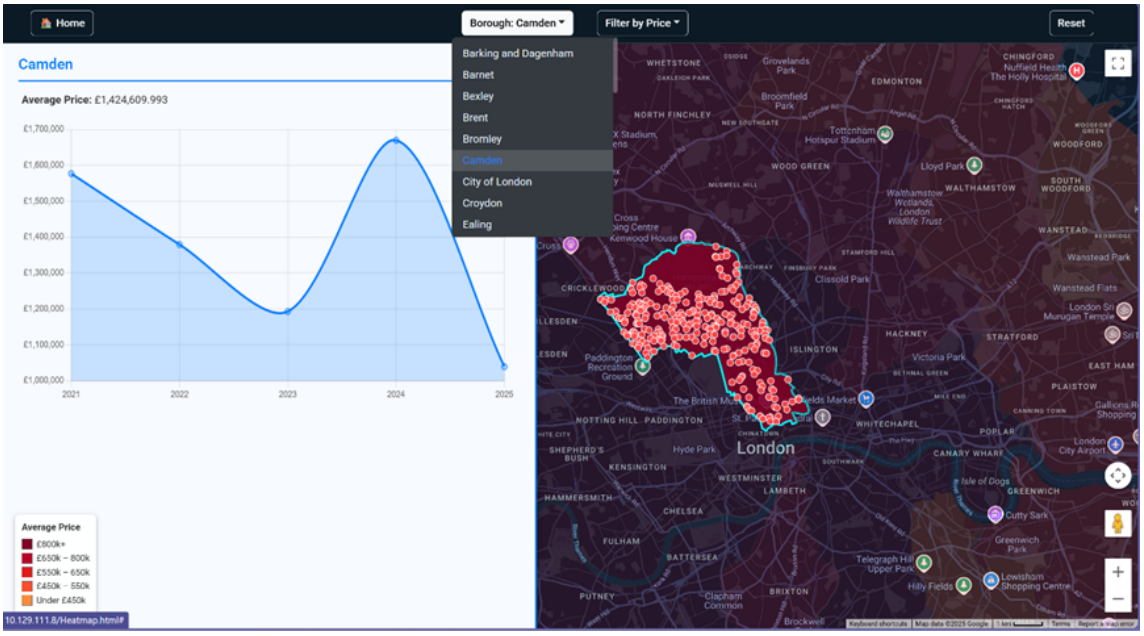

Figure 17: Borough Selector

5. Price filter

If the user is interested in exploring a specific price range, they can utilise the price filter in the menu bar to swiftly locate all boroughs falling within that price interval, with these areas highlighted for emphasis as Figure 18 displays. This functionality provides an intuitive visualisation of the spatial distribution characteristics of London property prices.

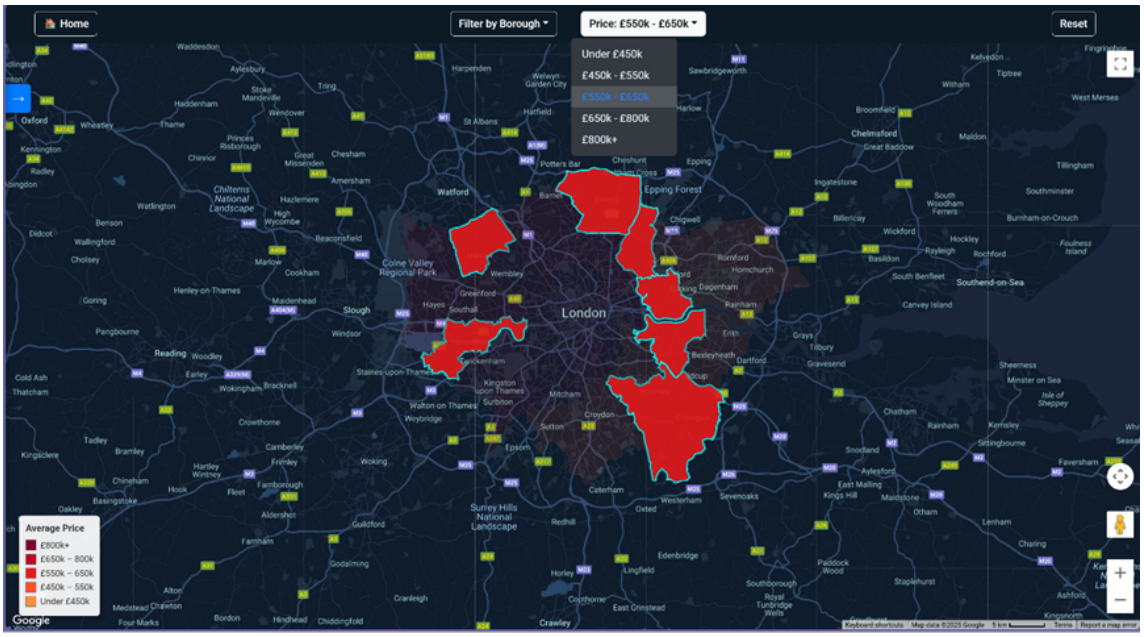

Figure 18: Price filter

6. Property Detail

   Upon selecting a specific area, the map will display detailed property transaction histories. Clicking on a marked point will pop up a detailed information box, including: transaction price, property type, property address, and transaction date. The left-hand pop-up window shows the average property price for that area, providing users with a comparison reference.

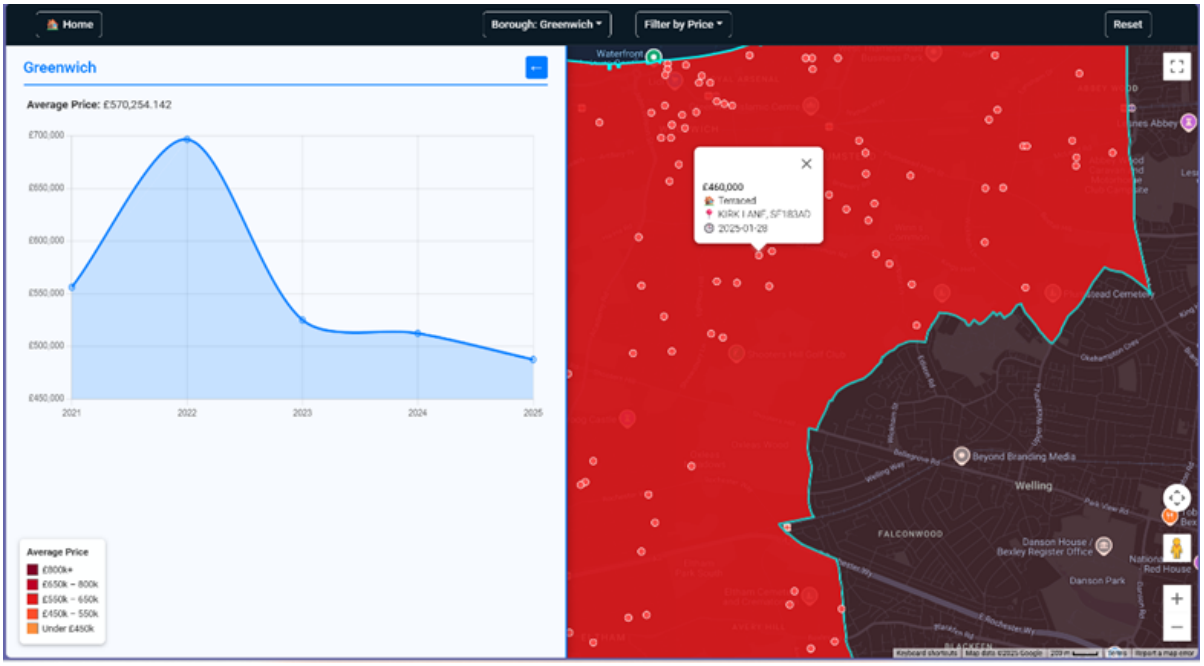

Figure 19: Property Detail

Conclusion

This project has developed a visual interactive platform called HouseLens, which is used to display the trends of the London housing market based on the official housing transaction data of the UK. HouseLens creates a database by merging and cleaning the dataset and has an interactive map front-end interface. The system effectively presents the changes in real housing transactions in each borough of London.

If more time is available, the development team will first consider adding the data type of energy consumption grade to the data set to support the environmental sustainable development initiative of the UK government. Secondly, a machine learning price prediction model will be added to enable the platform not only to provide descriptive information but also to offer predictions for future house prices. These changes will significantly enhance the performance of the website.

Biblography

1. Greater London Authority (GLA)(2023). London Housing Strategy. *Mayor of London* Available at:https://www.london.gov.uk/programmes-strategies/housing-and-land/mayors-priorities-londons-housing-and-land/london-housing-strategy (Accessed: 10 November 2025).
2. Nginx (2024) Nginx web server documentation. *nginx* Available at: https://nginx.org/en/ (Accessed: 10 November 2025).
3. NPM (2024) pm2. Available at: https://www.npmjs.com/package/pm2 (Accessed: 10 November 2025).
4. Wilson, W. and Barton, C. (2022). Early intervention: policy and provision. *House of Commons Library Research Briefing*, CBP-7647. Available at: https://commonslibrary.parliament.uk/research-briefings/cbp-7647/ (Accessed 10 Nov 2025).

Declaration of Authorship

We, Xintong Shao, Tianrui Min and Yewei, Bian, confirm that the work presented in this assessment is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Xintong Shao Tianrui Min Yewei Bian

2025.11.10