HouseHunt: Finding Your Perfect Rental Home

INTRODUCTION:

PROJECT OVERVIEW :

HouseHunt – A Smart Solution for Modern Real Estate Search
HouseHunt is a user-centric real estate platform aimed at transforming the way individuals search for, discover, and connect with properties. Whether for rent or purchase, HouseHunt provides a streamlined, intelligent, and location-based solution to help users find their ideal homes efficiently and confidently.

Purpose of this project:

Searching for suitable housing can be time-consuming, overwhelming, and inefficient, especially in urban areas. Traditional platforms often lack filtering flexibility, real-time updates, and personalized recommendations, leading to missed opportunities and poor user experiences.

IDEATION PHASE:

PROBLEM STATEMENT: In today's fast-paced and technology-driven world, finding a suitable home for rent or purchase can be a time-consuming and frustrating experience. Traditional methods—such as newspaper listings, word-of-mouth, or even fragmented online portals—often lack transparency, personalization, and real- time availability. Users face challenges like limited filtering options, outdated listings, unverified property details, and difficulty in contacting owners or agents.

Empathy Map Canvas

- **Who are we empathizing with?**

1.SAYS:

"I can’t find good listings in my budget."

- "It’s hard to know if a property is real or fake."

"I wish I could contact the owner directly."

- "The photos don’t match the real property."

"I don’t have time to visit multiple places."

2. THINKS:

- "What if this listing is outdated or already rented?"

"I hope the neighborhood is safe."

- "Will the broker charge hidden fees?"

"This platform must have genuine reviews."

- "I want to compare properties easily before deciding."

3. SEES:

Multiple websites with cluttered or outdated listings

- Brokers asking for commission even before property visits

Confusoin g UI/UX on real estate portals

- Lack of filters for real needs (e.g., pet-friendly, Wi-Fi included)

- Inconsistent or missing information (like rent, amenities, rule

5. PAINoS :

Lack of transparency in rent and deposit ter

Verified listing

Brainstorming:

The brainstorming phase is crucial in laying the foundation for a successful House Hunt platform

- It is the stage where Ideas are generated, challanges are identified,and potential solutions are explored with an open mind.

- Understand user needs and pain points.
- Define key features and functions of the platform
- Explore the technology stack best suited for the project.
- T. Think about scalability, user experience, and monetization options.

.

REQUIREMENT ANALYSIS:

Customer Journey Map

Awareness Searches online or hears about House Hunt from a friend or social media “I need a reliable platform to find a house.” Curious, hopeful Use targeted marketing and SEO; offer testimonials and demos
2. Consideration Visits the House Hunt website or installs the app “Is this platform trustworthy and easy to use?” Interested, slightly skeptical Simple UI/UX, quick overview of features, reviews & ratings
3. Registration Signs up or logs into the platform “I hope this doesn’t take too long.” Neutral, cautious Provide social login options; keep the signup process short
4. House Search Enters preferences (location, budget, size) and browses listings “Let me see what matches my needs.” Excited, overwhelmed Smart filters, AI recommendations, map integration

Solution Requirement

Functional Requirements

These are the core features and functionalities that the House Hunt application must offer:

a. User Management

User registration and login (including via email and social platforms)

Role-based access (User, Agent, Admin)

Profile management

Data Flow Diagram

User send search queries, registration details etc. and the system returns property results, visit confirmations,etc

Buyer interacts with the system to search, view, and inquire about properties.

Seller/Agent uses the system to list, update, or remove properties.

All data (users, listings, interactions) is stored in the central database.

Technology Stack

The **HOUSEHUNT** project employs a modern and efficient technology stack:

- **Python** serves as the core programming language.
- **Flask** manages web backend and API routing.
- **Scikit-learn** and **XGBoost** are used for training and deploying machine learning models.

  Pandas and **NumPy** are utilized for data manipulation and preprocessing.
- The **frontend** is designed using **HTML, CSS**, and **Jinja2 templating**, offering a responsive and intuitive user experience.
- **Matplotlib** or **SHAP** can be integrated for visual explanation of model predictions.
- The application runs locally or on cloud platforms and is version-controlled using **Git** and **GitHub** for efficient collaboration and deployment.

PROJECT DESIGN:

Problem Solution Fit

Finding a suitable house for rent or purchase is a time-consuming and frustrating process.
Users often struggle with:

Scattered listings across multiple platforms.

Outdated or inaccurate property information.

Lack of direct contact with sellers or agents.

Limited search filters to match preferences (budget, location, amenities).

Scattered and unreliable listings Centralized platform with verified listings
Poor property search experience Rich filters and smart recommendations
No real-time communication Built-in messaging/contact options
Difficult to manage favorite listings Save, bookmark, and receive alerts on favorites
Lack of transparency Detailed property descriptions with images and data

eliminating unnecessary procedures, the house hunt project fits the real world needs of both buyers and providing a digital transperant,and user friendly

Solution Architecture

The architecture of **Revolutionizing Liver Care** adopts a modular and efficient design tailored for ease of use and scalability:

- Scattered and unreliable listings Centralized platform with verified listings
- Poor property search experience Rich filters and smart recommendations
- No real-time communication Built-in messaging/contact options
- Difficult to manage favorite listings Save, bookmark, and receive alerts on favorites
- Lack of transparency Detailed property descriptions with images and data

PROJECT PLANNING & SCHEDULING:

Project Planning

The development of **HOUSEHUNT PROJECT** was structured into key milestones to ensure clarity, progress tracking, and efficient implementation. The planning process focused on understaning the needs, optimizing model performance, and creating a user-centric application.

Project Development Phases

1. Requirement Gathering & Research

- Gathering the information about the vacancies of houses for needed people

  And research on the project

- This stage focused on understanding the problem space,user needs, system goals,and technical requirements
- Understanding the real life challanges of property searching

2. Model Selection & Training

- Explored multiple machine learning algorithms including **Random Forest**, **Support Vector Machine (SVM)**, and **XGBoost**.
- Performed data preprocessing: handled missing values, normalized features, and encoded categorical data.
- Trained, validated, and tested models using standard metrics such as **accuracy**, **precision**, **recall**, and **F1-score** to select the best-performing model.

3. Frontend Design

- Designed a user-friendly and responsive interface using **HTML**, **CSS**, and **Jinja2** templating.
- Implemented structured forms for entering patient health data and uploading datasets (CSV/Excel).
- Ensured the UI maintained a clean layout, with minimal distractions, cross-browser compatibility, and accessibility on all screen sizes.

4. Backend Integration

- Developed RESTful APIs using **Flask** to handle incoming data, preprocess input features, and generate model predictions.
- Included robust error handling for cases such as incomplete data, invalid formats, or server-side processing errors.
- Integrated model prediction logic seamlessly with the frontend to enable real-time result generation.

5. Testing & Debugging

- Performed **unit testing** for individual components and **integration testing** for full data flow from input to prediction output.
- Evaluated edge cases and ensured the system returned meaningful error messages for invalid user inputs.
- Ensured smooth interoperability between model, backend server, and frontend interface with consistent performance.

6. Deployment Preparation

- Structured the project into modular directories for scalability and maintainability.

- Created comprehensive **README.md** and setup guides to assist users and developers.
- Prepared the application for deployment using platforms like **Heroku**, **Render**, or **Streamlit**, ensuring it runs efficiently in cloud or local environments.

FUNCTIONAL AND PERFORMANCE TESTING

Functional Testing

- **Verified input validation for each field and ensured that incomplete or incorrect data returned descriptive error messages.**
- **Tested model prediction accuracy by comparing known outputs with predicted results across different information.**
- **Handled edge cases, such as:**
    - o **Missing values in gathering data**
    - o **Invalid file formats**
    - o **Corrupted or incomplete input records**

Performance Testing

- The machine learning model produced liver cirrhosis predictions in **under 1 second** for most costumers, confirming fast response times.
- Lightweight models such as **Random Forest** and **XGBoost** were optimized for efficient computation, requiring minimal system resources.
- The **Flask backend** maintained stable performance during prolonged testing periods with no memory leaks or crashes, even under multiple concurrent requests.

User Experience

- The UI remained responsive across browsers like Chrome, Firefox, and Edge.
- Smooth navigation between input and result screens ensured a positive and productive user experience.
- Lightweight design allowed deployment and usage even on lower systems, making it accessible for rural or resource-constrained area

RESULTS:

Output Screenshots

1. Home Page / Landing Page Welcome message or banner
Navigation menu (Home, Login, Register, etc.)

Quick search bar or featured listings

- Screenshot Suggestion: Full page view with visible navigation.

- 2. User Registration Page
Form with fields like name, email, password, user type (buyer/seller)
Submit and validation messages

📌 Screenshot Suggestion: Show a filled-out form ready for submission.

- 3. Login Page
Email & password fields
"Forgot password" option (if available)
Redirect to dashboard upon successful login

Screenshot Suggestion: Logged-in success message or redirected dashboard view.

Result Page:

SearchBar Input for keywords, locations, or property type
Filters Dropdowns or sliders for price, size, type (Rent/Buy), bedrooms, etc.
Property Cards/Grid Display of results in cards showing image, title, price, location, etc.Contact/Saved Buttons to contact seller or add to wishlist
Map Shows pins for properties on a map (Google Maps API)

ADVANTAGES & DISADVANTAGES

Advantages:

- 1. Centralized Property Listings

Provides a single platform where buyers and renters can browse numerous property listings from various sellers and agents.

- 2. Time-Saving

Users can quickly filter and find properties based on specific needs (location, price, type), reducing the effort of offline visits.

3. Real-Time Information

Dyn•amic database updates ensure listings are current, reducing the risk of contacting sellers about unavailable properties.

4. User-Friendly Interface

•

Designed with simple navigation, search functionality, and clear property details that improve the user experience.

5. D•irect Buyer-Seller Communication

Eliminates middlemen by allowing direct chat or inquiry to the property owner

Disadvantages:

- 1. Limited to Internet Users

  Requires stable internet access; offline users or rural populations may be excluded.

- 2. Trust and Verification Issues

  Without strong verification mechanisms, fake or misleading property listings may appear.

- 3. No Legal Support

- Platform may not cover legal aspects of renting or purchasing, requiring users to handle verification and contracts separately.

  4. Maintenance Overhead

  Requires continuous server, database, and security maintenance to prevnt data.

Conclusion

The House Hunt Project successfully addresses the modern challenges faced by individuals in searching for rental or purchasable properties. By leveraging a full-stack web development approach (MERN stack), the platform provides a user-friendly, scalable, and efficient digital solution that connects buyers/renters with property owners and agents in real-time.

Throughout the development process, key functionalities such as user authentication, advanced property search filters, listing management, and direct communication features were implemented to improve the overall user experience. The system architecture ensures data consistency, responsiveness across devices, and secure transactions.

From problem identification to deployment, this project demonstrates how technology can simplify and enhance a traditionally complex process like house hunting. While there are areas for future improvement—such as adding AI-based recommendations, integrating payment gateways, or implementing verification layers—the current solution stands as a solid foundation for digitizing the property discovery experience.

APPENDIX

Dataset Link

Download link: https://www.kaggle.com/datasets/bhavanipriya222/liver-cirrhosis-prediction

GitHub Repository & Project Demo Link:

- **GitHub Repository: https://github.com/487deepthi/househunt.git**