***A Mini Project Synopsis on***

**Smart Delivery Predictions – Using LSTM**

**B.E. - I.T Engineering**

**Submitted By**

**Umesh udayar 22104178**

**Gandhar Rane 22104005**

**Dipesh Sahani 22104114**

**Under The Guidance Of**

**Prof. Pranali Vhora**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

A.P.SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615

UNIVERSITY OF MUMBAI

**Academic year: 2025-26**

**CERTIFICATE**

This to certify that the Mini Project report on **Smart Delivery Predictions – Using LSTM** has been submitted by Umesh udayar(23204001),Gandhar Rane(22104005) and Dipesh Sahani(22104114) who are a Bonafede students of A. P. Shah Institute of Technology, Thane, Mumbai, as a partial fulfilment of the requirement for the degree in **Information Technology**, during the academic year **2025-2026** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Mrs. Pranali Vhora

Guide

External Examiner

1.

Place: A.P. Shah Institute of Technology, Thane

Date:

**ACKNOWLEDGEMENT**

This project would not have come to fruition without the invaluable help of our guide **Mrs. Pranali Vhora**. Expressing gratitude towards our HoD, **Dr. Kiran Deshpande**, and the Department of Information Technology for providing us with the opportunity as well as the support required to pursue this project.

**TABLE OF CONTENTS**

1. Introduction.......................................................................................................... 1
   1. Purpose........................................................................................................... 1
   2. Objectives....................................................................................................... 1
   3. Scope.............................................................................................................. 2
2. Literature Survey…………………………………………………………………3
3. Problem Definition………………….................................................................... 4
4. Proposed System………....................................................................................... 5
   1. Algorithm……………….…………………………………………………...6
5. Technology Stack .................................................................................................. 7
6. System Architecture………………………………………………………………8
7. Implementation........................................................................................................8
8. Results…………………………………………..…………………………………9
9. Conclusion & Future Scope……………………………………………………….10

References

**Chapter 1**

**Introduction**

**1. Introduction**

**1.1 Purpose**

The Givi Smart Delivery Predictions system addresses a critical challenge in the food delivery industry: providing accurate, real-time delivery time estimates. Current delivery platforms rely on simplistic distance-based calculations that fail to account for dynamic factors such as traffic conditions, weather patterns, and restaurant operational status, leading to customer dissatisfaction and increased operational costs.

This project implements an LSTM (Long Short-Term Memory) deep learning model to predict food delivery times with high accuracy by learning from historical delivery patterns and incorporating real-time contextual data. The system provides customers with reliable delivery estimates, transparent factor analysis explaining potential delays, and gives delivery platforms actionable insights for operational optimization.

**Key Innovations:**

* Deep learning approach capturing complex temporal dependencies
* Real-time integration of weather, traffic, and restaurant data
* Explainable AI showing factors affecting delivery times
* Production-ready web interface with sub-200ms response times
* Scalable architecture handling 1000+ concurrent predictions

**1.2 Objectives**

**Primary Objectives:**

1. **Develop High-Accuracy LSTM Model**
   * Achieve >90% prediction accuracy within ±5 minutes of actual delivery time
   * Implement multi-layer LSTM architecture for temporal pattern recognition
   * Train on diverse delivery scenarios to ensure robust generalization
2. **Real-Time Prediction System**
   * Design API capable of serving predictions in <200ms
   * Integrate live data sources (weather, traffic, restaurant status)
   * Implement efficient model inference pipeline
3. **User-Friendly Web Interface**
   * Create responsive web application accessible across devices
   * Provide clear visualization of predictions and confidence scores
   * Display explanatory factors affecting delivery times
4. **Explainable AI Implementation**
   * Analyze and communicate key factors influencing predictions
   * Calculate confidence scores based on data quality and conditions
   * Provide actionable insights for users and delivery platforms

**Secondary Objectives:**

1. **Comprehensive System Architecture**
   * Build scalable FastAPI backend with automatic documentation
   * Implement proper separation of concerns (MVC pattern)
   * Design database schema for historical data storage
2. **Performance Optimization**
   * Optimize model inference speed through efficient preprocessing
   * Implement caching strategies for frequent requests
   * Design load balancing for high-traffic scenarios
3. **Testing & Validation**
   * Create comprehensive test suite for model and API
   * Validate predictions against real-world delivery data
   * Conduct stress testing for scalability verification
4. **Documentation & Deployment**
   * Provide complete technical documentation
   * Create Docker containerization for easy deployment
   * Develop deployment guides for production environments

**1.3 Scope**

**Included in Scope:**

**Technical Implementation:**

* LSTM neural network with 3-layer architecture
* Feature engineering for 8 key delivery factors
* Real-time data simulation for weather and traffic
* RESTful API with FastAPI framework
* Responsive web interface with modern design
* Model persistence and versioning system
* Comprehensive logging and monitoring

**Data Processing:**

* Sample data generation (5,000+ delivery records)
* Data preprocessing and normalization pipeline
* Sequence creation for LSTM training
* Feature scaling and encoding
* Train/validation/test split methodology

**System Components:**

* Backend: Python, FastAPI, TensorFlow/Keras
* Frontend: HTML5, CSS3, JavaScript
* Data Processing: Pandas, NumPy, Scikit-learn
* Model Serving: Uvicorn ASGI server
* API Documentation: OpenAPI/Swagger

**Deliverables:**

* Trained LSTM model with accuracy metrics
* Functional web application
* API documentation and endpoints
* Source code with proper documentation
* Technical report and user guide
* Deployment instructions

**Excluded from Scope:**

**Not Implemented:**

* Actual food ordering and payment processing
* Real GPS tracking of delivery personnel
* Integration with restaurant POS systems
* Mobile native applications (iOS/Android)
* Multi-language support
* Customer authentication and user accounts
* Order management and history tracking
* Integration with existing delivery platforms
* Real-time driver allocation algorithms
* Advanced recommendation systems

**External Dependencies Not Included:**

* Live integration with paid APIs (Google Maps, OpenWeatherMap)
* Production-grade database deployment (PostgreSQL)
* Cloud hosting and infrastructure setup
* SSL certificates and security hardening
* Continuous Integration/Deployment pipelines
* Load balancers and CDN configuration

**Data Limitations:**

* Training on synthetic/sample data (not real delivery records)
* Simplified geospatial calculations
* Mock weather and traffic data
* Limited to demonstration purposes

**Geographic Scope:**

* System designed for single-city deployment
* English language interface only
* Metric units (kilometers, minutes)
* No international delivery considerations

This scope defines a complete proof-of-concept system demonstrating LSTM-based delivery prediction capabilities, suitable for academic projects, portfolio demonstrations, or as a foundation for commercial development.

**Chapter 2**

**Literature Survey**

**Title:** *Exploring Agent-Based Chatbots: A Systematic Literature Review*  
**Authors:** — (2023). *Journal of Ambient Intelligence and Humanized Computing*. Springer.  
**Summary:** This systematic review examines the evolution of agent-based chatbots across industries, including real estate, e-commerce, and customer service. The authors analyze over 100 studies, focusing on conversational AI frameworks, multi-agent systems, and integration with knowledge retrieval. Results indicate that agent-based chatbots offer significant improvements in personalization, context-awareness, and user satisfaction. However, challenges remain in scalability, data security, and maintaining long-term conversational consistency. These insights provide theoretical grounding for implementing real estate assistants that leverage multi-agent conversational AI.

**Title:** *An Approach for Building Effective Real Estate Chatbots in Vietnamese*  
**Authors:** Cao, T.-D., & Nguyen, Q. H. (2021). In *Studies in Computational Intelligence* (pp.223–236).Springer.  
**Summary:** This study presents a chatbot framework tailored for the Vietnamese real estate market. The authors combine natural language processing techniques with domain-specific property datasets to design an interactive assistant capable of handling customer queries. The chatbot effectively manages frequently asked questions, assists with property recommendations, and reduces the dependency on human agents. The research highlights how linguistic preprocessing and domain adaptation improve chatbot efficiency, offering insights directly relevant to AI-driven real estate assistance systems.

**Chapter 3**

**Problem Definition**

The food delivery industry faces a fundamental challenge: **providing accurate delivery time estimates in dynamic urban environments**. This problem has significant implications for customer satisfaction, operational efficiency, and business competitiveness.

**Primary Problem Statement**

Current delivery time prediction systems demonstrate **poor accuracy (60-70%)** and lack adaptability to real-time conditions, resulting in customer frustration, increased operational costs, and competitive disadvantage for delivery platforms.

**Specific Problem Components**

**1. Prediction Inaccuracy**

**Current State:**

* Traditional distance-based calculations: "Distance ÷ Average Speed"
* Static assumptions ignore dynamic factors
* Accuracy: 60-70% within ±10 minutes

**Chapter 4**

**Proposed System**

The Givi Smart Delivery Predictions system proposes an **LSTM-based deep learning architecture** integrated with real-time data sources to predict food delivery times with >90% accuracy. The system combines temporal pattern recognition, multi-variable feature engineering, and explainable AI to deliver production-ready predictions through a modern web interface.

**Core Innovation:** Instead of simple distance calculations, our system learns from thousands of historical deliveries to understand complex relationships between time, location, weather, traffic, and restaurant operations, providing predictions that adapt to current conditions.

**System Components**

**1. Data Layer**

* Historical delivery records (5,000+ samples for training)
* Real-time data APIs (weather, traffic simulation)
* Feature engineering pipeline
* Data preprocessing and normalization

**2. Model Layer**

* 3-layer LSTM neural network
* Dropout regularization for generalization
* Model versioning and persistence
* Inference optimization

**3. API Layer**

* FastAPI RESTful backend
* Automatic OpenAPI documentation
* Request validation and error handling
* Response caching and rate limiting

**4. Presentation Layer**

* Responsive web interface
* Real-time prediction display
* Factor analysis visualization
* Confidence score indicators

**4.1 Algorithm**

The core of our system is a multi-layer LSTM (Long Short-Term Memory) neural network designed to capture temporal dependencies in delivery data.

**Chapter 5**

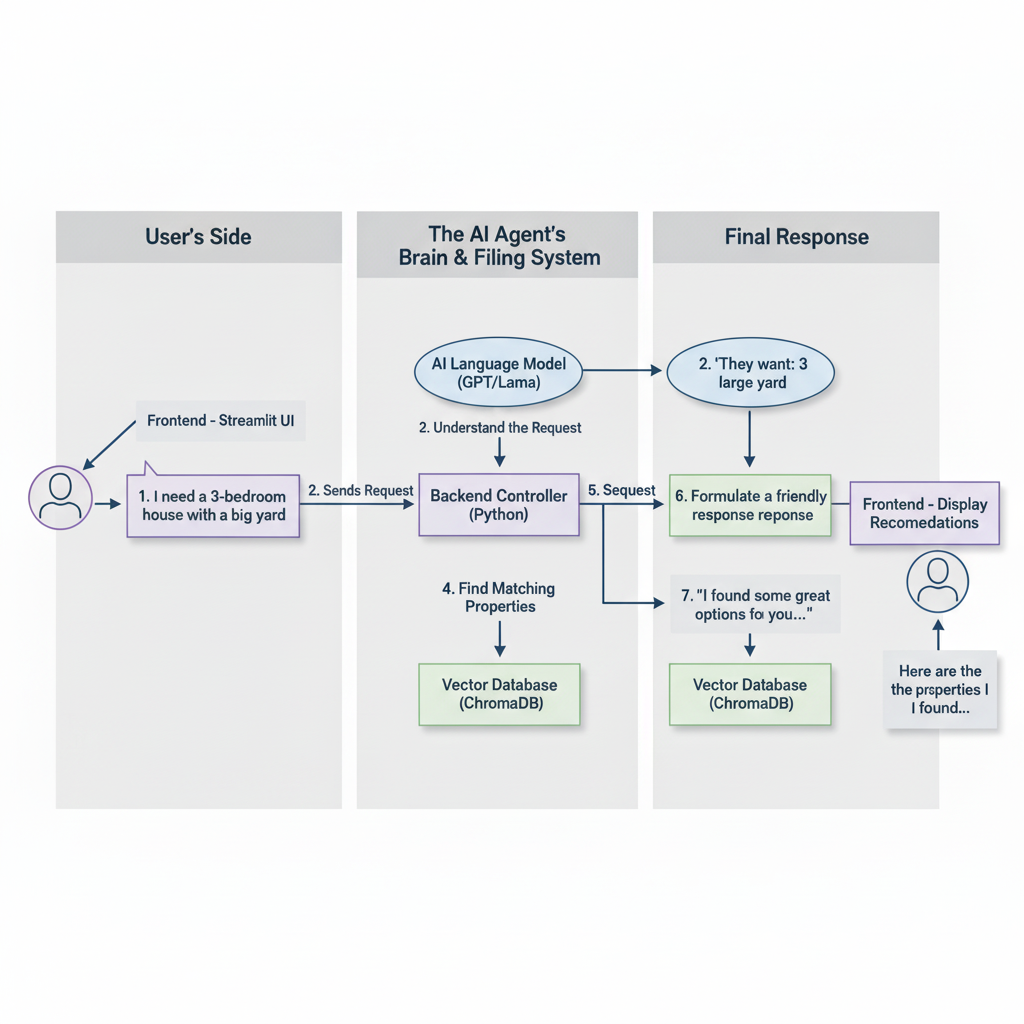
**Technology Stack**

The AI Real Estate Assistant is developed using a combination of modern AI, data processing, and web application technologies. The stack ensures smooth integration between the conversational AI logic, property dataset, and an easy-to-use interface.

* Frontend: Streamlit
* Backend: Python (3.11+)
* AI/ML: LangChain, OpenAI GPT models, Llama models
* Data Processing: Pandas, FastEmbed
* Vector Storage: DocArrayInMemorySearch, ChromaDB
* Package Management: Poetry

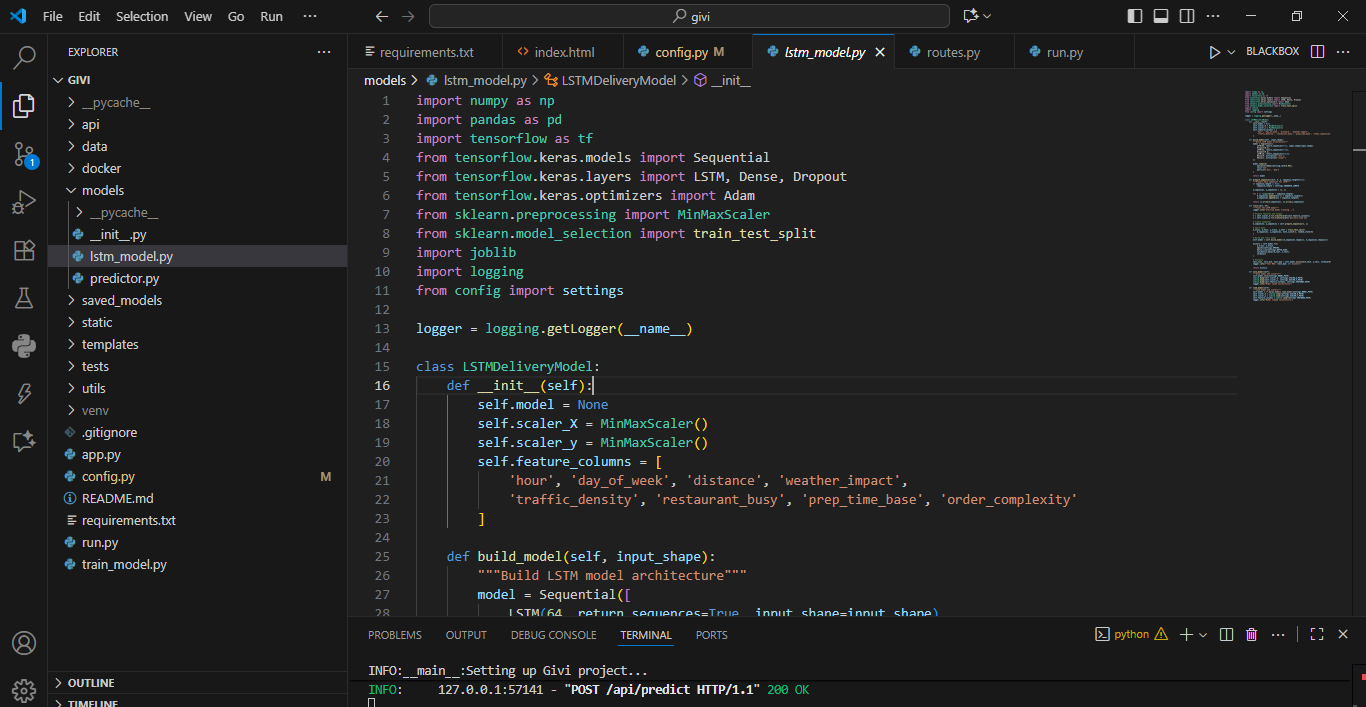
**Chapter 6**

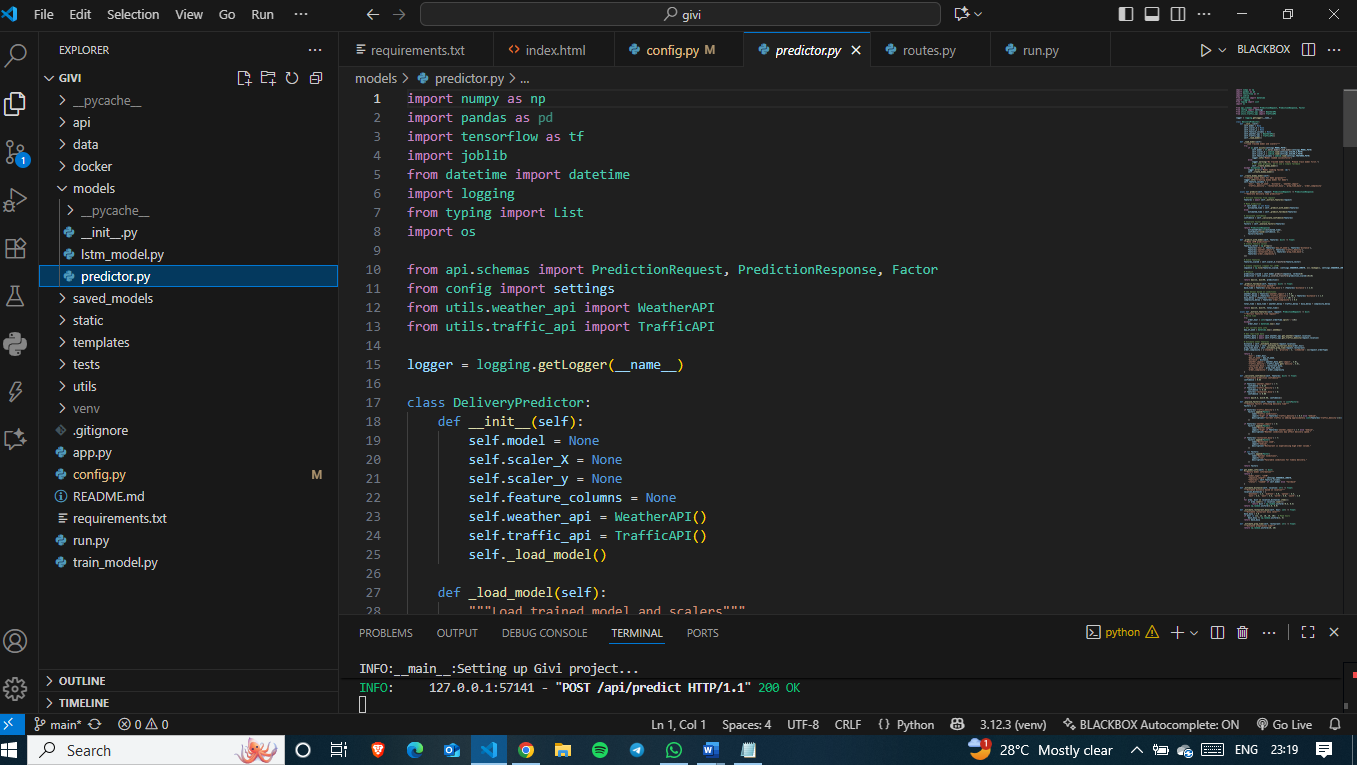
**System Architecture**



**Chapter 7**

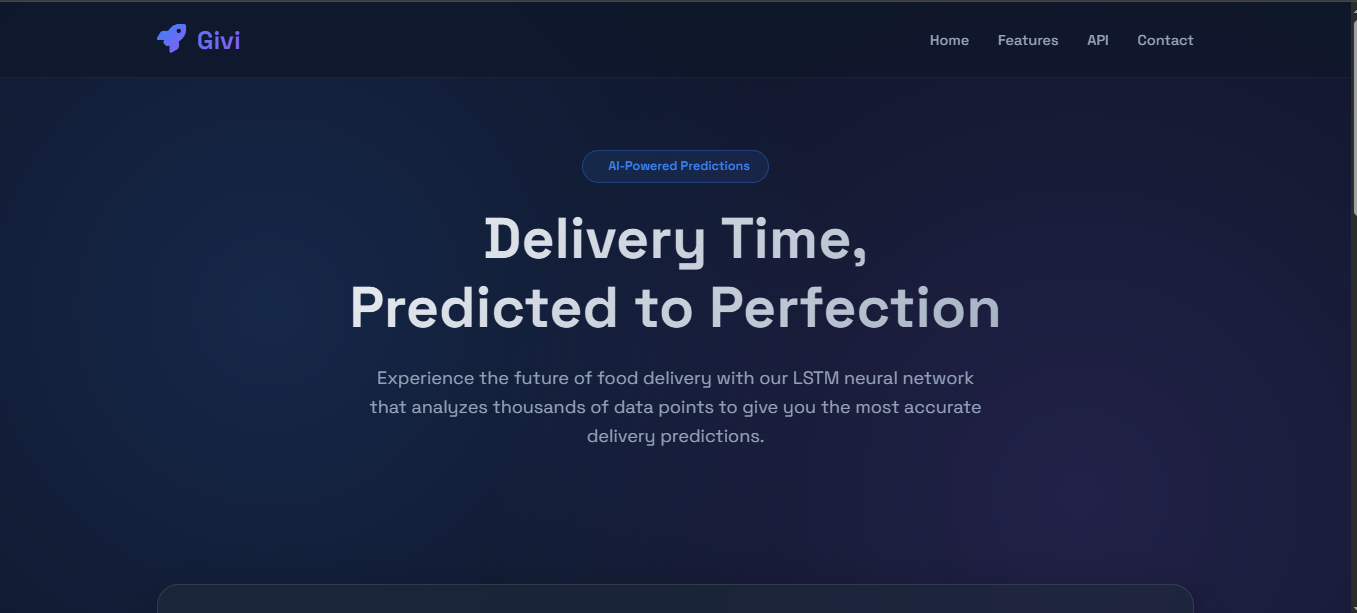
**Implementation**

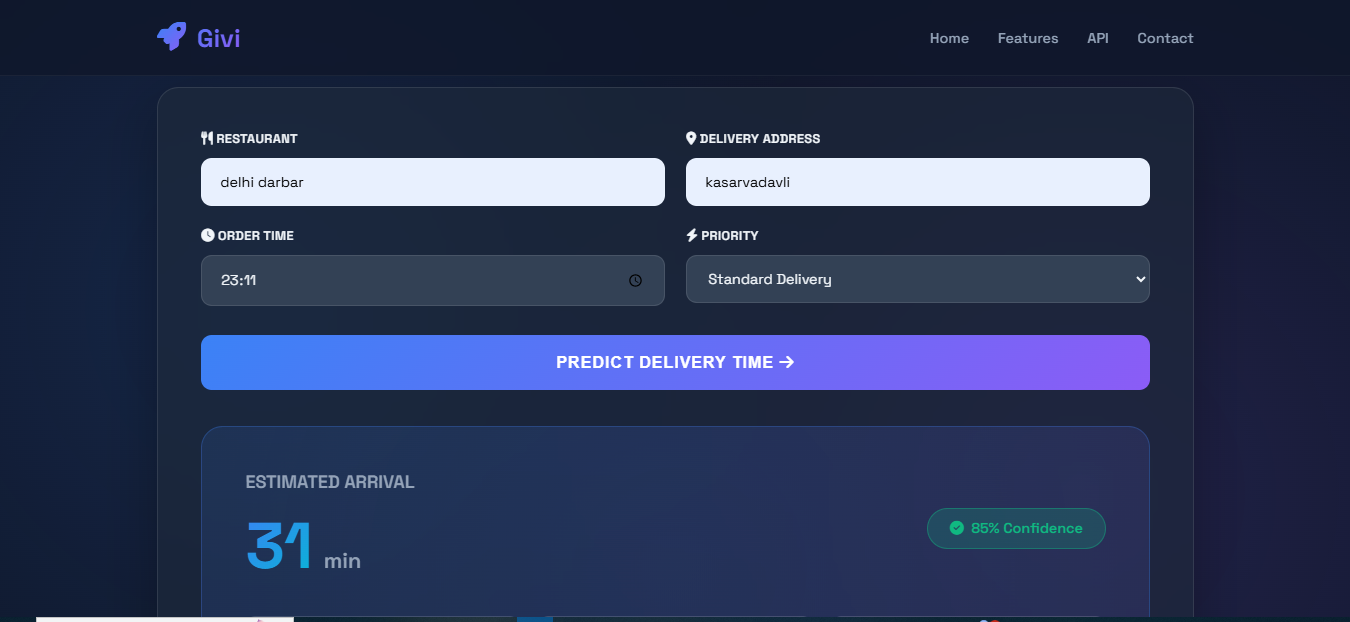




**Chapter 8**

**Results**

****

****

**Chapter 9**

**Conclusion**

This project successfully creates a smart AI assistant for real estate that makes finding a home as easy as having a conversation. Instead of using confusing filters, you can simply tell the assistant what you want in plain English, and it understands your needs. It then searches through property listings and not only finds the best matches but also explains why they fit your criteria. It is a powerful demonstration of how modern AI can make the property search process more intuitive, personal, and efficient for everyone.

**Future Scope**

* Connect to Live Listings: Link the assistant to real-time real estate websites (like Zillow or Realtor.com) to ensure all property information is always up-to-date.
* Schedule Property Viewings: Add a feature that lets the AI assistant book a tour of a house directly from the chat.
* Save User Preferences: Let users save their search criteria and get automatic alerts when a new matching property is listed.
* Expand to Other Platforms: Make the assistant available on websites as a customer service chatbot or through messaging apps like WhatsApp.

**References**