Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	3 February 2026
Team ID	LTVIP2026TMIDS80267
Project Name	Visualizing Housing Market Trends
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

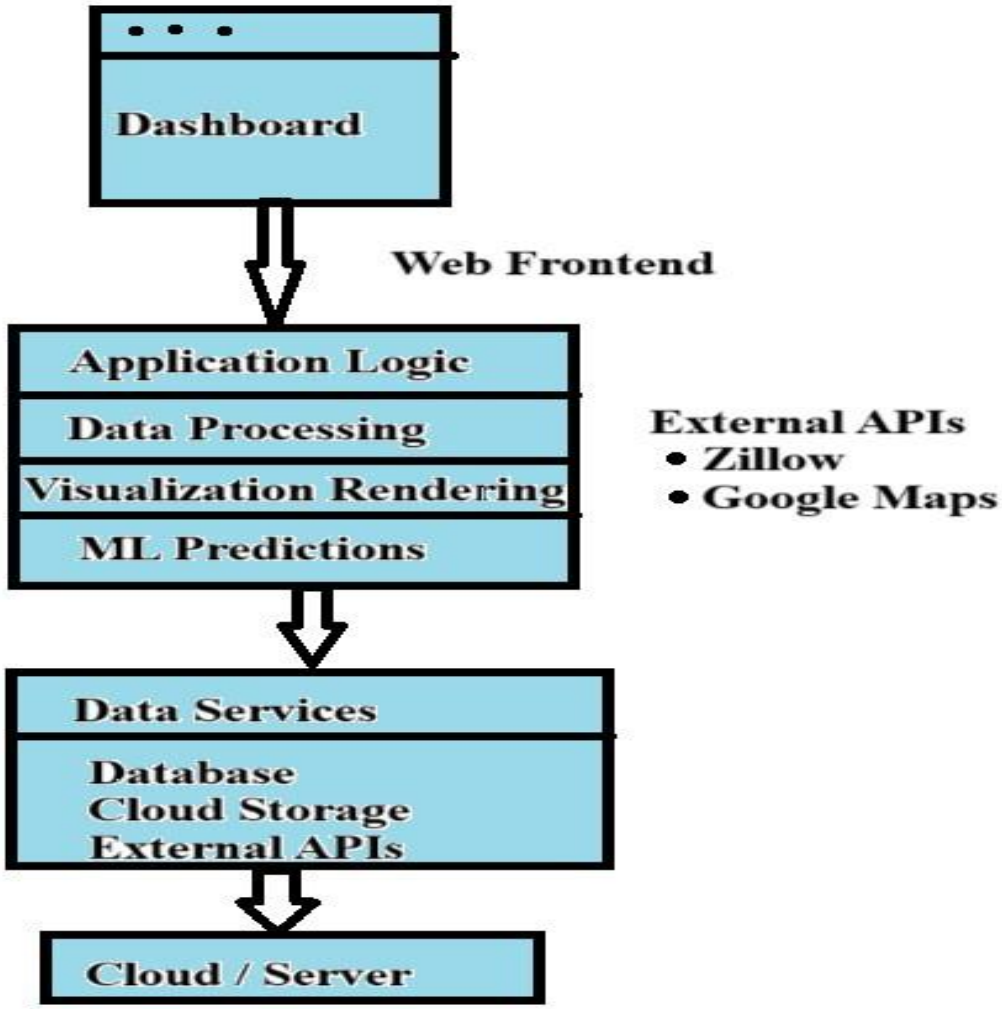
Dashboard
Web Frontend
Application Logic
Data Processing
Visualization Rendering
ML Predictions
External APIs
• Zillow
• Google Maps
Data Services
Database
Cloud Storage
External APIs
Cloud / Server

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web-based dashboard for data visualization	HTML, CSS, JavaScript, React.js / Angular
2.	Application Logic-1	Data preprocessing, aggregation	Python (Pandas, NumPy)
3.	Application Logic-2	Visualization generation & rendering logic	Tableau / D3.js / Plotly
4.	Application Logic-3	Machine Learning model for price prediction (if any)	Python (scikit-learn, XGBoost)
5.	Database	Store housing data (price, location, features)	MySQL / PostgreSQL
6.	Cloud Database	Scalable hosted database	AWS RDS / Firebase / Google Cloud SQL
7.	File Storage	Store raw datasets and files	AWS S3 / Google Cloud Storage / Local storage
8.	External API-1	Get real-time housing or pricing data	Zillow API / RapidAPI Real Estate APIs
9.	External API-2	Geolocation & mapping integration	Google Maps API
10.	Machine Learning Model	Predict housing prices based on features	Trained regression model in Python
11.	Infrastructure (Server / Cloud)	Deployment of app on cloud or local	Local server / Heroku / AWS / GCP / Docker

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frontend & backend frameworks	React.js, Flask, Django, Pandas, scikitlearn
2.	Security Implementations	Access controls, user authentication	JWT, OAuth2.0, HTTPS, Role-based Access
3.	Scalable Architecture	Use of microservices for backend and visualization layers	Docker, REST APIs, Kubernetes (optional)
4.	Availability	Deployed on scalable cloud platforms, load balancing via reverse proxy (if applicable)	AWS Load Balancer, Nginx
5.	Performance	Caching frequently accessed data, using optimized queries	Redis, CDN, Indexed DB Queries

References:

https://c4model.com/ https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/ https://www.ibm.com/cloud/architecture https://aws.amazon.com/architecture https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d