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

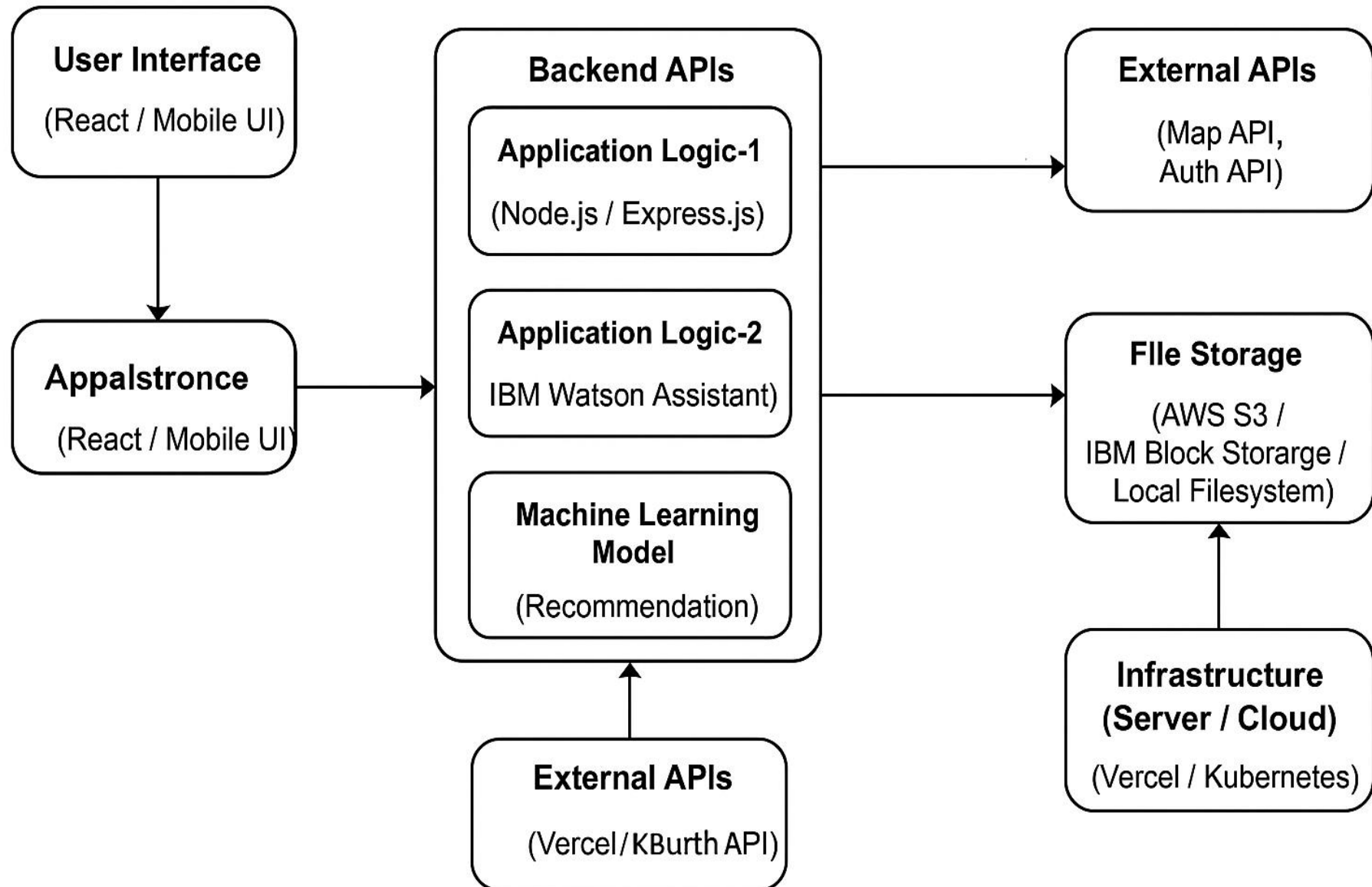| Date	28 June 2025
Team ID	LTVIP2025TMID49487
Project Name	House hunt: finding your perfect rental home
Maximum Marks	4 Marks

Technical Architecture:

HouseHunt

User Interface

(React / Mobile UI)

Appalstronce

(React / Mobile UI)

Backend APIs

Application Logic-1

(Node.js / Express.js)

Application Logic-2

IBM Watson Assistant)

Machine Learning Model

(Recommendation)

External APIs

(Map API,
Auth API)

FIle Storage

(AWS S3 /
IBM Block Storarge /
Local Filesystem)

External APIs

(Vercel / KBurth API)

Infrastructure
(Server / Cloud)

(Vercel / Kubernetes)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Frontend web/mobile UI.	HTML, CSS, JavaScript, React.js
2.	Application Logic-1	Backend business logic for user/property ops	Node.js / Express.js
3.	Application Logic-2	Text-based search and recommendation engine	Python + Scikit-learn
4.	Application Logic-3	Conversational assistant (future enhancement)	IBM Watson Assistant (optional)
5.	Database	Stores user, property, booking info	MongoDB (NoSQL)
6.	Cloud Database	Cloud-hosted DB	MongoDB Atlas / IBM Cloudant
7.	File Storage	Property images / documents	AWS S3 / IBM Block Storage / Local FS
8.	External API-1	Location/Map-based search	Google Maps API
9.	External API-2	Government ID verification (optional)	Aadhaar e-KYC API
10.	Machine Learning Model	Property recommendation engine	Python (Scikit-learn / TensorFlow Lite)
11.	Infrastructure (Server / Cloud)	Cloud deployment environment	Vercel (Frontend), Heroku / Render (BE), Kubernetes (Optional)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frontend and backend libraries and tools	React.js, Express.js, Node.js, MongoDB
2.	Security Implementations	Access control, data protection, API security	JWT Auth, HTTPS, bcrypt , CORS, OWASP
3.	Scalable Architecture	Tiered architecture with cloud support for scale	3-tier with load balancer and microservices
4.	Availability	High availability using distributed cloud hosting	Cloud Load Balancer, Multi-Region MongoDB
5.	Performance	Optimized REST APIs, Lazy loading, CDN for assets, Cache for search	Redis (optional), CDN (Cloudflare), Axios

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