

## A) Technological Requirements

Technology	Choice	Pros	Cons
<b>Programming Language</b>	JavaScript	<b>- Full-Stack Consistency:</b> JavaScript is used both on the front end (React) and back end (Node.js).	<b>- Single Language Learning Curve:</b> Developers need to be proficient in JavaScript for both front-end and back-end development.
		<b>- Wide Adoption:</b> JavaScript is one of the most widely used languages, fostering a large and active developer community.	
<b>Database</b>	MongoDB	<b>- Flexible Schema:</b> MongoDB, being a NoSQL database, allows for a flexible	<b>- Limited Support for Complex Queries:</b> MongoDB may not be as suitable

		schema.	for complex queries compared to relational databases.
		<b>- Scalability:</b> MongoDB is horizontally scalable, making it suitable for handling growing amounts of data.	<b>- No ACID Compliance:</b> MongoDB sacrifices some aspects of ACID compliance for flexibility, which might be a concern for certain applications.
		<b>- Document-Oriented:</b> Well-suited for document-oriented data structures, making it intuitive for representing rental-related information.	<b>- Schema Design Challenges:</b> Requires careful planning of data models due to the lack of a rigid schema.
<b>Back-End</b>	Express.js	<b>- Lightweight and</b>	<b>- Less Opinionated:</b>

<b>Framework</b>		<b>Fast:</b> Express.js is a minimal and flexible Node.js web application framework.	Some developers may find the framework's minimalism to be less opinionated, requiring more decisions during development.
		- <b>Middleware Support:</b> Express.js provides robust middleware support, enhancing the extensibility of the application.	- <b>Limited Built-In Features:</b> Compared to some full-fledged frameworks, Express.js requires additional libraries for certain features.
		- <b>Active Development:</b> Regular updates and a large community ensure the framework stays current and well-	

		supported.	
<b>Front-End Library</b>	React	<b>- Component-Based Architecture:</b> React's component-based architecture promotes reusability and maintainability of code.	<b>- Learning Curve:</b> Developers need to learn React's component-based paradigm, which might pose a learning curve initially.
		<b>- Virtual DOM for Efficiency:</b> React's virtual DOM enhances application performance by minimizing actual DOM manipulations.	<b>- JSX Syntax:</b> JSX might seem unusual at first, especially for developers used to traditional HTML.
		<b>- Large Ecosystem:</b> A vast ecosystem of libraries and community-driven	<b>- Boilerplate Code:</b> Setting up a React project can involve some boilerplate code,

		components simplifies development.	though tools like Create React App alleviate this.
<b>Back-End Runtime</b>	Node.js	- <b>Single Language Stack:</b> Node.js enables full-stack JavaScript development.	- <b>Callback Hell (Callback Pyramids):</b> Asynchronous callbacks may lead to complex nested structures (Callback Hell).
		- <b>Scalability:</b> Node.js is known for its ability to handle a large number of concurrent connections efficiently.	- <b>Young Ecosystem:</b> Being relatively newer, some mature enterprise features might be less established.
		- <b>Package Manager (npm):</b> npm simplifies package management,	

		making it easy to integrate third-party libraries.	
<b>DB-App Framework</b>	Mongoose	<b>- Simplified MongoDB Interaction:</b> Mongoose is an ODM (Object Data Modeling) library that simplifies MongoDB interactions.	<b>- Learning Curve for Schema Definitions:</b> Developers need to learn Mongoose's schema definitions, especially if they are new to MongoDB.
		<b>- Middleware Support:</b> Hooks for everything - pre and post-save, queries, validations. Mongoose is giving us control.	<b>- Potential Overhead:</b> Mongoose introduces some overhead, and for simple use cases, direct MongoDB interaction might be more suitable.
		<b>- Schema Validation:</b>	<b>- Complexity for</b>

		Mongoose enables schema-based validation, preventing inconsistent data storage.	<b>Simple Projects:</b> For smaller projects, the added complexity of Mongoose might be unnecessary.