*République Tunisienne*
*****
*Ministère de l'Enseignement Supérieur et de la Recherche Scientifique*
*****
*Université de Monastir*
*****
*Institut Supérieur d'Informatique et de Mathématiques de Monastir*
*****
*Département Informatique*

**N° d'ordre : L**INFO**

# Project Memory End of Studies

***Presented with a view to obtaining the***

National Bachelor of Science Diploma

Computer Science

***Speciality :***

Software and Information System Engineering

***By***
***Jaziri Ahmed***

## Development of a mobile application and web back office dedicated to real estate investment

***Defended on *** in front of the jury composed of:***

| | |
|---|---|
| **Mr :** | **President** |
| **Mr :** | **Reporter** |
| **Ms. : Bali Nadia** | **Educational Supervisor** |
| **Mr. : ZOUARI Khalil** | **Technical Supervisor** |

## Confidentiality Note

Some information in this report has been redacted for confidentiality reasons. Thank you for your understanding.

# SUMMARY

This work is part of the completion of our Final Year Project at the Higher Institute of Computer Science and Mathematics of Monastir for the academic year 2024-2025, aiming for the National Fundamental License Diploma in Computer Science. Conducted within the company **'KZ IT Services'**, our main objective is the development of a mobile application and a web back-office dedicated to real estate investment named **'KORPOR'**. We used MySQL to manage the databases, Express-Node.js to implement the backend, and React and Vite to implement the frontend. Project management followed the SCRUM Agile methodology, emphasizing agile practices such as sprint planning, sprint management, and regular meetings.

**Keywords:** Blockchain Technology, AI, MySQL, Express-Node.js, React, Vite.

# ABSTRACT

This project is part of our graduation project at the Higher Institute of Computer Science and Mathematics of Monastir for the 2024-2025 academic year, leading to the national diploma of fundamental license in Computer Science. Carried out within the company **"KZ IT Services"**, our main objective is the development of a mobile application and web back office dedicated to real estate investment called **"KORPOR"**. We used MySQL to manage the databases, Express-Node.js for the backend implementation, and React and Vite for the frontend. The project management followed the SCRUM Agile methodology, with an emphasis on agile practices such as sprint planning, sprint management, and regular meetings.

**Keywords:** Blockchain Technology, AI, MySQL, Express-Node.js, React, Vite.

# Dedication

*To the memory of my beloved father, whose guidance and wisdom continue to light my path.*
*Though no longer with us, your presence remains in every achievement of my life.*

*To my loving mother, whose strength and endless support shaped who I am today.*

*To my sister and brother, whose companionship and encouragement have been constant sources of joy and motivation.*

*To my little Aryouma, whose innocence and love bring happiness to our family every day.*

*To Mme Nadia, my professors and mentors, who have guided me with knowledge and patience throughout my academic journey.*

*To my friends, whose encouragement made this journey worthwhile.*

*This work is dedicated to all of you, but especially to you, Father.*

# Acknowledgement

*With profound gratitude, I acknowledge those whose wisdom and support were instrumental to the Korpor platform's development:*

### Academic Committee

**Prof. [Name]** & **Dr. [Name]** — for their invaluable evaluation and guidance.

### Supervisor

**Dr. Nadia Bali** — whose mentorship illuminated the path through complex technological challenges.

### Company Leader

**Mr. Khalil Ezouiri** — for his visionary leadership and providing the opportunity to contribute to this innovative platform.

### Institution

Faculty and staff who fostered an environment of innovation and excellence.

This work stands as testament to the transformative potential of integrating blockchain security with AI-driven insights to democratize investment opportunities.

*With appreciation,*
*Ahmed Jaziri*

April 24, 2025

# TABLE OF CONTENTS

# LIST OF FIGURES

*"The future of real estate investment lies at the intersection of blockchain security, AI-driven insights, and crowdfunding accessibility. When we remove barriers and democratize opportunity through technology, we don't just change markets—we transform lives and build communities."*

— Marc Andreessen

Co-founder of Andreessen Horowitz & Tech Investment Visionary

# General Introduction

In today's rapidly evolving financial landscape, traditional investment methods are often burdened by opaque processes, cumbersome bureaucracy, and significant entry barriers. Investors have long struggled with outdated systems that impede transparency, elevate risks, and complicate access to promising opportunities. Such challenges not only limit diversification but also expose users to uncertainties that modern technology can easily overcome.

**Korpor** was conceived to transform this paradigm by delivering a fully integrated, AI and blockchain-powered mobile investment platform. By harnessing advanced data analytics, machine learning, and cutting-edge blockchain technology, Korpor streamlines every facet of the investment process. The application offers a seamless user onboarding experience, intuitive project listings enriched with AI-driven recommendations, and a secure, automated investment flow that simplifies transactions while ensuring that every operation is recorded immutably on the blockchain. Investors can manage their portfolios effortlessly through a comprehensive dashboard, with real-time notifications, an interactive AI chatbot, and multi-language support delivering a personalized and globally accessible experience.

**Security and trust** are at the heart of Korpor's design. By employing state-of-the-art encryption, blockchain-based transparency, and strict compliance measures, the platform safeguards sensitive financial data and guarantees that every transaction is executed within a secure and verifiable framework. Continuous monitoring, performance optimization, and the immutable nature of blockchain records further ensure that the application remains resilient, scalable, and resistant to fraud in a dynamic market environment.

Developed under a flexible Agile framework that combines iterative development with strategic project management best practices, Korpor is designed to rapidly adapt to evolving market trends and user needs. This methodical approach allows for regular feedback, swift enhancements, and the seamless integration of innovative features throughout the development lifecycle.

### Document Structure

- The first chapter, **Project Context**, delves into the industry challenges and the vision that inspired Korpor's creation.
- The second chapter, **Analysis and Specification of Needs**, outlines the comprehensive requirements gathering, needs analysis, architectural design, and the selection of cutting-edge tools and technologies.
- Subsequent chapters document the progressive implementation of core features—from AI-enhanced project recommendations and blockchain-secured transactions to comprehensive portfolio management—each developed through clearly defined sprints encompassing analysis, design, and deployment phases.

Through this structured approach, we demonstrate how Korpor leverages modern technology to reimagine investment management, offering a secure, transparent, and dynamic solution that is set to redefine digital financial engagement.

# CHAPTER 1

# Project Context

## 1.1 Introduction

The aim of this chapter is to present the general framework of the Korpor project, a solution dedicated to real estate investment. In this section, we'll discuss successively:

- The presentation of the host organization.
- The context and challenges of the real estate sector.
- Analysis of existing solutions and identification of their limitations.
- Definition of functional and non-functional needs of the mobile application and web back office.

## 1.2 Project Context

This work is part of the end-of-study project for the national diploma of Applied Bachelor's degree in Computer Science from the Higher Institute of Computer Science and Mathematics of Monastir (ISIMM) for the year 2024/2025. I had the opportunity to do my end-of-study internship at the company "KZ IT Services", under the supervision of Mr. Khalil Zouari.

## 1.3 Hosting Company

The purpose of this section is to present the company within which I developed my project, as shown in Figure 1.1.

### 1.3.1 Hosting Company

**KZ IT Service** is a dynamic software company dedicated to delivering innovative IT solutions tailored to modern business needs. We specialize in designing and developing robust,

**Figure 1.1:** Hosting Company "KZ IT Services"

scalable applications that drive efficiency and digital transformation. Our experienced team leverages cutting-edge technology to create customized software that exceeds client expectations. With a strong commitment to quality and continuous improvement, we build lasting partnerships based on trust and excellence. At "KZ IT Service", innovation is at the core of everything we do, empowering our clients to achieve sustainable growth and success.

## 1.4 Preliminary Study

This preliminary study provides a review of some existing investment and asset management platforms. Further, the next section identifies some key concepts that will lead to further understanding of the domain in question.

### 1.4.1 Existing Solutions Study

For understanding the present scenario and to clearly demarcate our goals, some renowned investment platform analyses offering similar features, including "Aseel" and "Stake", are performed [1, 2].

### 1.4.2 Available solutions and analysis

#### The Aseel Platform

**Aseel** is a portal through which users can invest in different real estate projects with ease. The interface allows the clients to surf various investment opportunities, view the details of the properties, and then make an informed decision. Aseel introduces transparency in the investment process by offering financial data, updates regarding projects, and returns that are estimated. This platform comes with an easy-to-use dashboard through which one tracks their investments and manages their assets without any hassle. The interface of the Aseel Platform is shown in Figure 1.2.

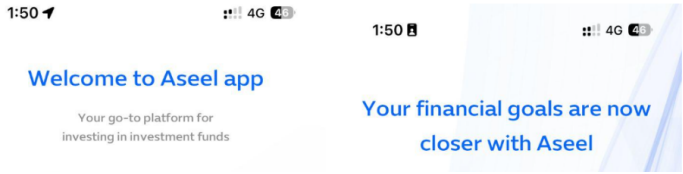
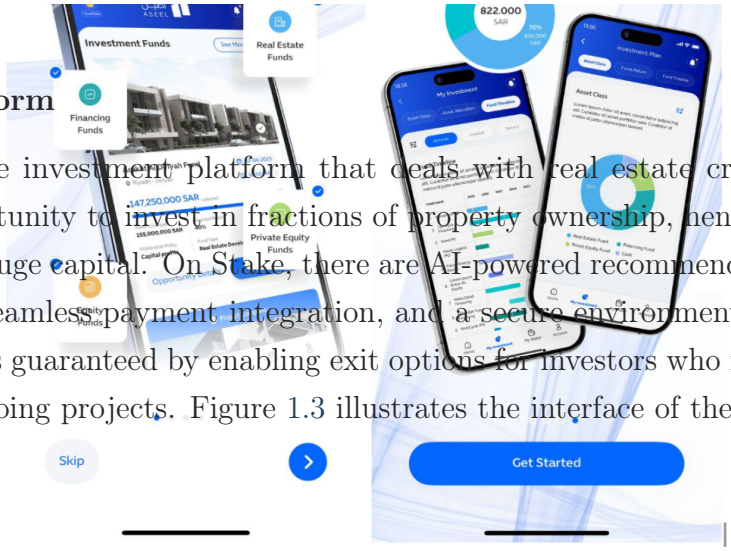



**Figure 1.2:** Interface of "The Aseel Platform"

### The Stake Platform

**Stake** is an online investment platform that deals with real estate crowdfunding. It provides the opportunity to invest in fractions of property ownership, hence diversifying a portfolio without huge capital. On Stake, there are AI-powered recommendations based on user preferences, seamless payment integration, and a secure environment for investment. Besides, liquidity is guaranteed by enabling exit options for investors who may want to sell their shares in ongoing projects. Figure 1.3 illustrates the interface of the Stake Platform.



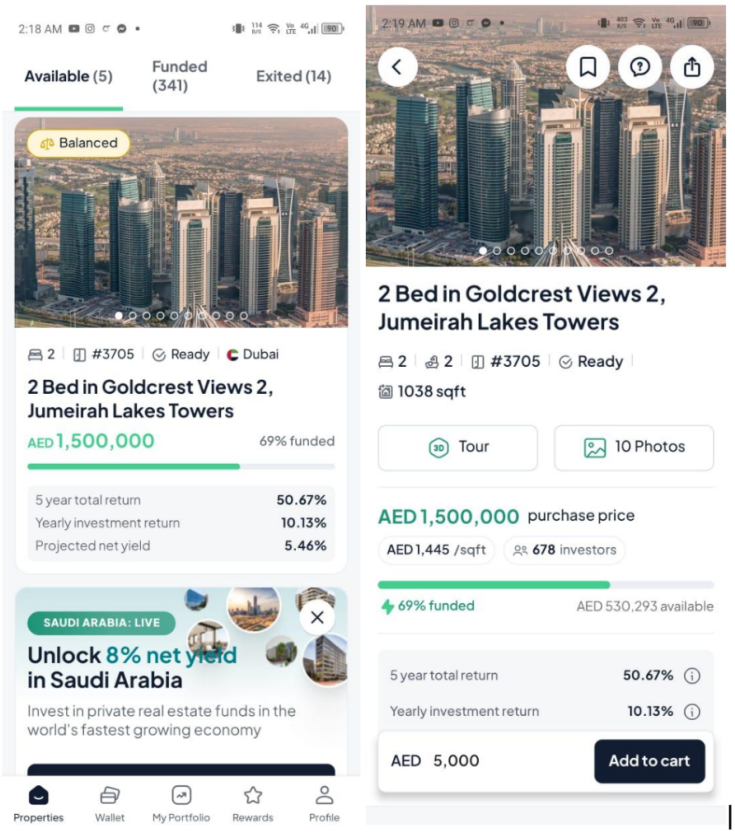



**Figure 1.3:** Interface of "The Stake Platform"

### 1.4.3 Comparative and Critical Analysis

We can summarize all that comes from our analysis based on a number of criteria used for the evaluation of these applications [3, 4].

- **Speed (C1)**: The platform should obtain value for the user as fast as possible and effectively, anticipating their proliferating expectations.
- **Costs (C2)**: With minimum software development costs, it is important to keep the pricing predictable and acceptable.
- **Quality (C3)**: Since the market expects quality, any kind of error might affect brand reputation. Improvement of the platform should be regular.
- **Reliability (C4)**: Since modern-day investment platforms need to make sure of minimum downtime and maximum availability of services, this factor is critical.
- **Security (C5)**: Such an investment platform enforces access rights, roles, and contribution rights through a powerful security system.
- **Performance (C6)**: Crucial features include AI-powered recommendations going through seamlessly, easy transaction tracking, and investment monitoring.

- **Stability (C7)**: The platform should have a proven track record, regular updates, and a large user base to ensure its longevity.
- **Resilience (C8)**: In order to prevent data loss and guarantee a smooth experience for investors, it must be able to restore lost functionalities should issues occur.
- **User Experience (C9)**: The interface should be intuitive and user-friendly, hence allowing investors to move with ease through it, thus making wiser decisions.

Table 1.1 presents the evaluation of the existing solutions based on these criteria:

**Table 1.1:** Evaluation Table

| Solution | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 |
|---|---|---|---|---|---|---|---|---|---|
| **Stake** | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ |
| **Aseel** | ✓ | ✓ | ✓ | × | ✓ | × | ✓ | × | ✓ |

### 1.4.4 Proposed Solution

Having studied the already working platforms for investments, we found strengths and weaknesses that could define what was required from the project. Our proposed solution will look at:

- Developing an efficient mobile application for investment management.
- Increasing the level of users' engagement with recommendations using the power of AI [5, 6, 7].
- Ensuring responsive and user-friendly interaction with the interface.
- Gaining the trust of investors by ensuring transparency and security in the investing platform.
- Enhancing the security of data and following all the financial regulations.

The **Korpor** platform will be offering the following features:

- A directory of investment opportunities with deep financial insights into those opportunities.
- AI-driven recommendations of investments as per users' preferences [8].
- Smooth funding and payout mechanisms.

- Real-time portfolio performance tracking on a single screen/dashboard.
- Forum for interactive discussions on strategy and market trends among its users.
- Referral and Rewards System: An engaging system for rewarding users via referral.

## 1.5 Development methodology

The completion of the project on its delivery date is the main problem of every software development team. One of the most common problems encountered in the production of software is insufficient technical specifications, poor time management in the face of the use of emerging technology, and sudden changes in needs. In order to avoid these critical issues, we follow an agile methodology for project management, using tools like Git [9] for version control and GitHub [10] for collaborative development.

### 1.5.1 SCRUM

**Scrum** is an agile development approach that is used to create software using incremental and iterative methods. Scrum is a quick, flexible, and efficient agile methodology that is intended to provide value to the client at every stage of the project's development [11]. Scrum is founded on empiricism and lean thinking, employing an iterative, incremental approach guided by the three pillars of transparency, inspection, and adaptation [12, 11]. Scrum's main goal is to meet customer needs by fostering an atmosphere of open communication, group accountability, and constant improvement, underpinned by the Scrum values of Commitment, Focus, Openness, Respect, and Courage [11]. The development process begins with a broad concept of what must be constructed, developing a list of features that the product owner desires, and arranging them according to priority (product backlog).

### 1.5.2 Agile Scrum roles and responsibilities

#### The Product Owner

Understands the customer and business requirements, then creates and manages the product backlog based on those requirements.

**Responsibilities:**

- Managing the scrum backlog
- Release management
- Stakeholder management

#### Developers

In Scrum, the term developer or team member refers to anyone who plays a role in the development and support of the product and can include researchers, architects, designers, programmers, etc.

**Responsibilities:**

- Delivering the work through the sprint
- To ensure transparency during the sprint, they meet daily at the daily scrum

#### Scrum Master

The role responsible for gluing everything together and ensuring that scrum is being done well. In practical terms, that means they help the product owner define value, the development team deliver the value, and the scrum team get better.

The Scrum Master focuses on:

- Transparency
- Empiricism
- Self-organization
- The Scrum events

### 1.5.3 The Scrum Events

The Scrum events are key elements of the Scrum Framework. They provide regular opportunities for enacting the Scrum pillars of Inspection, Adaptation and Transparency [11]. In addition, they help teams keep aligned with the Sprint and Product Goals, improve Developer productivity, and remove impediments and reduce the need to schedule too many additional meetings.

- **Sprint**: All work in Scrum is done in a series of short projects called Sprints. This enables rapid feedback loops.
- **Sprint Planning**: The Sprint starts with a planning session in which the Developers plan the work they intend to do in the Sprint. This plan creates a shared understanding and alignment among the team.
- **Daily Scrum**: The Developers meet daily to inspect their progress toward the Sprint Goal, discuss any challenges they've run into, and tweak their plan for the next day as needed.

- **Sprint Review**: At the end of the Sprint, the Scrum Team meets with stakeholders to show what they have accomplished and get feedback.
- **Sprint Retrospective**: Finally, the Scrum Team gets together to discuss how the Sprint went and if there are things they could do differently and improve in the next Sprint.

## 1.6 Conclusion

In conclusion of this chapter, it is clear that planning and methodology are essential pillars to ensure the success of the project. By fully understanding the project framework, including the host organization's expectations and the challenges ahead, the team is better prepared to meet the challenges ahead.

This chapter lays the solid foundation on which the entire project will be built, providing a valuable guide for the next steps. The next chapter will allow us to analyze and specify the needs developed in our project.

# CHAPTER 2

# Analysis and Specification of Needs

## 2.1 Introduction

In this chapter, we will present the analysis and specification of needs. We start by presenting the specification of the requirements, illustrating them using the diagram of the global use cases. Then we will present our project architecture and our working environment, and finally we will present our product backlog and releases planning, and we will close our chapter with a conclusion.

## 2.2 Requirements Specification

In this section, we will define the actors of our application and the functional and non-functional needs that our application aims to fulfill.

### 2.2.1 Identifying Actors

We define actors as a shorthand for the roles played by entities outside the system that interact directly with them [13, 14]. In our system, we identify four types of actors:

- **Super Admin**: Responsible for the global configuration of the platform, they have extended privileges to manage administrators, oversee security, and ensure compliance. They can also configure advanced features and control all system resources.
- **Admin**: In charge of the day-to-day management of the platform, they can add, modify, or delete listings, supervise agency and user profiles, and ensure smooth operations. They are also responsible for monitoring and assisting other actors.
- **Real Estate Agent**: Dedicated to creating and updating real estate listings, they manage property information, handle investor requests, and finalize transactions

related to sales or rentals. They can also coordinate property visits and propose tailored offers.

- **Investor**: A user who wishes to browse and finance real estate projects. They have access to all available offers, can make investments in a few simple steps, and monitor the evolution of their portfolio. They also benefit from personalized insights to optimize their investments.
- **System**: The entity that automatically manages all basic functionalities, such as authentication, notification generation, transaction validation, and adherence to security protocols. It ensures the coherence and reliability of the application at all times.

### 2.2.2 Functional Requirements

After several meetings with our client, the various functional requirements of our application are illustrated as follows:

**For the Super Admin (Korpor)**

- **Authenticate**: The super admin enters their credentials to access the advanced management console.
- **Log Out**: After viewing or updating global settings, they can securely log out.
- **Manage Admin Accounts**: Create, enable/disable, or modify admin profiles associated with different real estate companies.
- **Monitor Security & Compliance**: Oversee transactions, data integrity, and regulatory adherence using specialized reporting and audit tools.
- **Configure Platform Features**: Define key parameters (payment methods, AI/blockchain integrations, etc.) and roll out feature updates.
- **View Global Reports**: Generate and analyze consolidated metrics (financials, user activity, transactions) for overall performance insights.
- **Moderate Content**: Review and remove any inappropriate or erroneous property listings or user-generated data.

**For the Admin (Real Estate Company)**

- **Authenticate**: The admin logs in with valid credentials to manage daily operations.
- **Log Out**: They can end their session to maintain account security.

- **Manage Real Estate Listings**: Add, update, or delete property listings visible to investors.
- **Oversee Real Estate Agents**: Create and manage agent accounts, assign properties, and monitor performance and commissions.
- **Track Transactions & Commissions**: Review incoming payments, calculate commissions owed to agents, and track the history of completed deals.
- **Address Investor Inquiries**: Respond to questions or concerns from investors, ensuring a smooth user experience.
- **Access Agency Dashboard**: View comprehensive statistics on properties, sales, rentals, and market trends.

**For the Real Estate Agent**

- **Authenticate**: The agent logs in to manage assigned properties and interact with potential investors.
- **Log Out**: Securely exit the account after completing tasks.
- **Manage Assigned Properties**: Create new listings, update property details, set prices, and upload images.
- **Handle Investment Requests**: Review purchase or rental offers, negotiate terms, and initiate contract finalization.
- **Contribute to AI Estimates**: Provide or refine data to improve AI-driven pricing and market analysis.
- **Maintain Client Relationships**: Communicate with investors, schedule property visits, and follow up on inquiries.
- **View Commissions**: Track earnings based on successful sales or rentals.

**For the Investor (Mobile App User)**

- **Create an account & authenticate**: Register to gain access to the platform's core features.
- **Log Out**: End the session to protect personal and financial data.
- **Browse Listings & Invest**: Explore available properties, filter according to preferences, and commit to an investment in a few steps.

- **Track Portfolio**: Monitor owned assets, property status, and receive real-time updates on performance.
- **Make Payments**: Use integrated payment methods (credit cards, digital wallets, etc.) to complete transactions.
- **Access AI Recommendations**: View data-driven insights and return-on-investment estimates generated by the system.
- **Manage Withdrawals & Earnings**: Withdraw profits, monitor rental income, or exit investments under the right conditions.

**For the System**

- **Automate Authentication**: Validate credentials, manage sessions, and maintain user roles and permissions.
- **Generate Notifications**: Send real-time alerts (e.g., new listings, completed transactions, commission updates) to relevant users.
- **Ensure Compliance & Security**: Leverage blockchain for data integrity, verify payments, and detect anomalies or fraudulent activities.
- **Coordinate AI Insights**: Aggregate and analyze real estate data to produce market predictions and price recommendations.
- **Maintain Transaction Consistency**: Update dashboards, user balances, and property statuses automatically upon each operation.
- **Optimize Performance**: Monitor server load, scale resources, and ensure a smooth, responsive application experience.

### 2.2.3 Non-functional Requirements

In order to ensure the proper functioning of the decision-making system and to avoid any kind of anomaly, the implemented solution must meet a set of non-functional needs such as:

- **Maintainability**: The system must be designed for simplicity so that tasks, updates, and bug fixes can be executed with minimal complexity [15, 16].
- **Evolution**: Platform administration must remain attentive to user needs and feedback, continuously enhancing the services offered while preserving the application's utility and efficiency [17, 18].

- **Security**: Robust security measures are essential. The platform must enforce strong authentication protocols, access privileges, and comprehensive data encryption (both at rest and in transit) [19, 20]. The integration of blockchain technology further ensures the immutability and integrity of sensitive information [21, 22].
- **Efficiency**: The application must be effective in all circumstances, delivering prompt and reliable functionality regardless of external conditions [23, 24].
- **Performance**: The system must operate optimally across diverse environments. It should consistently provide a responsive and reliable experience, even under high transaction volumes or varying network conditions [25, 26].

## 2.3 Requirements Analysis

In this section, we'll outline the various features that our app should offer, using a general use case diagram [13].

### 2.3.1 General use case diagram

Below, we present the various actors of the application and the actions they are authorized to perform. The overall diagram is illustrated in Figure 2.1:

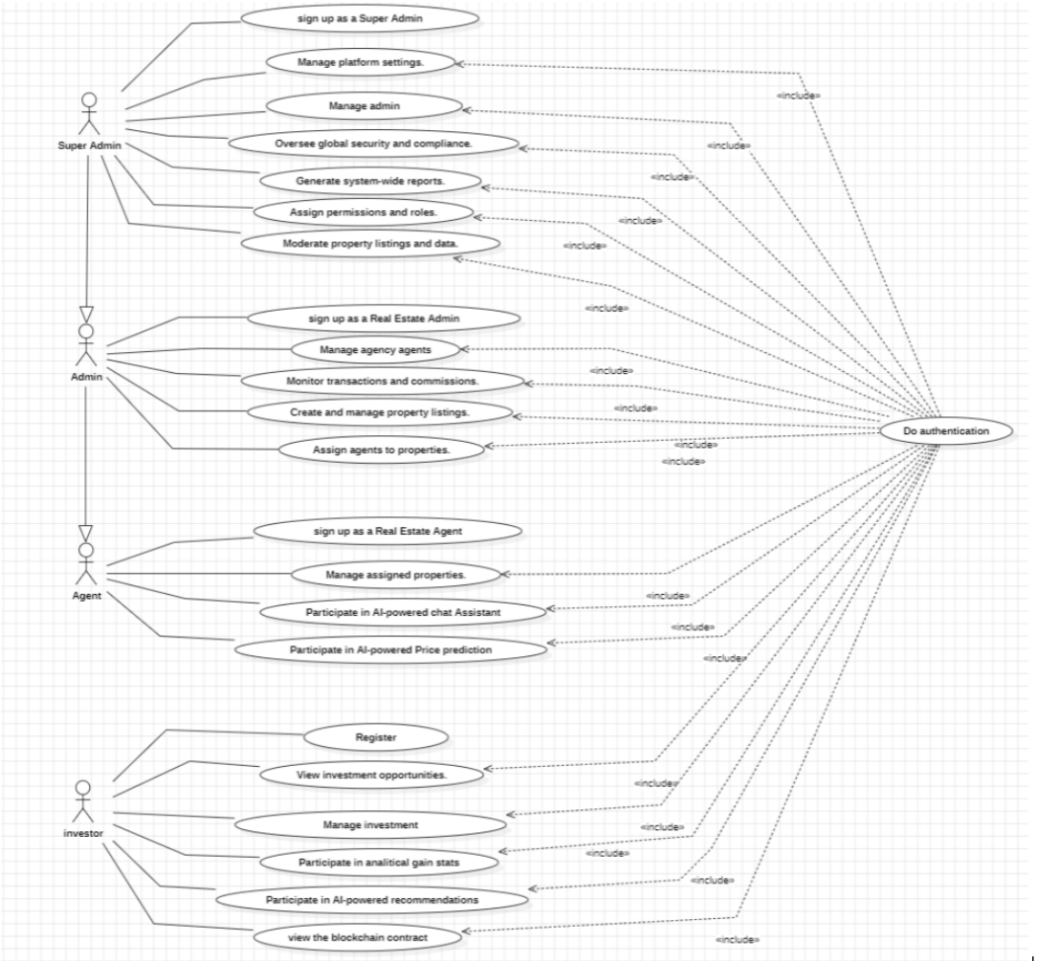



**Figure 2.1:** General use case diagram

## 2.4 Software architecture

Before starting the design and development of any computerized system, it is essential to prepare the architecture.

### 2.4.1 Physical architecture

The physical architecture of Korpor leverages modern, scalable technologies to deliver a seamless investment platform. The frontend is built with Expo, React, and TypeScript using Vite [27] for rapid development and Tanstack for robust state management and data visualization, while Storybook supports isolated UI component development [28]. The backend relies on Express.js [29] with user authentication managed by Clerk [19], containerization via Docker [25], and MySQL [30] for data storage hosted on Microsoft Azure [31]. Integrated AI modules provide personalized insights [32], and blockchain technology ensures transactional security and data immutability [21, 22]. This setup is further supported by GitHub [10] for version control, Postman [33] for API testing, and end-to-end testing tools like Maestro and Playwright [34], with architectural designs and documentation maintained using StarUML [35] and Overleaf.

**Figure 2.2:** Deployment diagram

### 2.4.2 Logical architecture

To better manage code organization and ensure maintainability, we've designed our application based on the MVC (Model-View-Controller) [36, 37] architectural pattern:

- **Model**: Represents application data and business logic.
- **View**: Presents data to users through interfaces.
- **Controller**: Processes incoming requests, performs operations using Models, and returns appropriate Views.

Architecture Components:

- **Model:** The MySQL database serves as the core data source, responsible for storing and managing application data, including user profiles, real estate listings, transactions, and investment records. The data layer interacts with the backend through ORM or query-based operations, ensuring efficient data retrieval and persistence.

- **Controller:** The Express.js backend acts as the intermediary between the database and the frontend, handling user requests, business logic, and data validation. It processes API calls from the frontend and interacts with services such as Clerk for authentication, AI modules for predictive analytics, and blockchain integration for secure transactions.
- **View:** The frontend is built with React, TypeScript, and TanStack tools, ensuring a responsive and interactive UI. The frontend communicates with the backend via API requests, displaying dynamic content and allowing seamless user interaction.

The logical architecture is illustrated in Figure 2.3.

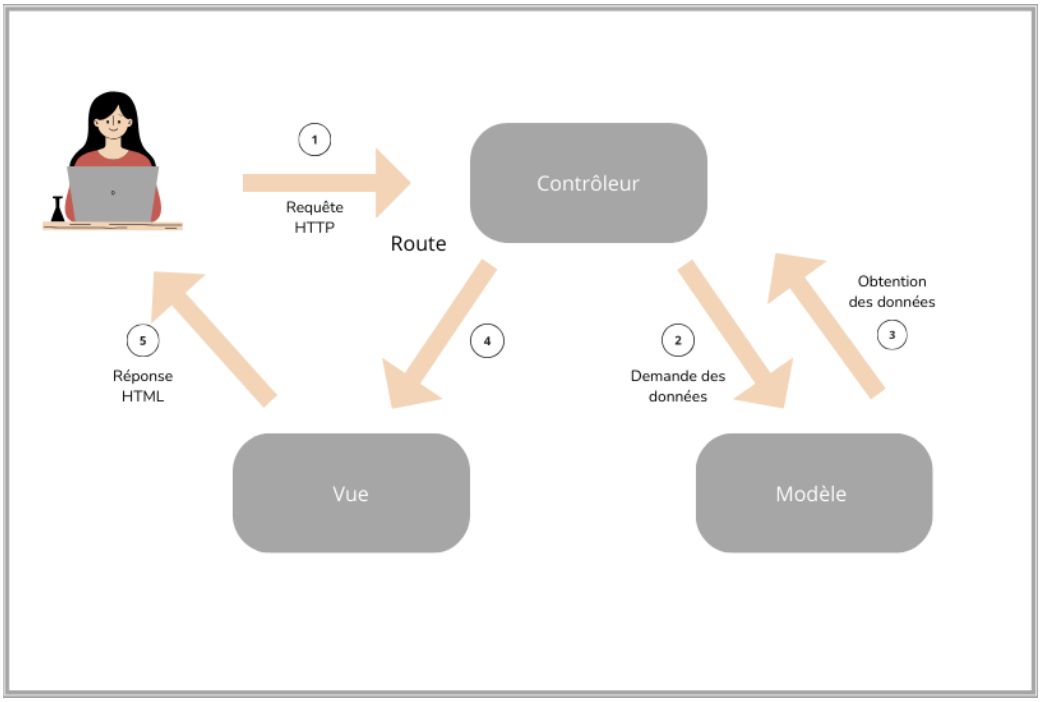



**Figure 2.3:** Logical architecture

Request Flow:

1. A user action (e.g., viewing property listings) triggers a request in the frontend.
2. The request is sent to the backend via an API call.
3. The Express.js controller processes the request, interacting with the database and other services.
4. Data is retrieved, processed, and returned as a response.
5. The frontend updates the UI dynamically based on the received data.

This structured approach ensures a scalable, secure, and high-performance system, optimizing Korpor's real estate investment and management operations [23].

## 2.5 Work Environment

In this part, we will talk about our work environment, focusing on different aspects: our material environment, the techniques we used in the realization of our project as well as the tools we used in our report, the product backlog and sprint planning, and finally, we will conclude this section [38, 39].

### 2.5.1 Physical environment

The work was carried out by a laptop computer that is equipped with these detailed features presented in Table 2.1 below:

| | |
|---|---|
| **Computer Name** | MSI |
| **Processor** | i5 10th gen |
| **Hard disk** | 512 Go SSD |
| **RAM** | 24.0 Go |
| **Operating system** | Windows 11 Pro |

**Table 2.1:** Physical environment

### 2.5.2 Used technologies

#### Expo

Expo is an open-source platform for making universal native apps for Android, iOS, and the web with JavaScript and React.

#### TypeScript

TypeScript (abbreviated as TS) is a free and open-source high-level programming language developed by Microsoft that adds static typing with optional type annotations to JavaScript [40]. It is designed for the development of large applications and transpiles to JavaScript.

#### Tanstack

High-quality open-source software for web developers [41]. Headless, type-safe, & powerful utilities for Web Applications, Routing, State Management, Data Visualization, Datagrids/Tables, and more.

#### Clerk

Clerk [19] is a complete suite of embeddable UIs, flexible APIs, and admin dashboards to authenticate and manage your users.

### Maestro

Maestro [42] is the simplest, most powerful, and most trusted end-to-end testing platform for mobile and web apps.

### Google cloud platform

Google cloud platform, or just GCP, is the cloud computing platform developed by Google. It has management, access and development of applications and services to individuals, companies, and governments through its global infrastructure.

### GitHub

GitHub [10] is a cloud-based service that helps developers store and manage their code, as well as track and control changes to their code.

### Express.js

Express.js [29] is a minimal and flexible Node.js [43] web application framework that provides a list of features for building web and mobile applications easily.

### Postman

Postman [33] is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

### Vite

Vite [27] is a modern build tool that provides a fast and optimized development experience for React 17 applications. It leverages native ES modules and offers a highly efficient development server with hot module replacement (HMR).

### React

React [44], sometimes referred to as a frontend JavaScript framework, is a JavaScript library created by Facebook.

### MySQL

MySQL [30] is an open-source relational database management system. It is based on structured query language (SQL), which is used to add, access and manage content in a database.

#### Docker

Docker is an open platform for developing, shipping, and running applications [25]. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.

#### Playright

Playwright [34] is an open-source testing and automation framework that can automate web browser interactions. To put it simply, you can write code that can open a browser [15].

#### Storybook

Storybook is a frontend workshop for building UI components and pages in isolation. It helps you develop and share hard-to-reach states and edge cases without needing to run your whole app [45].

#### StarUML

StarUML [35] is a sophisticated software modeler aimed to support agile and concise modeling. It provides eleven different types of diagrams and it accepts UML 2.x standards.

#### Node.js

Node.js [43] is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside a web browser, allowing developers to use JavaScript for server-side scripting.

### 2.5.3 Tools used for the report

#### Overleaf

Overleaf is a collaborative cloud-based LaTeX editor used to write, edit, and publish scientific papers.

#### Canva

Canva is a global company that empowers people to design anything and publish anywhere. Learn about its mission, values, commitments, awards, product, and careers.

## 2.6 Product backlog

The backlog was created before the sprints to plan the milestones and determine the content of each sprint based on project needs [**ProductBacklogGuide2024**, 46]. It includes the following fields:

- **Code:** The unique identifier of the task.
- **Theme:** The subject of a user story.
- **User Story:** A short description of the functionality requested by the client.
- **Priority:** A value indicating the importance of the functionality [47, 48].
  - **Must:** The feature is essential and must be implemented.
  - **Should:** The feature should be implemented if possible.
  - **Could:** The feature is optional and may be deprioritized.
  - **Won't:** The feature is not a priority at this time.

Table 2.2 shows the product backlog for our Korpor project:

**Table 2.2:** Korpor Product Backlog

| Code | Theme | User story | Priority |
|---|---|---|---|
| **Authentication & User Management** | | | |
| PB001 | Authentication | As a user, I want to create an account and authenticate securely | Must |
| PB002 | User Management | As a user, I want to manage my profile information | Must |
| PB003 | Authentication | As a user, I want to securely log out of my account | Must |
| PB004 | Admin Management | As a Super Admin, I want to manage admin accounts for different real estate companies | Should |
| **Super Admin Features** | | | |
| PB005 | Security | As a Super Admin, I want to monitor security and compliance across the platform | Could |
| PB006 | Configuration | As a Super Admin, I want to configure platform-wide features and settings | Could |
| PB007 | Analytics | As a Super Admin, I want to generate and analyze global performance reports | Could |

Continued on next page

| | | | |
|---|---|---|---|
| PB008 | Moderation | As a Super Admin, I want to moderate content across the platform | Won't |
| PB009 | AI Integration | As a Super Admin, I want to chat with an AI assistant that can securely access database information | Could |
| **Admin Features** | | | |
| PB010 | Listing Management | As an Admin, I want to manage real estate listings in my company | Must |
| PB011 | Agent Management | As an Admin, I want to oversee real estate agents and their permissions | Should |
| PB012 | Transaction Management | As an Admin, I want to track transactions and calculate agent commissions | Should |
| PB013 | Customer Service | As an Admin, I want to address investor inquiries and issues | Could |
| PB014 | Analytics | As an Admin, I want to access a comprehensive agency dashboard | Could |
| PB015 | AI Integration | As an Admin, I want to input property details and receive AI-powered valuation | Should |
| **Real Estate Agent Features** | | | |
| PB016 | Listing Management | As an Agent, I want to create and manage property listings | Must |
| PB017 | Investment Management | As an Agent, I want to handle investment and purchase requests | Could |
| PB018 | Data Management | As an Agent, I want to contribute data for AI-driven estimates | Could |
| PB019 | Customer Relations | As an Agent, I want to maintain client relationships and communications | Could |
| PB020 | Finance | As an Agent, I want to view my commissions on sales and rentals | Should |
| **Investor Features** | | | |
| PB021 | Property Discovery | As an Investor, I want to browse available property listings | Must |
| PB022 | Search Functionality | As an Investor, I want to filter properties based on my preferences | Could |
| PB023 | Investment Process | As an Investor, I want to invest in properties through a simple process | Could |

Continued on next page

| | | | |
|---|---|---|---|
| PB024 | Portfolio Management | As an Investor, I want to track my investment portfolio in real-time | Must |
| PB025 | Payment Processing | As an Investor, I want to make secure payments through various methods | Should |
| PB026 | AI Recommendations | As an Investor, I want to receive personalized property recommendations | Must |
| PB027 | AI Assistance | As an Investor, I want to consult an AI assistant for real estate legal questions | Could |
| PB028 | Financial Prediction | As an Investor, I want to see predictions of potential earnings | Could |
| PB029 | Finance Management | As an Investor, I want to manage my earnings and withdrawals | Could |
| **AI & Machine Learning Features** | | | |
| PB030 | AI System | As the System, I want to analyze user interactions for personalized recommendations | Must |
| PB031 | AI Prediction | As the System, I want to predict property valuations and rental prices | Should |
| PB032 | AI Forecasting | As the System, I want to forecast potential investment returns | Should |
| PB033 | NLP Integration | As the System, I want to provide real estate legal information via NLP | Could |
| PB034 | Security | As the System, I want to secure AI database access for authorized queries | Could |
| **Blockchain & Smart Contract Features** | | | |
| PB035 | Blockchain | As the System, I want to create and manage virtual contracts for transactions [21] | Must |
| PB036 | Blockchain | As an Investor, I want my property investments to be secured via blockchain [22] | Must |
| PB037 | Blockchain Management | As an Admin, I want to verify and validate blockchain transactions | Should |
| PB038 | Data Integrity | As the System, I want to store transaction records immutably on blockchain | Must |
| PB039 | System Monitoring | As a Super Admin, I want to monitor blockchain health and performance | Should |
| **System & Security Features** | | | |

Continued on next page

| | | | |
|---|---|---|---|
| PB040 | Authentication | As the System, I want to automate authentication and session management | Should |
| PB041 | Notifications | As the System, I want to generate real-time notifications for relevant users | Could |
| PB042 | Data Consistency | As the System, I want to maintain transaction consistency across the platform | Should |
| PB043 | Security | As the System, I want to ensure secure communication between AI and database | Should |
| **DevOps & Infrastructure** | | | |
| PB044 | CI/CD | As a Developer, I want CI/CD pipelines to build project images on GitHub [26] | Must |
| PB045 | Deployment | As a Developer, I want to automate image deployment to GCP registry | Must |
| PB046 | Deployment | As a Developer, I want to auto-deploy backend services and database | Must |
| PB047 | Containerization | As a Developer, I want to containerize application components with Docker | Must |
| PB048 | Web Deployment | As a Developer, I want to automatically deploy the web app frontend | Should |
| PB049 | Mobile Deployment | As a Developer, I want to automate mobile app deployments to Play Store | Should |
| PB050 | Versioning | As a Developer, I want to implement versioning for mobile app releases | Should |
| PB051 | Monitoring | As an Admin, I want to monitor system health and performance | Could |
| PB052 | Configuration | As a Super Admin, I want to manage environment configurations | Could |
| **Mobile App Specific Features** | | | |
| PB053 | Mobile UI/UX | As an Investor, I want a responsive, intuitive mobile interface [49] | Should |
| PB054 | Notifications | As an Investor, I want to receive push notifications about my investments | Should |
| PB055 | Offline Access | As an Investor, I want offline access to my basic portfolio information | Should |

## 2.7 Sprint planning

In order to complete the project within the deadlines set by the internship agreement, planning is an important step in the process [50, 51]. It was therefore necessary to define the essential steps and estimate the time to be devoted to the completion of the various tasks. To do this, we made a GANTT chart.

In our project management, we opted for the proportional distribution method in order to estimate the costs [52, 53]. Figure 2.4 shows the Gantt chart that describes the progress of our project:

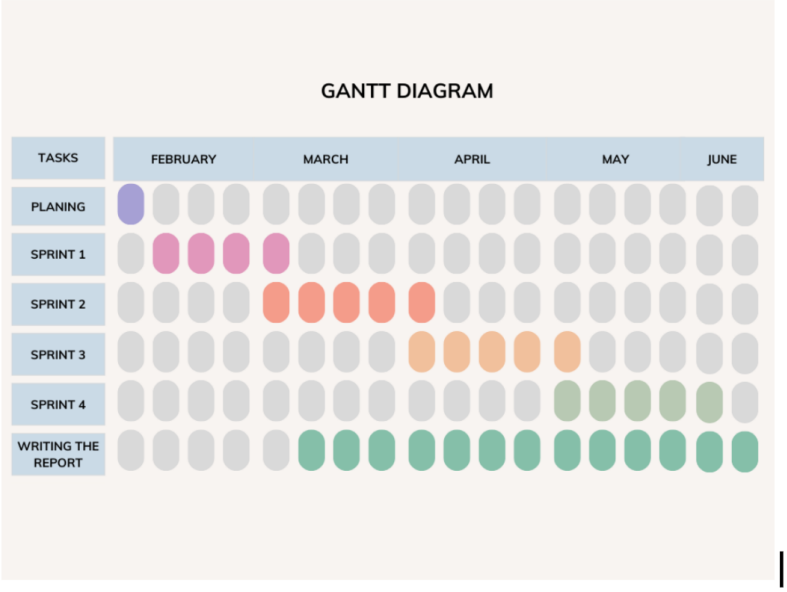



**Figure 2.4:** GANTT chart with sprint planning

## 2.8 Conclusion

Our Sprint 0 marked the exciting start of our KORPOR project [54, 50]. We defined global and specific objectives, developed a solid architecture, and configured an optimal working environment. With a clear vision of the initial product backlog and preliminary planning for upcoming sprints, we are ready to achieve our vision and achieve our goals successfully [46, 51].

# CHAPTER 3

# Implementation

## 3.1 Introduction

This chapter details the implementation phase of our project, which follows an agile methodology with four sprints. Each sprint focuses on delivering specific features and functionality according to the project backlog. The implementation utilizes a modern tech stack consisting of React [44] with Vite [27] for the frontend, Node.js [43] with Express.js [29] for the backend, MySQL [30] for database management, and Tailwind CSS [55] for styling.

## 3.2 Sprint 1: Authentication and User Management

### 3.2.1 Overview

The first sprint focuses on establishing the core authentication system and user management functionality. This foundation is critical for all subsequent features as it defines user roles and access controls.

### 3.2.2 User Types

The system supports three distinct user types, each with different permissions and capabilities:

- **Super Admin**: Has complete access to all system features and can manage admins and agents.
- **Admin**: Can manage agents and has access to administrative features within their assigned scope.
- **Agent**: Has limited access to the system based on their assigned responsibilities.

### 3.2.3 Authentication System

#### Sign-up Process

The sign-up process is illustrated in Figure **??** below. The diagram shows the authentication flow for new users registering in the system.

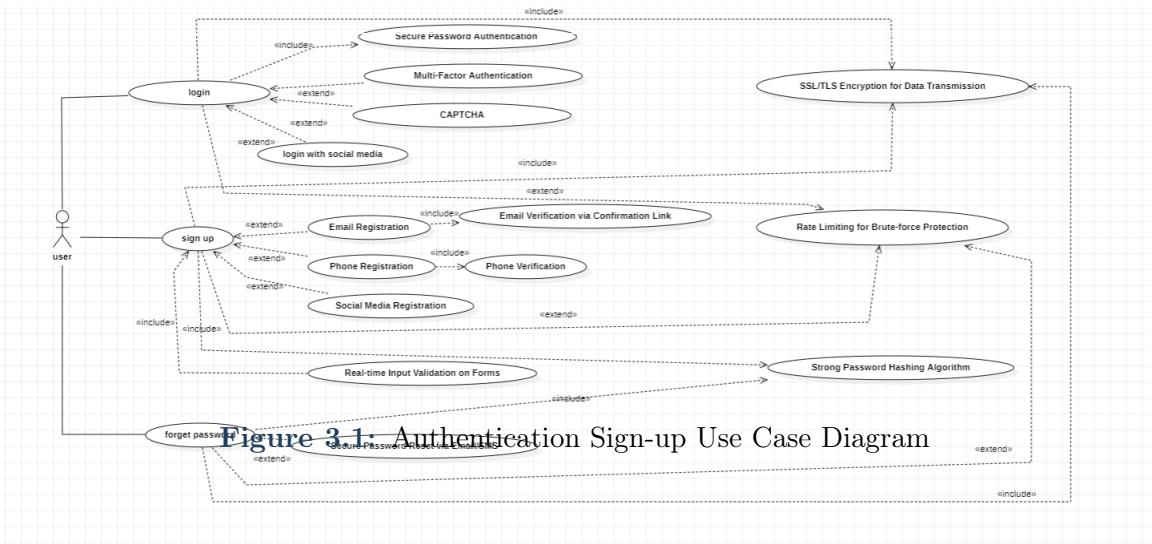



**Figure 3.1:** Authentication Sign-up Use Case Diagram

The sign-up process includes user registration, role assignment, and account verification steps. During registration, users are categorized into one of the three user types: Super Admin, Admin, or Agent, with each type having different permissions and access levels within the system.

### 3.2.4 Sprint 1 Deliverables

The key deliverables for Sprint 1 include:

**Web Backoffice Authentication**

- Admin dashboard login and authentication system
- Role-based access control for backoffice users (Super Admin, Admin, Agent)
- User management interface for creating and managing user accounts
- Permission management system for different user roles
- Security logs and audit trails for backoffice activities
- Session management and secure token handling

**Blockchain Foundation**

- Blockchain architecture setup and configuration
- Smart contract development for property transactions
- Implementation of test transactions in blockchain test environment
- Integration of blockchain APIs with backend services
- Transaction validation and verification mechanisms
- Wallet creation and key management system

**AI Property Evaluation System**

- AI model development for rent price prediction using TensorFlow.js [7]
- Property sale value estimation algorithm
- Data collection and preprocessing pipeline for real estate information
- Training environment setup for prediction models
- Integration of prediction APIs with property management system
- Evaluation dashboard for property value assessments

**DevOps Infrastructure**

- CI/CD pipeline configuration on GitHub [10] for backend services
- GCP Cloud Run integration for containerized deployment
- Automated testing and deployment workflow
- Initial Docker setup for application components:
  - Frontend container configuration with React [44] and Vite [27]
  - Mobile application build environment
  - Web application container
  - Backend API services container using Express.js [29] and Node.js [43]
  - Nginx configuration for routing and load balancing
  - AI services containerization
- Monitoring and logging infrastructure setup

**Mobile Application**

- Mobile authentication screens and functionality
- Biometric authentication integration
- Investment properties listing interface
- Property search and filtering functionality
- User profile management on mobile
- Push notification system for investment opportunities

# CHAPTER 4

# Artificial Intelligence Features

## 4.1 introduction

# CHAPTER 5

# Blockchain & Backoffice Features

## 5.1 Introduction

# CHAPTER 6

# DevOps & Mobile App Features

## 6.1 Introduction

# CHAPTER 7

# Conclusion & Future Work

*The best way to predict the future is to create it.*

— Abraham Lincoln

## 7.1 Summary of Achievements

## 7.2 Challenges Encountered

## 7.3 Future Improvements

## 7.4 Lessons Learned

# CHAPTER 8

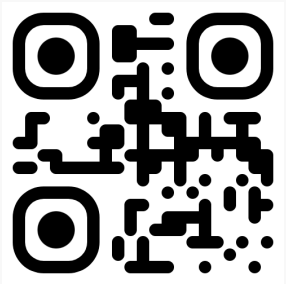
# Startup Vision: Launching Korpor

## 8.1 Introduction

This chapter outlines the prospective journey of transforming the Korpor project into an independent startup venture. It details the vision, the strategic arrangement with the sponsoring company, and includes key legal documents underpinning this transition.

**The Entrepreneurial Leap**

The culmination of this project is not merely the delivery of a functional platform, but the foundation for **my future tech company, Korpor**. The sponsoring company has graciously agreed to provide the core application codebase, enabling the launch of this independent startup. Korpor will be dedicated to the continuous development and enhancement of this real estate investment platform. This section will elaborate on the operational model for Korpor, focusing on its evolution into a Software-as-a-Service (SaaS) provider for the wider real estate industry, its market strategy, and its potential impact.



*Scan to visit the Korpor page.*

## 8.2 Founding Documents

**Legal Framework**

This section contains the formal documentation establishing the legal groundwork for the Korpor startup. The primary document included signifies the official handover of the application codebase developed during the initial project phase.
This agreement, originating from the project lead, confirms the transfer of the core technology to the new venture, Korpor, paving the way for its independent operation and future development as a SaaS provider.

## 8.3 Path Forward

**Strategic Roadmap**

With the application codebase secured and the legal groundwork laid, the strategic path forward for **Korpor as my tech company** involves several key milestones:

- **Secure Funding:** Actively pursue seed funding to support initial operations, team expansion, and marketing efforts.
- **Build the Core Team:** Recruit talented individuals for key roles in development, marketing, sales, and operations.
- **Develop SaaS Model:** Refine the platform architecture and features to support a scalable, multi-tenant Software-as-a-Service offering.
- **Iterate and Enhance:** Continuously improve the platform based on user feedback and market analysis, focusing on delivering exceptional value to client companies.
- **Market Penetration:** Implement a targeted marketing strategy to acquire initial real estate company clients and build brand presence.

The ultimate goal is to establish Korpor as a trusted and innovative SaaS provider, empowering other real estate businesses with advanced technology solutions.

**KHALIL ZOUARI**

+216 95 848 985

Khalilzouari@icloud.com

6 avenue Habib Bourguiba,
4116 Djerba Midoun

**1st May 2025**

Subject: Formal Transfer of the Korpor Application Codebase

To whom it may concern,

This letter serves to formally document the transfer of the software application known as 'Korpor', which was developed as part of the Final Year Project completed at KZ IT Services for Ahmed jaziri.

Effective from the date stated herein, the complete source code, associated technical documentation, build artifacts, and deployment assets related to the Korpor application are hereby transferred. This transfer is intended to grant the necessary rights to utilize, maintain, modify, distribute, and further develop the application for commercial purposes under the new Korpor startup venture.

The purpose of this transfer is to provide the foundational technology asset for the Korpor startup, enabling its launch and future operations. This will allow the startup to pursue its mission of developing and offering the Korpor platform as a Software-as-a-Service (SaaS) solution for the real estate industry. This transfer is made in accordance with the prior understanding and agreements established during the project phase.

Khalil Zouari
chef de projet

company stamp

KHALIL ZOUARI
IT SERVICES
+216 95 848 985
khalilzouari@icloud.com
MF:1883516/R/A/M/000

# APPENDIX A

# Technical Documentation

# APPENDIX B

# User Manual

# APPENDIX C

# Project Timeline

# Bibliography


[1] Jeremy Moser. *How to Conduct a Competitive Analysis of Software Solutions*. G2 Learn. Jan. 2024. URL: https://learn.g2.com/software-competitive-analysis (visited on 04/14/2025).

[2] Team Asana. *How to create a competitive analysis (with examples)*. Asana. Feb. 2024. URL: https://asana.com/resources/competitive-analysis-example (visited on 04/18/2025).

[3] Kai Tomboc. *UX competitive analysis*. Lyssna Blog. Apr. 2024. URL: https://www.lyssna.com/blog/ux-competitive-analysis/ (visited on 05/02/2025).

[4] Ovidijus Jurevicius. *The Complete Guide to Competitive Analysis*. Strategic Management Insight. Apr. 2024. URL: https://strategicmanagementinsight.com/tools/competitive-analysis/ (visited on 05/24/2025).

[5] Mobile Reality Team. *How AI in Real Estate is Transforming the Future of Property and Investment*. The Mobile Reality. Mar. 2024. URL: https://themobilereality.com/blog/proptech/ai-in-real-estate (visited on 03/19/2025).

[6] HouseCanary Team. *5 AI Tools for Real Estate Investors*. HouseCanary Blog. Dec. 2024. URL: https://www.housecanary.com/blog/5-ai-tools-for-real-estate-investors (visited on 02/10/2025).

[7] Daniel Smilkov et al. "Tensorflow.js : Machine learning for the web and beyond". In: *Proceedings of Machine Learning and Systems* 1 (2019), pp. 309–321.

[8] RECAP. *AI, fintech, and automation on the real estate industry*. LinkedIn. Feb. 2024. URL: https://www.linkedin.com/pulse/ai-fintech-automation-real-estate-industry-recapeg-wup1f (visited on 03/15/2025).

[9] *Git*. Software Freedom Conservancy. URL: https://git-scm.com/ (visited on 03/20/2025).

[10] *GitHub*. GitHub, Inc. URL: https://github.com/ (visited on 03/20/2025).

[11] Ken Schwaber and Jeff Sutherland. *The Scrum Guide*. The official Scrum Guide, November 2020 version. Scrum.org. 2020. URL: https://scrumguides.org/scrum-guide.html (visited on 05/15/2025).

[12] Atlassian Team. *Three Pillars of Scrum: Understanding Scrum's Core Principles*. Atlassian Agile Coach. 2023. URL: https://www.atlassian.com/agile/project-management/3-pillars-scrum (visited on 04/20/2025).

[13] Alistair Cockburn. *Writing Effective Use Cases*. Boston, MA: Addison-Wesley Professional, 2002. ISBN: 978-0201702255.

[14] Alistair Cockburn. *Writing Effective Use Cases*. Addison-Wesley Professional, 2000. ISBN: 978-0201702255.

[15] Jennifer Davis and Ryn Daniels. *DevOps Foundations: Modern Software Development Practices*. O'Reilly Media, 2023. ISBN: 978-1492097136.

[16] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. 2nd ed. Addison-Wesley Professional, 2018. ISBN: 978-0134757599.

[17] Mary Poppendieck and Tom Poppendieck. "Lean Software Development: An Agile Toolkit". In: *Agile Software Development Series* (2012).

[18] Henrik Kniberg. *Lean from the Trenches: Managing Large-Scale Projects with Kanban*. Pragmatic Bookshelf, 2013. ISBN: 978-1934356852.

[19] *Clerk Authentication Documentation*. Clerk, Inc. 2023. URL: https://clerk.com/docs (visited on 05/27/2025).

[20] OWASP Foundation. *OWASP Top Ten Security Principles*. Open Web Application Security Project. 2021. URL: https://owasp.org/www-project-top-ten/ (visited on 05/30/2025).

[21] Hui Wang, Yuming Chen, and Kaili Zhu. "Blockchain Applications in Real Estate: Current State and Future Prospects". In: *International Journal of Strategic Property Management* 27.1 (2023), pp. 32–49. DOI: 10.3846/ijspm.2023.18372.

[22] McKinsey & Company. *Blockchain in Real Estate: Transforming Property Transactions*. McKinsey & Company. 2023. URL: https://www.mckinsey.com/industries/real-estate/our-insights/blockchain-in-real-estate-transforming-property-transactions (visited on 04/22/2025).

[23] Gene Kim et al. "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations". In: *IT Revolution Press* (2018).

[24] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. 4th ed. Addison-Wesley Professional, 2021. ISBN: 978-0136886099.

[25] Docker Documentation Team. *Docker Architecture and Container Design*. Docker, Inc. 2023. URL: https://docs.docker.com/get-started/overview/ (visited on 05/27/2025).

[26] Nicole Forsgren and Jez Humble. *DevOps Metrics That Matter: Measuring Success in 2023*. Google Cloud. 2023. URL: https://cloud.google.com/blog/products/devops-sre/the-2023-accelerate-state-of-devops-report-now-out (visited on 04/22/2025).

[27] *Vite*. Evan You and Vite contributors. URL: https://vitejs.dev/guide/ (visited on 05/23/2025).

[28] *Storybook Documentation*. Storybook Contributors. 2022. URL: https://storybook.js.org/docs/ (visited on 05/30/2025).

[29] *Express - Node.js web application framework*. Node.js Foundation. URL: https://expressjs.com/ (visited on 05/23/2025).

[30] *MySQL Documentation*. Oracle Corporation. URL: https://dev.mysql.com/doc/ (visited on 05/23/2025).

[31] Microsoft Azure Team. *Azure Cloud Platform Services*. Microsoft Corporation. 2024. URL: https://azure.microsoft.com/en-us/services/ (visited on 05/27/2025).

[32] Michael R. Johnson and Wei Zhang. "Advanced AI Techniques in Property Valuation: Challenges and Opportunities". In: *Journal of Real Estate Technology* 15.2 (2024), pp. 78–95. DOI: 10.1080/14735789.2024.2175633.

[33] *Postman*. Postman, Inc. URL: https://www.postman.com/ (visited on 03/20/2025).

[34] *Playwright End-to-End Testing Documentation*. Microsoft Corporation. 2023. URL: https://playwright.dev/docs/intro (visited on 05/30/2025).

[35] *StarUML*. MKLabs Co., Ltd. URL: https://staruml.io/ (visited on 04/18/2025).

[36] Andri Sunardi and Suharjito. "MVC Architecture : A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based". In: *Procedia Computer Science* 157 (2019), pp. 134–141. DOI: 10.1016/j.procs.2019.08.150. URL: https://doi.org/10.1016/j.procs.2019.08.150.

[37] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994. ISBN: 978-0201633610.

[38] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change*. 2nd ed. Addison-Wesley Professional, 2004. ISBN: 978-0321278654.

[39] Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2017. ISBN: 978-0134494166.

[40] *TypeScript Documentation*. Microsoft Corporation. 2023. URL: https://www.typescriptlang.org/docs/ (visited on 05/30/2025).

[41] *TanStack - High-quality open-source software for web development*. TanStack. 2023. URL: https://tanstack.com/ (visited on 05/30/2025).

[42] *Maestro Documentation*. Maestro Labs, Inc. 2022. URL: https://maestro.mobile.dev/ (visited on 05/30/2025).

[43] *Node.js*. OpenJS Foundation. URL: https://nodejs.org/en (visited on 05/18/2025).

[44] *React*. Meta Platforms, Inc. URL: https://react.dev/ (visited on 04/16/2025).

[45] Ethan Marcotte. *Responsive Web Design: Patterns and Principles for the Modern Web*. 3rd ed. A Book Apart, 2023. ISBN: 978-1937557522.

[46] Jeffrey Schwarz. *The Scrum Product Backlog: A Guide for Product Owners*. Scrum.org. 2019. URL: https://www.scrum.org/resources/blog/scrum-product-backlog-guide-product-owners (visited on 05/30/2025).

[47] Dai Clegg and Richard Barker. *The MoSCoW Method: A Prioritization Technique for Project Management*. ProductPlan. 2021. URL: https://www.productplan.com/glossary/moscow-prioritization/ (visited on 05/27/2025).

[48] Dai Clegg and Richard Barker. *Case Method Fast-Track: A RAD Approach*. Addison-Wesley, 2004. ISBN: 978-0201624328.

[49] Jakob Nielsen and Kara Pernice. "Responsive Design and User Experience: Impact on Mobile Interfaces". In: *User Experience Magazine* 19.3 (2019).

[50] Jeff Sutherland and J.J. Sutherland. *Scrum: The Art of Doing Twice the Work in Half the Time*. Currency, 2020. ISBN: 978-0385346474.

[51] Kenneth S. Rubin. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional, 2012. ISBN: 978-0137043293.

[52] Mike Cohn. *Agile Estimating and Planning*. Prentice Hall, 2005. ISBN: 978-0131479418.

[53] James Grenning. "Planning Poker or How to avoid analysis paralysis while release planning". In: *Renaissance Software Consulting* 3 (2002). URL: https://www.renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf.

[54] Scaled Agile, Inc. *SAFe for Lean Enterprises 6.0*. Scaled Agile, Inc. 2024. URL: https://www.scaledagileframework.com/ (visited on 04/22/2025).

[55] *Tailwind CSS*. Tailwind Labs Inc. URL: https://tailwindcss.com/ (visited on 04/16/2025).