

CHAPTER 2. SOFTWARE REQUIREMENTS

2.1 Tools and technology

2.1.1. Reactjs

ReactJS, or simply React, is an open-source JavaScript library for building user interfaces. It was developed by Facebook and has gained immense popularity in recent years due to its simplicity, performance, and flexibility. React allows developers to build reusable UI components and manage state effectively, making it easy to develop complex applications with a scalable and maintainable codebase. React is based on a component-based architecture, where each UI element is treated as a self-contained component. React components can be easily reused and composed to build complex user interfaces. The library also offers a virtual DOM, which helps to optimize the performance of web applications by minimizing the number of updates to the actual DOM. React uses a declarative approach to programming, where developers define what they want the UI to look like and React handles the updates and rendering of the UI automatically. This makes React easy to learn and use for developers of all levels of expertise. React has a large and active community that provides numerous resources, tools, and libraries to support developers in building web applications. This has made React a top choice for web development and has been adopted by many popular websites and web applications.

2.1.2. Nextjs

Next.js is a framework built on top of ReactJS that aims to provide a simpler and more streamlined approach to building server-rendered React applications. It offers features such as automatic code splitting, server-side rendering, and static site generation, making it easier to build performant, scalable, and SEO-orderly web applications. One of the main differences between Next.js and ReactJS is that Next.js provides server-side rendering out of the box, which means that web pages can be pre-rendered on the server and delivered to the client as fully-formed HTML pages. This allows for faster initial page loads, improved SEO,

and better performance on slow or unreliable networks. ReactJS, on the other hand, is a client-side library that relies on the browser to render web pages. Another key difference is that Next.js provides built-in support for routing and code splitting, which simplifies the development process and improves performance. ReactJS, on the other hand, requires developers to implement their own routing and code splitting solutions.

2.1.3. Nodejs

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine that allows developers to build server-side applications using JavaScript. It provides a platform for building scalable, high-performance applications using event-driven, non-blocking I/O techniques. Node.js is often used for building APIs, microservices, and other server-side applications. One of the key benefits of Node.js is its performance. Because it uses an event-driven, non-blocking I/O model, it is able to handle large numbers of concurrent connections with high throughput and low latency. It also allows for easy scalability, with the ability to add more instances to handle increasing traffic. Another benefit of Node.js is its flexibility. It can be used for building a wide range of applications, from simple command-line tools to complex, distributed systems. In summary, Node.js provides developers with a powerful and flexible platform for building server-side applications using JavaScript. Its event-driven, non-blocking I/O model and performance capabilities make it a popular choice for building scalable, high-performance applications.

2.1.4. Mongoddb

MongoDB is a popular NoSQL document-oriented database that provides a flexible, scalable, and high-performance solution for storing and managing data. Unlike traditional relational databases, MongoDB uses a flexible document data model that allows developers to store data in a more natural and expressive way. One of the key benefits of MongoDB is its scalability. It is designed to scale horizontally across multiple servers, allowing for the efficient handling of large amounts of data. It also provides built-in support for sharding,

which enables data to be distributed across multiple servers. Another benefit of MongoDB is its ease of use. Its document data model makes it easy to work with data in a more natural and intuitive way, and it provides powerful querying and indexing capabilities that allow for efficient data retrieval. In addition, MongoDB is open-source and has a large and active community of developers, making it easy to find support and resources. In summary, MongoDB is a powerful, scalable, and easy-to-use NoSQL database that provides a flexible and high-performance solution for storing and managing data. Regenerate response

2.1.5. Bootstrap

Bootstrap is a popular front-end web development framework that provides developers with a set of pre-built HTML, CSS, and JavaScript components for creating responsive, mobile-first websites and web applications. It was originally developed by Twitter and is now maintained by a large community of developers. One of the key benefits of Bootstrap is its ease of use. It provides a wide range of pre-built components, such as navigation menus, forms, buttons, and typography, that can be easily customized and incorporated into web projects. It also provides a responsive grid system that makes it easy to create layouts that adapt to different screen sizes. Another benefit of Bootstrap is its cross-browser compatibility. It is designed to work with all major web browsers and provides consistent styling and functionality across different devices and platforms. In addition, Bootstrap is open-source and has a large and active community of developers, making it easy to find support and resources. In summary, Bootstrap is a powerful, easy-to-use front-end web development framework that provides a range of pre-built components and tools for creating responsive, mobile-first websites and web applications.

2.1.6. Visual Studio

Visual Studio is an Integrated Development Environment (IDE) by Microsoft. This IDE enable developers to develop games, web application, mobile applications, and Windows application in many popular programming languages and runtimes, such as C++, C#, JavaScript, TypeScript, NodeJS, Python, dotNet. Not only application, but also developing database solutions for SQL Server and Azure SQL Database. Another interesting feature from Visual

Studio is Collaboration feature, where developers can manage version control by using Git or GitHub and work together in real-time by using LiveShare feature. Visual Studio is available in Community Version (free), Professional (paid), Enterprise (paid). In this study, the Visual Studio Community will be used.

2.1.7. Postman

Postman is a popular tool used by developers for testing and debugging APIs. It provides a user-orderly interface that allows developers to easily send requests to APIs, view responses, and test functionality. Postman supports a wide range of HTTP methods, including GET, POST, PUT, and DELETE, and allows developers to send requests with various headers and parameters. One of the key benefits of Postman is its ease of use. Its intuitive interface makes it easy to create and send requests to APIs, and it provides a variety of features for testing and debugging, such as built-in support for testing scripts and automation. Another benefit of Postman is its flexibility. It supports a wide range of APIs and protocols, including REST, SOAP, and GraphQL, and provides a number of tools for working with different data formats, such as JSON and XML. In addition, Postman is available for free with basic features and provides paid plans with additional functionality for enterprise use. In summary, Postman is a powerful and user-orderly tool that provides developers with an easy and flexible way to test and debug APIs.

2.2 Functional requirements

Client side

2.2.1. Frontend UI using Atomic Design

Building Frontend using Atomic Design is a methodology for designing and building user interfaces that breaks down the design into smaller, more manageable pieces. It is a popular approach used by developers and designers to create reusable, scalable, and maintainable UI components.

The Atomic Design methodology is based on the idea that user interfaces can be broken down into :

1. Atoms
2. molecules
3. organisms
4. templates
5. pages.

Atoms are the smallest building blocks of a UI, such as buttons, input fields, or icons. Molecules are groups of atoms that work together to form a more complex component, such as a search bar or a dropdown menu. Organisms are combinations of molecules that make up more complex sections of a UI, such as a header or a footer. Templates are collections of organisms that define the layout of a page, while pages are specific instances of a template that represent actual content.

Using the Atomic Design methodology can help designers and developers create consistent and cohesive UI components that can be easily reused across different projects. By breaking down the design into smaller pieces, it becomes easier to manage and update the UI, as changes can be made to individual components without affecting the entire system. To implement Atomic Design, designers and developers can use a variety of tools and frameworks, such as React, Vue, or Angular, to create UI components that can be reused across different projects. They can also use design systems to define the visual and

functional characteristics of each component, such as colors, typography, and spacing.

Overall, Atomic Design is a powerful methodology for building frontend UIs that can help designers and developers create more consistent, maintainable, and scalable user interfaces.

Admin side

2.2.2. Authentication and Authorization

Authentication is one of the most important features in a software application.

Authentication is the ability to prove that a user or application is genuinely who that person or what that application claims to be, meanwhile the Authorization protects critical resources in a system by limiting access only to authorized users and their applications. It prevents the unauthorized use of a resource or the use of a resource in an unauthorized manner.

Authentication and authorization can be realized by implementing sign-in feature. In this project the writer use JWT (Jason Web Token) as the authentication.

JSON Web Token (JWT) is a popular method of implementing authentication in web applications. It is a compact and self-contained mechanism for transmitting user authentication information between the client and server as a JSON object. The JWT standard defines a way to securely transmit information between parties as a JSON object. The basic idea behind JWT is to use a token to verify a user's identity, rather than sending the user's credentials, such as a username and password, with each request. The token is generated on the server after the user logs in and includes user information, such as the user ID and roles. The token is then sent to the client and stored in local storage or a cookie.

On subsequent requests, the client sends the token in the header of the HTTP request, where it can be verified by the server. The server uses a secret key to validate the token and extract the user information from it. If the token is valid, the server processes the request and sends a response back to the client. One of the key advantages of JWT is that it is stateless, meaning that the server does not need to keep track of the user's session or store any data on the server. This makes JWT a scalable and efficient method of implementing authentication in web applications.

In summary, JWT is a secure and efficient method of implementing authentication in web applications. It provides a scalable and stateless approach to transmitting user authentication information between the client and server, which can help improve the performance and scalability of web applications.

2.2.3. Login

This requirement enable admin to open the dashboard admin then user can arrange product's requirement, for example the price, the category, even discount, etc. But before Login, to record a user into our software, we need to implement registration. Here the registration is not show up to the user (need development) later, so registration step is manually added by the system on database. The data of the login user will be used for authentication and authorization process of the software, thus making login is an important requirement to have. Login should contain email and password.

| Field Name | Description | Data Format |
|------------|-----------------|--------------|
| email | User's email | Email's type |
| password | User's password | string |

Table 2.1 Login Requirement filed

2.2.4. CRUDE

This requirement enable admin to create subject matter, category files, nominal product, voucher, bank type, and payment method.

| Field Item | Description | Field Data |
|----------------|---|---|
| Category | Name Category files that become product | No, Name Category, Action |
| Nominal | Coin Price (discount) | No, Coin Name, Amount Coin, Coin Price, Action |
| Voucher | promo | No, Course Name, Category, Nominal, Status, Action |
| Bank | Bank information transaction | No, Owner Name, Bank Name, Bank Number, Action |
| Payment Method | Type of Payment | Number, Type of Payment, |

| | | |
|-------------|--------------------------|---|
| | | Bank, Status, Action |
| Transaction | List of user do transact | No, Course Name, Item, Total Price, Status, Buyer name, Action. |

Table 2.2 CRUDE requirement table

2.3 Non-Functional Requirements

2.3.1 Performance

As the purpose of this study to develop a sample prototype and will not be released in public, the application is expected to be able to handle 100 concurrent users at the same time. According to Blazor Documentation, each circuit uses approximately 250 kilobytes of memory for a minimal „Hello-World“ application. Blazor Documentation recommend to reserver 273 kilobyte of memory per user, so this application expected to consume at least 27.3 megabyte of memory.

2.3.2 Reliability

The application should not be crashed when user input is incorrect type, the system should be able to handle that, for example if the expected input is email but the user does not input an email address, the system should show error to the user prompt. This application also should guarantee that the text message should be sent exactly one time to prevent duplication of message.

2.3.3. Security

Due to the natural connection of authentication utilize time period of user access the platform, every time period user use a webpage, it will be treated as different connection, so the user needs to re-login. This improves the security as the login information is not saved on the device.