**Chapter 5**

**Technical Specifications**

1. **Vue.js**

Vue is a JavaScript framework for building user interfaces. It builds on top of standard HTML, CSS and JavaScript, and provides a declarative and component-based programming model that helps you efficiently develop user interfaces, be it simple or complex.

Vue is a framework and ecosystem that covers most of the common features needed in frontend development. But the web is extremely diverse - the things we build on the web may vary drastically in form and scale. With that in mind, Vue is designed to be flexible and incrementally adoptable. Depending on your use case, Vue can be used in different ways:

* Enhancing static HTML without a build step
* Embedding as Web Components on any page
* Single-Page Application (SPA)
* Fullstack / Server-Side-Rendering (SSR)
* Jamstack / Static-Site-Generation (SSG)
* Targeting desktop, mobile, WebGL or even the terminal

1. **Tailwind CSS**

Tailwind CSS is a "utility-first" CSS framework that provides a deep catalogue of CSS classes and tools that lets you easily get started styling your website or application. The underlying goal is that as you're building your project, you don't need to deal with cascading styles and worrying about how to override that 10-selector pileup that's been haunting your app for the last 2 years.

Tailwind's solution is to provide a wide variety of CSS classes that each have their own focused use. Instead of a class called .btn that is created with a bunch of CSS attributes directly, in Tailwind, you would either apply a bunch of classes like bg-blue-500 py-2 px-4 rounded to the button element or build a .btn class by applying those utility class to that selector.

1. **NOSQL**

NoSQL databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

As storage costs rapidly decreased, the amount of data that applications needed to store and query increased. This data came in all shapes and sizes — structured, semi-structured, and polymorphic — and defining the schema in advance became nearly impossible. NoSQL databases allow developers to store huge amounts of unstructured data, giving them a lot of flexibility.

Each NoSQL database has its own unique features. At a high level, many NoSQL databases have the following features:

* Flexible schemas
* Horizontal scaling
* Fast queries due to the data model
* Ease of use for developers

1. **Git**

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning-fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

1. **GitHub**

At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, you need to know two connected principles:

* Version control
* Git

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.

GitHub’s interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technically savvy and use of the command line.

GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects – like writing books. Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

As a company, GitHub makes money by selling hosted private code repositories, as well as other business-focused plans that make it easier for organizations to manage team members and security.

1. **CRUD**

Within computer programming, the acronym CRUD stands for create, read, update and delete. These are the four basic functions of persistent storage. Also, each letter in the acronym can refer to all functions executed in relational database applications and mapped to a standard HTTP method, SQL statement or DDS operation.

It can also describe user-interface conventions that allow viewing, searching and modifying information through computer-based forms and reports. In essence, entities are read, created, updated and deleted. Those same entities can be modified by taking the data from a service and changing the setting properties before sending the data back to the service for an update. Plus, CRUD is data-oriented and the standardized use of HTTP action verbs.

1. **FirebaseFirestore**

Firestore is a NoSQL document database built for automatic scaling, high performance, and ease of application development. While the Firestore interface has many of the same features as traditional databases, as a NoSQL database it differs from them in the way it describes relationships between data objects

1. **FirebaseAuth**

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.