CEP-CCIT FAKULTAS TEKNIK
UNIVERSITAS INDONESIA

Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET

FINAL PAPER
INTEGRATED SYSTEM

RIKI AWAL SYAHPUTRA
2120010136

INFORMATION TECHNOLOGY PROGRAM
INTERNET-BASED SYSTEM AUTOMATION
JAKARTA
AUGUST 2023

UNIVERSITAS INDONESIA

CEP-CCIT

FAKULTAS TEKNIK

CEP-CCIT FAKULTAS TEKNIK
UNIVERSITAS INDONESIA

Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET

FINAL PROJECT
INTEGRATED SYSTEM PROJECT

Submitted as one of the graduation requirements from the 2 Year Professional Program in Information Technology CEP-CCIT FTUI

RIKI AWAL SYAHPUTRA
2120010136

INFORMATION TECHNOLOGY PROGRAM
INTERNET-BASED SYSTEM AUTOMATION
JAKARTA
AUGUST 2023

ORIGINALITY DECLARATION PAGE

This Integrated System Final Project paper is the result of my own work,
and all sources both cited and referred to
have been properly stated by me.

Name	**: Riki Awal Syahputra**
No Registration	**: 2120010136**
Signature	**:**
Date	**: August 18th 2023**

ENDORSEMENT PAGE

This paper was submitted by:

Name : Riki Awal Syahputra

No. Registration : 2120010136

Study Program : Information Technology – Internet-Based System Automation

Paper Title : Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET

EXAMINER

Advisor : Listyo Edi Prabowo, S.T.,M.T. ( )

Examiner : Tirta Akdi Toma Mesoya Hulu S. Kom. ( )

Set in : Jakarta

Date : August 18th 2023

ACKNOWLEDGMENTS

My gratitude goes to God Almighty, because of His blessings and grace, I can finish this paper. The writing of this paper was carried out in order to fulfill one of the graduation requirements in the FTUI CEP-CCIT Information Technology program. I realize that without the help and guidance of various parties, from the lecture period to the preparation of this paper, it would be very difficult for me to complete this paper. Therefore, I would like to thank:

1. Mr. Listyo Edi Prabowo, S.T.,M.T., as the supervising lecturer who has provided the time, energy, and thoughts to direct me in the preparation of this paper;
2. PT. IDX Consulting who have helped a lot in the effort to obtain the data I need;
3. My parents and family who have provided material and moral support; and
4. Friends who have helped me a lot in completing this paper.

Finally, I hope that God Almighty is pleased to repay all the kindness of all those who have helped. Hopefully, this paper will bring benefits to the development of science.

Jakarta, 18 August 2023
Writer

ABSTRACT

Name : Riki Awal Syahputra
No registration : 2120010136
Title : Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET
Advisor : Listyo Edi Prabowo, S.T.,M.T.

Rapid progress and development in the business world has driven increased scalability and accelerated operations to support economic growth, both individually, in groups and at the national level. In this context, the use of technology, especially in the form of digitization, has proven itself as an effective approach to achieving this goal. Digital transformation, which is primarily reflected through online platforms, continues to evolve to achieve flexibility in development, reduced time and costs, and attractive results for consumers.

Especially in the development of online platforms, the NextJS framework based on JavaScript and the Tailwind CSS design approach have become highly recommended solutions. NextJS enables high-efficiency web application development through support for server-side rendering (SSR) and static site generation (SSG). Meanwhile, Tailwind CSS, as a class-based layout framework, enables fast responsive interface design through the use of utility classes.

It is important to note that on a larger scale, such as large enterprises, web development frameworks such as ASP.NET have a significant role. ASP.NET stands out in the development of large-scale applications with impressive processing speeds. In addition, its advanced features support server-based development and solid integration with the Microsoft services ecosystem.

However, in choosing a technology, it is necessary to consider the business context, the complexity of the project, as well as the development of the latest technology trends. Successful web development requires an in-depth understanding of the principles of design, security, performance, and compliance with a company's long-term goals.

Keywords:
Digitalization, scalability, web development

ABSTRAK

Nama : Riki Awal Syahputra
No registrasi : 2120010136
Judul : Pendekatan Praktis dengan Next.js & Tailwind CSS: Menjelajahi Tren Pengembangan Web Modern Bersama Tinjauan Akademik ASP.NET
Pembimbing : Listyo Edi Prabowo, S.T.,M.T.

Kemajuan dan perkembangan yang pesat di dunia bisnis mendorong peningkatan skalabilitas dan percepatan operasional guna mendukung pertumbuhan ekonomi, baik secara individual, kelompok, maupun pada tingkat nasional. Dalam konteks ini, pemanfaatan teknologi, khususnya dalam bentuk digitalisasi, telah membuktikan diri sebagai salah satu pendekatan yang efektif untuk mencapai tujuan tersebut. Transformasi digital, yang terutama terefleksi melalui platform daring, terus mengalami evolusi guna mencapai fleksibilitas dalam pengembangan, pengurangan waktu dan biaya, serta hasil yang menarik bagi konsumen.

Terutama dalam pengembangan platform daring, kerangka kerja NextJS yang berbasis JavaScript dan pendekatan desain Tailwind CSS telah menjadi solusi yang sangat dianjurkan. NextJS memungkinkan pengembangan aplikasi web dengan efisiensi tinggi melalui dukungan terhadap renderisasi di sisi server (server-side rendering/SSR) dan pembuatan situs statis (static site generation/SSG). Sementara itu, Tailwind CSS, sebagai kerangka kerja tata letak berbasis kelas, memungkinkan desain antarmuka yang responsif dengan cepat melalui penggunaan kelas utilitas.

Penting untuk dicatat bahwa dalam skala yang lebih besar, seperti perusahaan besar, kerangka kerja pengembangan web seperti ASP.NET memiliki peran signifikan. ASP.NET menonjol dalam pengembangan aplikasi skala besar dengan kecepatan pemrosesan yang impresif. Selain itu, fitur-fitur canggihnya mendukung pengembangan berbasis server dan integrasi yang solid dengan ekosistem layanan Microsoft.

Meskipun demikian, dalam memilih teknologi, perlu mempertimbangkan konteks bisnis, kompleksitas proyek, serta perkembangan tren teknologi terbaru. Pengembangan web yang berhasil memerlukan pemahaman mendalam tentang prinsip-prinsip desain, keamanan, kinerja, serta kesesuaian dengan sasaran jangka panjang perusahaan.

Katakunci:
Digitalisasi, skalabilitas, pengembangan web

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

In today's increasingly complex and competitive business landscape, the success of a company is no longer solely determined by the products or services it offers. Instead, it hinges upon its ability to optimize profitability through data-driven strategies and decision management. Recognizing the imperative of this approach, IDX Partner emerges as a prominent provider of data analytics consulting and solutions.

With a specialization in data analytics, IDX Partner serves as a strategic ally for companies navigating the challenges of the modern business environment. The company has established itself as a pivotal partner, aiding enterprises in harnessing the power of data to drive profitability. Its comprehensive solutions encompass various critical aspects of business management, including risk assessment and scoring, customer retention management, combatting financial crime, collection and recovery strategies, decision engine implementation, and business intelligence integration.

IDX Partner's diverse portfolio of solutions caters to clients spanning a multitude of industries, including financial services, telecommunications, manufacturing, and retail. This versatility allows companies across different sectors to rely on IDX Partner as a key collaborator in formulating data-informed decisions. Through this approach, enterprises are empowered to enhance operational efficiency, gain insights into their operations, and bolster their competitiveness within dynamic markets.

The company's track record exemplifies its competency in delivering impactful data analytics solutions that provide tangible benefits to its partnered clients. Leveraging its extensive experience and a team of skilled professionals well-versed in various facets of data analytics, IDX Partner is committed to consistently delivering substantial value to its clients. By doing so, it supports their business growth trajectory and navigates them toward success in an economy driven by data.

1.2 Problem Domain

Based on the background that has been described, several problems can be formulated, including:

A. NextJS Exploration and Implementation:

1. Understanding the concept and features of NextJS.
2. Assessing the advantages and disadvantages of using NextJS.
3. Identifying the companies that have implemented NextJS.
4. Developing a TODO LIST project using NextJS.
5. Research references for the NextJS project.
6. Designing the database structure for the project.
7. Designing the website interface.
8. Implementing the project based on the chosen design.
9. Exploring and analyzing the used software and technologies.
10. Evaluating the results and outcomes of the NextJS project.
11. Examining the dashboard and advanced case references.
12. ASP.NET Exploration and Abstraction Programming:

B. Understanding the concept and variants of ASP.NET.

1. Analyzing the advantages and disadvantages of utilizing ASP.NET.
2. Identifying the companies that have adopted ASP.NET.
3. Exploring the abstraction programming techniques in ASP.NET.
4. Understanding concepts like delegates, events, generics, extensions methods, lambda expressions, LINQ, and asynchronous programming.
5. Investigating the broader implementation of ASP.NET across various contexts.
6. Conclusion and Suggestions:

C. Drawing conclusions based on the exploration of NextJS and ASP.NET.

D. Providing suggestions for improvements or future directions related to NextJS and ASP.NET.

1.3 Writing Objective

This paper aims to comprehensively explore and analyze the role of technology-driven advancements, particularly in the context of digitalization, in enhancing scalability and accelerating business operations to foster economic growth. The primary objectives of this paper include:

1. **Examine the Impact of Digital Transformation**: Investigate how the application of digitization in the business landscape has contributed to increased scalability and accelerated operations, thereby supporting economic growth at individual, group, and national levels.
2. **Evaluate NextJS and Tailwind CSS Frameworks**: Conduct an in-depth assessment of the NextJS framework based on JavaScript and the design approach of Tailwind CSS. Analyze their effectiveness in enabling efficient and responsive web application development, considering features such as server-side rendering (SSR), static site generation (SSG), and utility class utilization.
3. **Assess the Role of ASP.NET in Large-Scale Development**: Evaluate the significance of ASP.NET as a web development framework, particularly in the context of large enterprises. Explore its capabilities in facilitating the development of high-speed, large-scale applications and its integration with the Microsoft services ecosystem.
4. **Highlight Considerations in Technology Selection**: Examine the factors that should be taken into account when choosing a technology for web development. Consider the importance of aligning technology choices with the business context, project complexity, and emerging technology trends.
5. **Understand the Prerequisites for Successful Web Development**: Investigate the critical elements for successful web development, including a deep understanding of design principles, security measures, performance optimization, and the alignment of development strategies with long-term business objectives.
6. **Fulfill one of the graduation requirements for the CEP-CCIT**: Information Technology Study Program, Faculty of Engineering, University of Indonesia.

By addressing these objectives, this paper seeks to provide a comprehensive understanding of how technology, particularly digitalization and web development frameworks, plays a pivotal role in enhancing business scalability, accelerating operations, and contributing to overall economic growth.

1.4 Writing Methodology

The methodology for this paper is based on a combination of internet research, personal experience gained during an internship at a company, and the knowledge acquired from the internship. The approach integrates qualitative analysis of existing data, practical insights from real-world experience, and theoretical understanding obtained through formal education. The methodology can be broken down into the following key components:

1. **Literature Review and Internet Research:**
   Extensive internet research will be conducted to gather relevant academic papers, articles, and reports related to the role of technology in business scalability and acceleration. These sources will be used to establish a foundational understanding of the subject and to identify key trends, concepts, and best practices.

2. **Case Study from Internship Experience:**
   Insights and observations gained during the internship at a company will be leveraged to provide real-world context to the paper. This will include anecdotal evidence, practical examples, and firsthand accounts of how technology, digitalization, and specific frameworks were employed within the company's operations to enhance scalability and accelerate business processes.

3. **Application of Theoretical Knowledge:**

   The theoretical understanding of relevant concepts acquired during formal education, particularly from the knowledge shared by the hosting company, will be integrated into the analysis. These concepts include principles of design, security, performance optimization, and alignment with long-term business objectives.

4. **Data Synthesis and Analysis:**

   The data gathered from internet research, internship experience, and theoretical knowledge will be synthesized and organized. This synthesis will be used to draw connections between theoretical frameworks and practical applications, demonstrating how technology has contributed to scalability and operational acceleration in different business contexts.

5. **Comparative Analysis of Frameworks:**

   A comparative analysis of different frameworks, such as NextJS, Tailwind CSS, and ASP.NET, will be conducted based on the information gathered from internet research and internship experience. This analysis will explore the strengths, weaknesses, and appropriateness of each framework for specific business scenarios.

6. **Discussion of Findings:**

   The findings from the research, internship experience, and theoretical insights will be discussed in the context of the research objectives. The discussion will highlight key takeaways, patterns, and lessons learned from the application of technology in business scalability and acceleration.

7. **Conclusion and Implications:**

   The paper will conclude by summarizing the main findings, emphasizing the significance of technology in business growth, and discussing the implications for future practices and research in the field.

The methodology described above combines both primary and secondary data sources to provide a comprehensive exploration of the role of technology in enhancing business scalability and accelerating operations. The triangulation of information from internet research, practical experience, and theoretical knowledge ensures a well-rounded and informed analysis.

1.5 Place and Time of Activity

The time and place of implementation are as follows:

Place	: Remote from (Millennium Centennial Center Level 41 Jl. Gen. Sudirman Kav. 25, Jakarta, Indonesia 12920)
Date	: 1st June 2023 – 31st July 2023
Working Days	: Monday to Sunday (flexible)
Hours of Operation	: 08.00 – 17.00

1.6 Writing Framework

This report is divided into 4 chapters, each of which explains the following:

CHAPTER I INTRODUCTION

This chapter describes the background, the purpose of writing, the scope of the problem, the place and time of implementation, the writing methodology, and the reporting systematics.

CHAPTER II LITERATURE REVIEW

This chapter describes the history of the company, purpose, vision and mission of the company, and company logo.

CHAPTER III RESULTS AND PROBLEM ANALYSIS

In this chapter describes the discussion of the formulation of the problem

CHAPTER IV CONCLUSIONS AND RECOMMENDATIONS

This chapter contains the conclusions on this final project report. The conclusion briefly presents all the results obtained during the Field Work Practice

CHAPTER 2

LITERATURE REVIEW

2.1 Historical Overview of Company

Founded in 2002, id/x Partners has emerged as a dynamic entity, spearheaded by former bankers and seasoned management consultants whose expertise spans a wide spectrum of domains, including credit cycle and process management, scoring development, and performance management. With their profound insights and wealth of experience, the team brings a remarkable blend of knowledge that has been harnessed across Asia and Australia, spanning diverse industries that encompass financial services, telecommunications, manufacturing, and retail.

The core essence of id/x Partners revolves around its commitment to offering avant-garde consulting services. These services are not only rooted in harnessing the power of data analytics and decisioning (DAD) solutions but also intertwine seamlessly with a comprehensive, integrated risk management and marketing discipline. This holistic approach serves as a catalyst, empowering clients to harness the full potential of their portfolios' profitability and streamline their intricate business processes.

The cornerstone of id/x Partners' value proposition lies in its seamless integration of consulting expertise and cutting-edge technology solutions. This symbiotic amalgamation positions id/x Partners as a definitive one-stop service provider, uniquely poised to address the multifaceted needs of modern businesses. By weaving together deep industry insights, advanced data analytics, and pragmatic decisioning strategies, id/x Partners stands as a beacon of innovation and efficiency, fostering not just growth but also resilience for its esteemed clientele.

2.2 Purpose, Vision, and Mission

A. Purpose

Delivering the right solution and superior service for the customer to realize maximum business value.

B. Vision

provides consulting services that specializes in utilizing data analytic and decisioning (DAD) solutions combined with an integrated risk management and marketing discipline to help clients optimize the portfolio profitability and business process.

C. Mission

- C: Customer
  Deliver right solution and superior service for the customer to realize maximum business value
- H: Honorable
  Focus on profitable growth without sacrificing integrity & quality.
- A: Agile
  Respond quickly and meaningfully to new opportunities (or threats).
- M: Mentorship
  Develop team's continuous personal and professional growth.
- P: Proactive
  Put forth new ideas in order to improve our work.
- I: Innovative
  Cultivate a creative working environment to solve tough problems.
- O: Ownership
  Exercise accountability and ownership of our work and responsibilities.
- N: Numeric
  Make decisions based on facts and numbers.

CHAPTER 3

RESULT AND PROBLEM ANALYSIS

3.1 Exploring NextJS

In the ever-evolving landscape of web development, NextJS stands as a beacon of innovation, redefining the paradigms of building dynamic and responsive web applications. This sub-chapter embarks on a comprehensive exploration of NextJS, unravelling its core concepts, features, and the pivotal role it plays in shaping the modern web development ecosystem. By delving into the depths of NextJS, writers uncover the intricacies that make it a formidable choice for developers seeking enhanced performance, efficiency, and interactivity in their web projects.

3.1.1 What is NextJS

In the realm of modern web development, NextJS has emerged as a powerful and dynamic framework that revolutionizes the way writers approach building web applications. This sub-chapter delves into the fundamental aspects of NextJS, offering a comprehensive understanding of its definition, advantages, disadvantages, and its implementation across various industries. By exploring the essence of NextJS, writers embark on a journey to uncover the innovative solutions it brings to the forefront of web development.

3.1.1.1 NextJS Definition

At the heart of contemporary web development, NextJS emerges as a pivotal framework that empowers developers to create seamless, efficient, and highly performant web applications. NextJS is a robust React framework that introduces a layer of server-side rendering (SSR) and static site generation (SSG) capabilities, elevating the user experience by leaps and bounds.

Unlike traditional single-page applications, NextJS optimizes initial load times by pre-rendering pages on the server, resulting in faster page rendering and improved search engine optimization (SEO). This approach brings about a unique blend of dynamic interactivity and the benefits of static content delivery, allowing websites to effortlessly cater to user demands while maintaining high-speed performance.

NextJS further enhances its flexibility with an intuitive routing system and smart data fetching options, ensuring that developers can seamlessly integrate their projects with various data sources. With its built-in support for TypeScript and Babel, NextJS fosters an environment of convenience and versatility, allowing developers to write code in the language they are most comfortable with.

In essence, NextJS heralds a new era of web development, where the boundaries between server-rendered and client-rendered approaches blur, and the focus shifts towards delivering an unparalleled user experience. Its ability to effortlessly marry performance, interactivity, and SEO-friendliness positions NextJS as a cornerstone of modern web development, shaping the digital landscapes of today and beyond.

3.1.1.2 Advantage and Disadvantage

In the journey through the landscape of modern web development, it is imperative to dissect both the strengths and weaknesses of the tools at writer disposal. In this section, we delve into the advantages and disadvantages inherent to Next.js, shedding light on its capabilities and potential limitations. By critically evaluating these aspects, we can gain a comprehensive understanding of how Next.js can impact the development process and shape the final outcomes of writer’s projects.

A. Advantages:

- Enhanced Performance: The incorporation of server-side rendering (SSR) and static site generation (SSG) significantly reduces initial load times, leading to faster rendering and improved user experiences.
- SEO Friendliness: The pre-rendered content and efficient crawling of pages by search engines contribute to improved search engine optimization (SEO).
- Intuitive Routing: NextJS boasts a streamlined routing system that simplifies navigation and enhances user engagement, vital for creating seamless user experiences.
- Data Fetching Flexibility: The framework provides diverse data fetching strategies, including server-side data fetching and static generation, catering to varying project needs.
- Developer-Friendly: With support for TypeScript, Babel, and hot module replacement, NextJS fosters an environment that promotes ease of development and code maintainability.
- Hybrid Approach: NextJS seamlessly bridges the gap between server-side rendering and client-side rendering, offering the best of both worlds for interactive applications.

B. Disadvantages:

- Learning Curve: NextJS introduces a learning curve for developers transitioning from traditional React development due to its unique server-side rendering and routing concepts.
- Build Complexity: Implementing server-side rendering and static site generation can sometimes lead to increased build complexity, demanding careful planning and execution.
- Server Load: While SSR offers improved performance, it may lead to increased server load, especially for applications with extensive traffic and complex rendering requirements.

- Dependency Management: Integrating third-party libraries and dependencies may require special considerations to ensure compatibility with NextJS's server-side rendering approach.

3.1.1.3 NextJS and Tailwind CSS

In the realm of modern web development, the strategic pairing of frameworks often leads to groundbreaking outcomes. This sub-chapter delves into the synergistic fusion of NextJS and Tailwind CSS, two potent tools that, when combined, amplify the development process's speed, scalability, and the creation of captivating user interfaces.

A. NextJS: Enabling Performance and Scalability

NextJS, as a dynamic React framework, introduces server-side rendering (SSR) and static site generation (SSG) capabilities. This foundational architecture significantly improves initial load times, a vital factor in enhancing user experience. The seamless integration of dynamic interactivity and high-speed performance aligns perfectly with Tailwind CSS's principles of rapid design and development.

B. Tailwind CSS: Streamlined Interface Design

Tailwind CSS, a utility-first CSS framework, epitomizes efficiency in interface design. By harnessing utility classes, developers can swiftly create responsive layouts and visually captivating components. This speed is vital in the ever-accelerating digital landscape, where users demand seamless experiences that load swiftly and remain engaging across various devices.

C. A Synergy That Fuels Efficiency and Appeal

When utilized together, NextJS and Tailwind CSS form a potent alliance that addresses the key aspects of modern web development:

1. **Accelerated Production**: The combination of NextJS's optimized rendering and Tailwind CSS's utility classes accelerates project development. Developers can focus on creating features and content without getting bogged down by complex styling intricacies.
2. **Scalability and Performance**: With NextJS's SSR and SSG capabilities, the framework ensures optimal performance even as applications scale. Tailwind CSS's lightweight utility classes further contribute to streamlined performance.
3. **Interactive and Engaging UI**: Tailwind CSS empowers developers to create visually captivating interfaces, while NextJS's interactivity enhances user engagement, making applications more appealing and compelling.
4. **Responsive Design**: Tailwind CSS's utility classes simplify the creation of responsive layouts, ensuring that applications adapt seamlessly to various screen sizes and devices.
5. **Code Maintainability**: Both frameworks encourage structured and modular development, promoting clean and maintainable codebases that facilitate collaborative work and future updates.

In the convergence of NextJS and Tailwind CSS, developers find an ecosystem where their productivity is enhanced, scalability is ensured, and interfaces are intuitively engaging. This powerful synergy transforms the development journey, leading to applications that not only perform exceptionally but also captivate users with their seamless experiences and visually enticing designs.

3.1.1.4 The Company who Implemented NextJS

The practical adoption of NextJS in various esteemed enterprises showcases its robust capabilities and impact on modern web development. Here, writers present a compilation of companies that have harnessed the power of NextJS to elevate their web applications, enhancing performance, interactivity, and user experience.

1. **Hulu**: The streaming giant Hulu optimizes its website's performance and interactivity through NextJS. Leveraging server-side rendering, Hulu ensures swift content delivery, which is pivotal in the competitive streaming industry.
2. **TicketSwap**: TicketSwap, a hub for ticket trading, ensures a seamless user experience by effectively employing NextJS. Combining server-side rendering and prefetching, TicketSwap guarantees swift access to event information, enhancing user engagement.
3. **Marvel**: Marvel, a design and prototyping platform, employs NextJS to provide a streamlined experience. Prioritizing rapid loading and dynamic interactivity, NextJS enhances the usability of Marvel's tools.
4. **Typeform**: Revered for its survey and form-building prowess, Typeform harnesses NextJS to elevate user interfaces and experiences. The fusion of server-side rendering and client-side interactivity augments Typeform's responsiveness.
5. **TikTok**: Embracing NextJS, TikTok bolsters its web platform's performance and user engagement. The framework's capabilities contribute to the seamless delivery of TikTok's dynamic content.
6. **Hashnode**: Hashnode leverages NextJS to enhance its developer community platform. The framework's integration supports the efficient rendering of Hashnode's diverse content.

7. **Twiitch Mobile**: Twiitch Mobile benefits from NextJS's capabilities in optimizing mobile web experiences. The framework aids in delivering smooth interactions and content presentation.
8. **Binance**: As a prominent cryptocurrency exchange, Binance harnesses NextJS to ensure the optimal performance of its web-based platform. The framework's features contribute to a seamless trading experience.
9. **Ticket Master**: Ticket Master, a renowned ticketing platform, embraces NextJS to enhance its web interface's interactivity and responsiveness. This enhances the ticket purchasing experience for users.
10. **Deliveroo**: Deliveroo, a food delivery platform, adopts NextJS to optimize its web-based service. The framework's capabilities contribute to swift order placement and improved user engagement.
11. **AT&T**: NextJS enhances AT&T's web presence, delivering a seamless browsing experience to its users. The framework's features align with AT&T's commitment to user satisfaction.
12. **PlayStation** Competitor Center: The PlayStation Competitor Center leverages NextJS to enhance its web-based features for gamers. The framework's interactivity contributes to an engaging user experience.
13. **InVision**: InVision employs NextJS to ensure the dynamic and responsive nature of its design platform. The framework's capabilities align with InVision's commitment to user-centered design.
14. **Pusher**: Pusher harnesses NextJS to enhance its real-time communication platform's web interface. The framework's features contribute to responsive and efficient data transfer.
15. **SumUp**: SumUp, a payment solution provider, optimizes its web application's efficiency and reliability through NextJS. This enhances the user experience during payment processes.
16. **Nubank**: Nubank employs NextJS to provide a seamless and engaging interface for its financial services. The framework's capabilities enhance the accessibility of Nubank's offerings.

17. **Verge**: The Verge, a media platform, integrates NextJS to improve the performance of its web-based content delivery. This contributes to a smooth browsing experience for readers.

These companies demonstrate the diverse applications and benefits of NextJS in modern web development. The framework's capabilities contribute to enhanced performance, interactivity, and user engagement across various industries and platforms.

3.1.2 Implement NextJS Project with TODO LIST

Embarking on a practical exploration of NextJS, writers delve into the development of a web application featuring a Todo List using this powerful framework. This sub-section chronicles the systematic process undertaken to create an interactive and dynamic Todo List, showcasing the efficiency and flexibility of NextJS.

3.1.2.1 Reasearching of Reference

In the pursuit of developing a dynamic and efficient TodoList application, thorough research is conducted to explore the implementation of a Kanban board-inspired user interface. This sub-section delves into the process of sourcing references from popular applications like Trello and ClickUp, known for their successful integration of Kanban board functionality.

1. **Understanding Kanban Methodology**: The research begins with a comprehensive understanding of the Kanban methodology. Kanban boards are renowned for their visual representation of tasks, enabling users to track progress, manage workflows, and enhance collaboration.

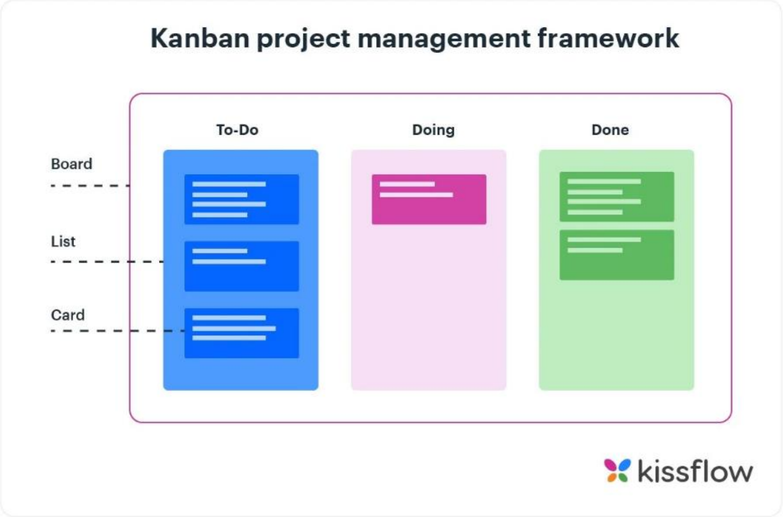


Figure 3. 1 Kanban Illustraton (REF: https://kissflow.com/project/agile/kanban-methodology/)

2. **Analyzing Trello's Approach**: Trello, a widely-used project management tool, embodies the Kanban approach. By examining Trello's user interface and features, writers gain insights into how tasks are organized, visualized, and managed using boards, lists, and cards.

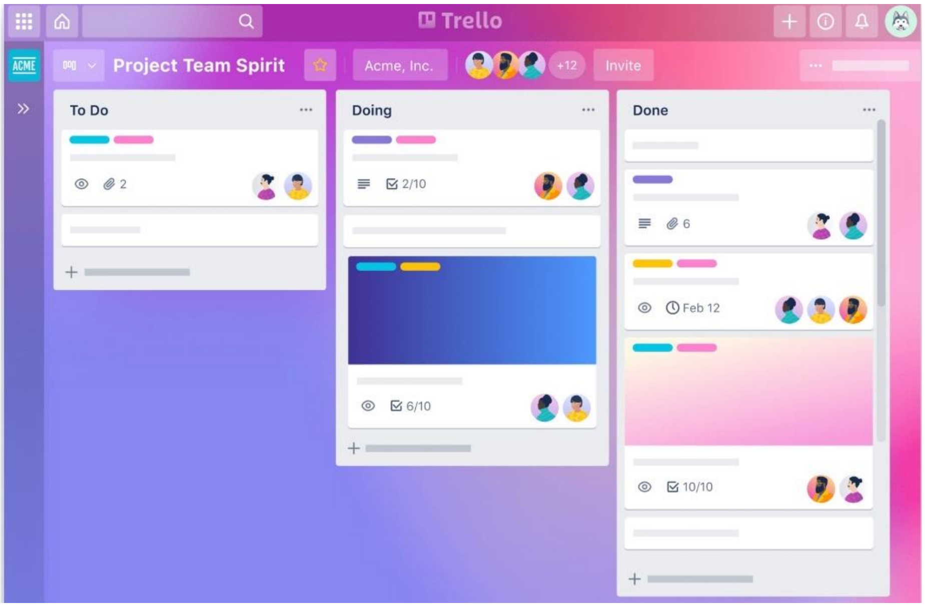
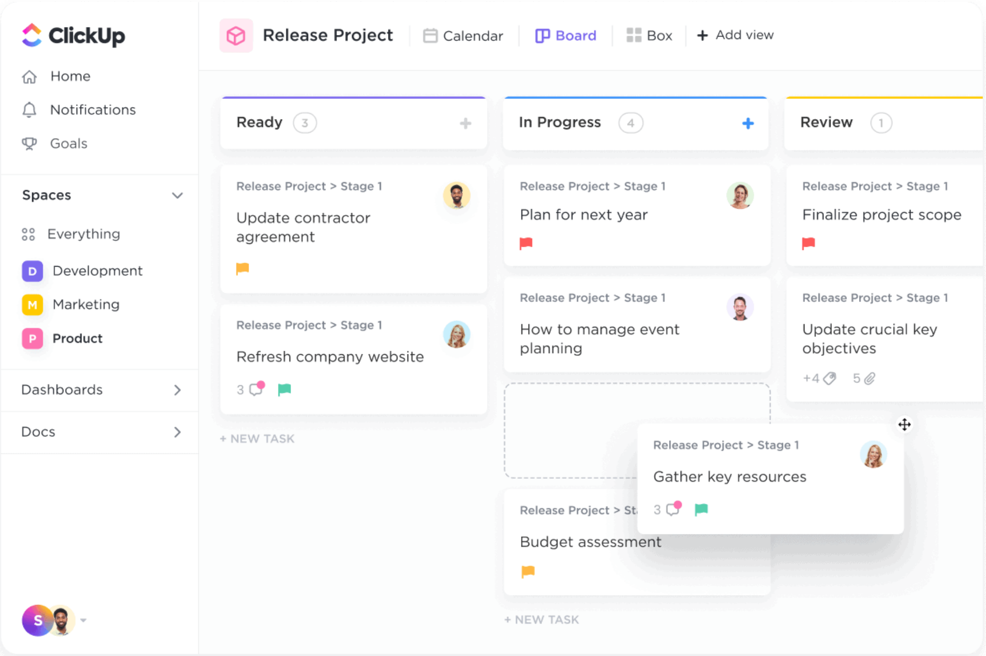


Figure 3. 2 Trello Dashboard (REF: https://trello.com/)

3. **Exploring ClickUp's Integration**: ClickUp, another prominent project management platform, also employs Kanban boards. By exploring ClickUp's implementation, writers uncover the unique features, customization options, and user interactions that contribute to an effective Kanban board experience.



Release Project > Stage 1
Update contractor agreement

Release Project > Stage 1
Refresh company website
3

Release Project > Stage 1
Plan for next year

Release Project > Stage 1
How to manage event planning

Release Project > Stage 1
Gather key resources
3

Release Project > St
Budget assessment

Release Project > Stage 1
Finalize project scope

Release Project > Stage 1
Update crucial key objectives
+4 5

3. 3 ClickUp Dashboard (REF: https://clickup.com/)

4. **Identifying Key Components**: From both Trello and ClickUp, key components of a Kanban board are identified. These include columns representing task stages, movable cards, drag-and-drop functionality, and real-time updates, and for the project, writer's going to wrap the project as simple as it can, so writer tend to choose these key components:

- Input task functionality
- Todo list card
- Edit functionality
- Delete functionality

5. **Adapting to NextJS**: The research process involves deciphering how the Kanban board concept can be adapted within the NextJS framework. This includes identifying libraries, components, and strategies that facilitate the dynamic rendering of columns and cards.

   a) Adapting to NextJS: The Integration of Kanban Board Functionality

      The adaptation of the Kanban board concept within the NextJS framework involves a meticulous exploration of how the core functionalities of adding tasks, editing, and deleting tasks can be seamlessly integrated. To accomplish this, a well-defined architectural approach is employed to ensure the efficient realization of the TodoList's capabilities while utilizing a JSON server to store task data.

   b) Architectural Overview:

      1) The architecture of the TodoList application combines the capabilities of NextJS, React, and a JSON server to deliver a responsive and interactive user experience. The chosen architecture ensures the separation of concerns, promoting modularity, maintainability, and scalability.

      2) NextJS Framework: The NextJS framework serves as the foundation for the application. Leveraging its server-side rendering (SSR) and client-side rendering (CSR) capabilities, the application achieves efficient page loading, rendering, and interactivity. Dynamic routing ensures that different views of the application correspond to various routes, such as the main task list and individual task details.

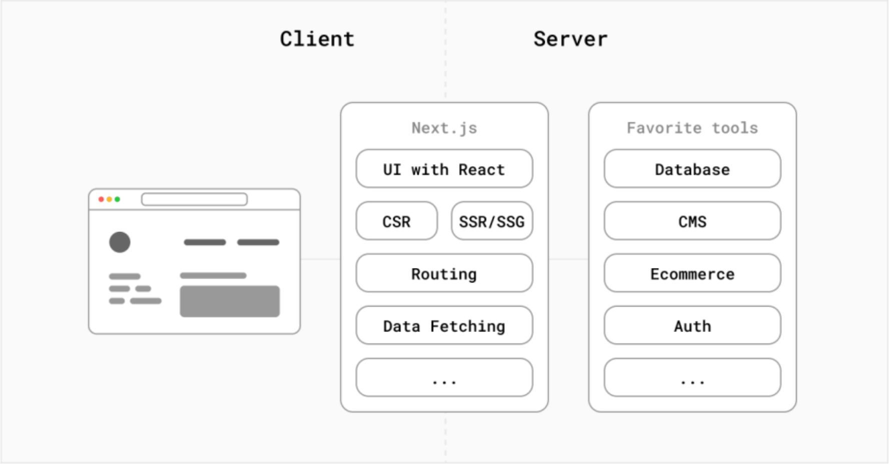


Figure 3. 4 NextJS server & client side rendering (REF:https://nextjs.org/)

3) React Components: The application's user interface is constructed using React components. Components are modular, reusable building blocks that encapsulate specific functionalities, such as task cards, columns, and buttons. By breaking down the user interface into components, the application benefits from improved code organization and reusability.

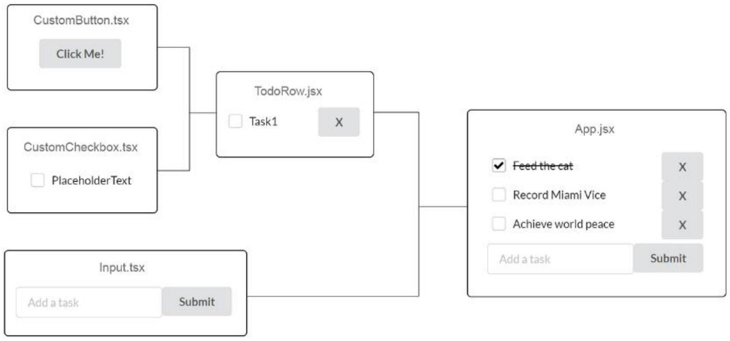


Figure 3. 5 React Component Architecture

4) JSON Server: For data persistence, a JSON server is employed. The JSON server acts as a simple backend that allows the application to perform CRUD (Create, Read, Update, Delete) operations on task data. Task data is stored in JSON format, facilitating easy data retrieval and storage.

c) Functionality Implementation:

The Kanban board-inspired functionality is seamlessly integrated within the application using the following strategies:

1) **Adding Tasks**: A user can add a new task by providing task details in an input form. Upon submission, the task is dynamically rendered as a card within the designated column, representing its current status. React state management ensures the real-time update of the UI.
2) **Editing Tasks**: Tasks can be edited by clicking on the task card. This action triggers the rendering of an editable form populated with the task's existing details. Once the edits are submitted, the task card's content is updated, maintaining the fluidity of the user experience.
3) **Deleting Tasks**: Users can remove tasks by interacting with a delete button associated with each task card. Clicking the button triggers the removal of the respective task card, ensuring a seamless removal process.

d) User Experience and Performance:

The architectural choices and integration strategies employed prioritize user experience and performance:

1) The usage of NextJS's server-side rendering ensures swift page loading and initial rendering of task data.
2) JSON server's lightweight backend facilitates efficient data manipulation and storage.
3) React's virtual DOM and component-based architecture enhance UI rendering and responsiveness.

By combining these architectural elements, the TodoList application achieves the goals of interactive Kanban board functionality, fluid user interactions, and seamless data management. The collaborative harmony between NextJS, React, and a JSON server results in an efficient and user-centric TodoList application.

6. **User Experience and Responsiveness**

The research also delves into user experience considerations and responsiveness. Learning from Trello and ClickUp's approach to responsive design ensures that the Kanban board remains accessible and functional across various devices. The research on user experience considerations and responsiveness also includes the integration of Tailwind CSS into the application's design. Inspired by the responsive design principles observed in Trello and ClickUp, Tailwind CSS offers a dynamic approach to user interface design, enabling rapid creation of visually appealing and responsive components. Its utility-first approach expedites UI development by reducing the need for custom styling and minimizing CSS code. The extensive array of pre-designed variants enhances UI flexibility, allowing developers to adjust component styles based on screen sizes and device types. This ensures a consistent and functional layout across various devices. The streamlined design process, combined with optimized performance, aligns well with the goal of delivering a seamless and engaging user experience.

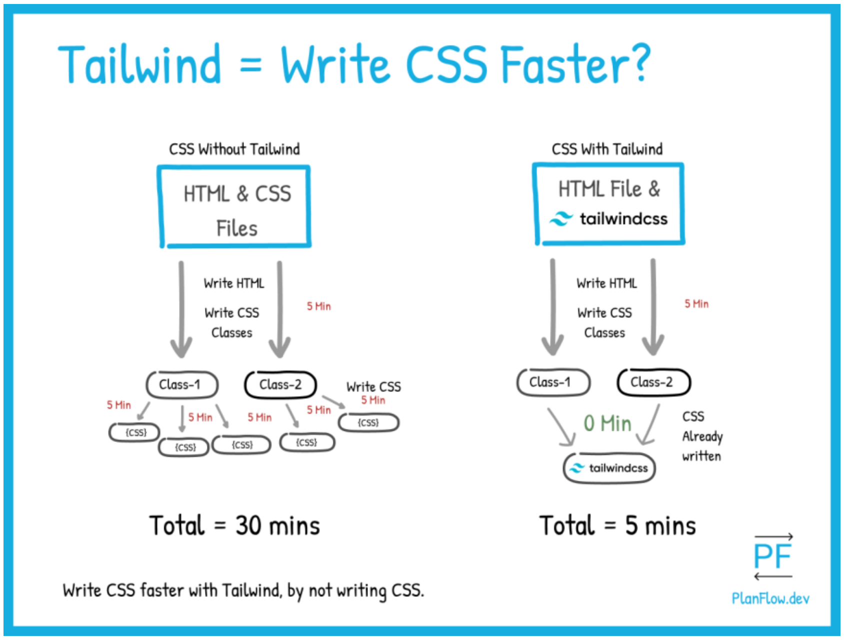
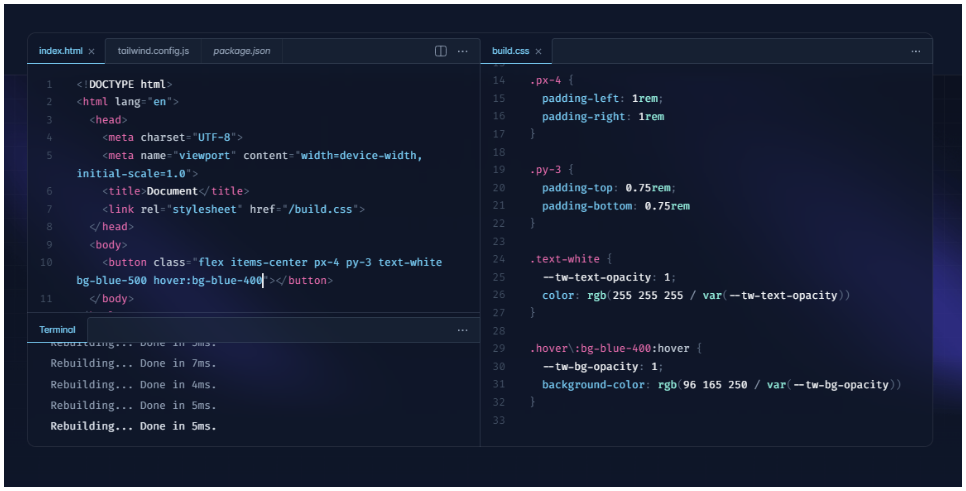


Figure 3. 6 Tailwind CSS comparison with CSS (REF: https://planflow.dev/blog/the-main-advantage-of-tailwindcss)



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="/build.css">
  </head>
  <body>
    <button class="flex items-center px-4 py-3 text-white
bg-blue-500 hover:bg-blue-400"></button>
  </body>
```

```
Rebuilding... Done in 7ms.
Rebuilding... Done in 4ms.
Rebuilding... Done in 5ms.
Rebuilding... Done in 5ms.
```

```css
.px-4 {
  padding-left: 1rem;
  padding-right: 1rem
}

.py-3 {
  padding-top: 0.75rem;
  padding-bottom: 0.75rem
}

.text-white {
  --tw-text-opacity: 1;
  color: rgb(255 255 255 / var(--tw-text-opacity))
}

.hover\:bg-blue-400:hover {
  --tw-bg-opacity: 1;
  background-color: rgb(96 165 250 / var(--tw-bg-opacity))
}

```

Figure 3. 7 Tailwind CSS and CSS Simplicity Comparison (REF: https://tailwindcss.com/)

The culmination of this research equips the development process with valuable insights from successful Kanban board implementations in applications like Trello and ClickUp. By adapting these principles within the NextJS framework, the goal is to create an intuitive, efficient, and visually engaging Kanban-inspired TodoList application.

3.1.2.2 Database Design

In the realm of database design, the strategic implementation of JSON Server takes center stage for our TodoList application. JSON Server, renowned for its efficiency and simplicity, seamlessly aligns with the project's objectives of rapid development and optimal user experience.

At the core of JSON Server's appeal lies its lightweight and agile nature. Utilizing a JSON-based data format, JSON Server facilitates swift storage and retrieval of task-related information. This inherent efficiency ensures rapid responses to user interactions, maintaining the application's responsiveness.

The decision to employ JSON Server goes beyond expediting development; it also underscores streamlined data management. JSON's innate structure enables the organization of task data within arrays and objects, simplifying the representation of tasks, statuses, and relevant metadata. This streamlined data structure facilitates efficient data fetching and updates, further enhancing the application's overall performance.

JSON Server's ease of integration is equally compelling. By creating a JSON file to serve as the data store, developers can effortlessly define API endpoints for various operations such as fetching, adding, updating, and deleting tasks. This straightforward setup accelerates development, allowing the team to focus on refining application features.

The real-time update capabilities of JSON Server seamlessly align with the dynamic nature of our TodoList application. As users interact by adding, editing, or removing tasks, JSON Server ensures instant synchronization of data across clients. This real-time synchronization contributes to a seamless collaborative experience, making it particularly valuable in scenarios where multiple users engage with the application concurrently.

In summary, the strategic utilization of JSON Server as the database solution fortifies the foundation of our TodoList application. Its lightweight structure, streamlined data handling, ease of integration, and real-time synchronization collectively contribute to a robust backend infrastructure. This, in turn, enhances the overall architecture, ensuring optimal performance and a superior user experience.

3.1.2.3 Website Design

Derived from insights obtained in the "Researching Reference" phase, the "Website Design" segment embodies the amalgamation of practical principles from platforms such as Trello and ClickUp, tailored to our project's scope. This section elucidates the meticulous crafting of the user interface (UI) and user experience (UX) to deliver a streamlined, intuitive, and effective environment for task management.

Drawing inspiration from the user-friendly interfaces of Trello and ClickUp, the UI/UX design of our TodoList application champions clarity and ease of use. By adopting a clean and uncluttered design approach, users can seamlessly navigate, interact with, and manage their tasks without any distractions.

Furthermore, our design adheres to responsive principles similar to those of Trello and ClickUp, ensuring a consistent and adaptable experience across various devices and screen sizes. Elements such as task cards, action buttons, and the overall layout dynamically adjust to retain optimal functionality and visibility. This adaptability guarantees that users can efficiently interact with the application regardless of the device they're using, be it a desktop or a mobile device.

Incorporating insights from the streamlined interaction patterns observed in Trello and ClickUp, our design prioritizes user-friendly actions such as adding, editing, and deleting tasks. These actions are seamlessly integrated into the interface, minimizing user effort and enhancing overall efficiency. The goal is to create an intuitive interaction model that users can easily embrace, contributing to a seamless and satisfying user experience.

In conclusion, the "Website Design" phase seamlessly weaves principles from Trello, ClickUp, and our project's specific vision. By embedding UI/UX principles that emphasize clarity, responsiveness, and streamlined interactions, our application endeavors to provide users with a straightforward and effective environment for task management.

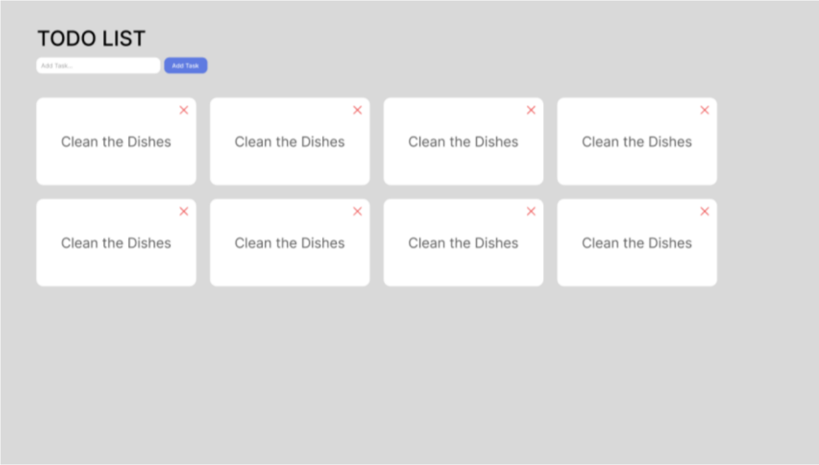


Figure 3. 8 Todo List App UI Plans

3.1.2.4 Implementation

The culmination of meticulous research, thoughtful design, and strategic choices leads the way to the "Implementation" phase. In this stage, I embark on transforming the vision into a functional reality. This section unveils the steps taken to bring the TodoList application to life, guided by the insights gained from prior sections and fueled by the desire to create a seamless task management experience.

At the heart of the implementation lies a well-defined architectural blueprint. The flowchart architecture, depicted in the attached diagram, outlines the interplay of components that power the TodoList application. This blueprint provides a roadmap for the integration of user interface (UI) elements, data storage, and the underlying functionality.

The integration of various components forms the backbone of the implementation strategy. The user interface, inspired by the user-friendly designs of platforms like Trello and ClickUp, is carefully constructed to offer an uncluttered and intuitive task management experience. Buttons for adding, editing, and deleting tasks are seamlessly integrated, aligning with the streamlined interaction patterns observed in the research.

JSON Server, the chosen backend solution, plays a pivotal role in managing task data efficiently. Drawing from the simplicity and speed of JSON-based storage, this component ensures prompt retrieval and manipulation of tasks. The JSON Server architecture, influenced by its real-time update capabilities, enables instant synchronization of task data across users.

Building on the responsive design principles inspired by platforms like Trello and ClickUp, the UI is designed to seamlessly adapt across devices. CSS elements, particularly from Tailwind CSS, are harnessed to ensure that task cards, columns, and buttons elegantly adjust to varying screen sizes. This meticulous responsiveness guarantees an optimal user experience on desktops, tablets, and mobile devices.

Integral to the implementation is rigorous testing. The application undergoes thorough testing to identify and address potential issues, ensuring its functionality aligns with the initial vision. This iterative process fosters the refinement of features, responsiveness, and overall usability, culminating in a polished and user-friendly TodoList application.

In essence, the "Implementation" phase marks the conversion of design and research into a tangible product. By employing a carefully crafted architectural blueprint, integrating user-centric components, leveraging JSON Server for efficient data management, implementing responsive design principles, and subjecting the application to rigorous testing, the TodoList app transitions from concept to reality. This journey epitomizes the harmonious convergence of research, design, and technical execution, driving closer to delivering a seamless task management experience to users.

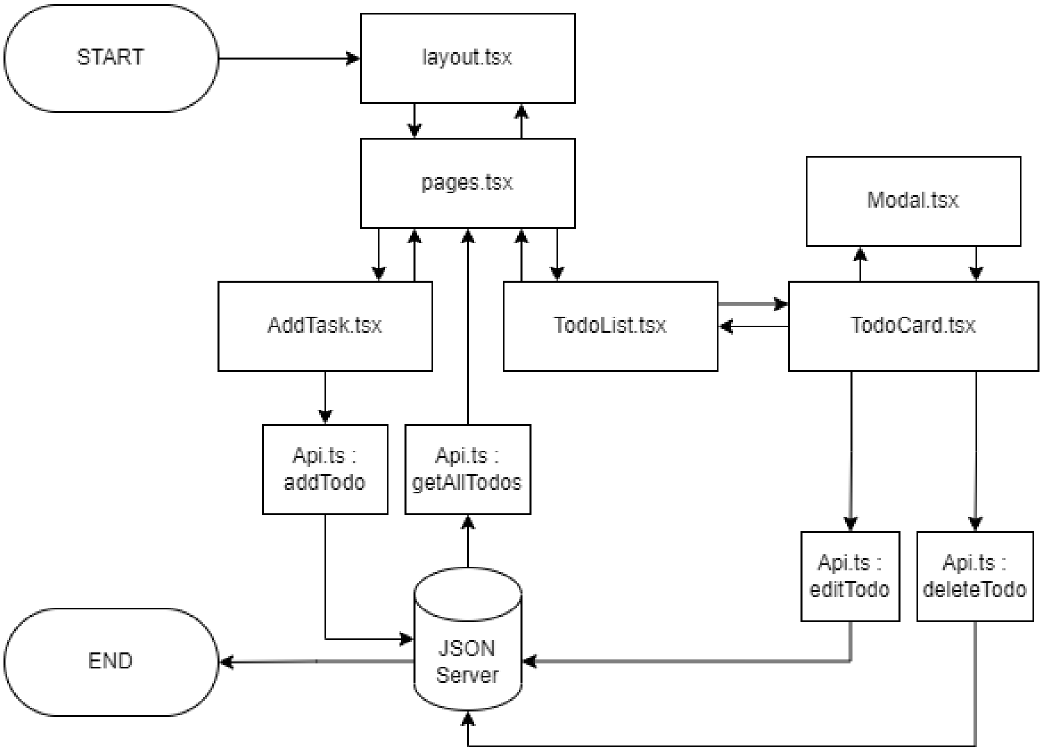


Figure 3. 9 Flow Diagram Plans for Todo List App

3.1.3 Used Software

3.1.3.1 Visual Studio Code

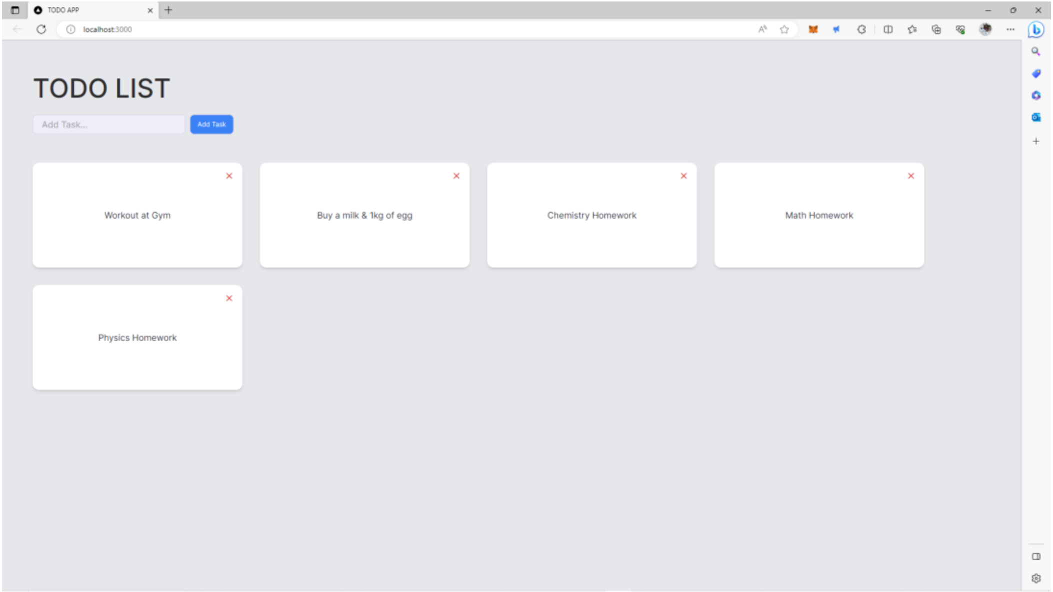
Visual Studio Code is a source code editor developed by Microsoft that can be used on various platforms (Linux, Mac OS and Microsoft). Visual Studio Code can also be used as a source code editor for various programming languages from PHP, Java, Python, Javascript, Node.js, C/C++, Go, C# to editors for web scripts such as HTML and CSS.

3.1.4 Project Result

As writers traverse through the culmination of our research, design, and implementation endeavors, the "Project Result" section emerges as a testament to our dedication and collaborative spirit. Here, writers unveil the fruition of our collective efforts, showcasing the tangible outcome of our journey to create a seamless and efficient TodoList application. This segment encapsulates the final product, presenting the user-centric features, functionalities, and overall experience that have been meticulously crafted and refined throughout the course of our project. The writers also include the whole code inside the following link: https://github.com/Qyuzet/Web2-TodoApp.git

3.1.4.1 Dashboard

The dashboard interface encapsulates the essence of user-centric design, drawing inspiration from industry pioneers like Trello and ClickUp. At the core of the dashboard lies the user's ability to effortlessly add, edit, and delete tasks, all while embracing a visually appealing and clutter-free environment.



Workout at Gym

Buy a milk & 1kg of egg

Chemistry Homework

Math Homework

Physics Homework

Figure 3. 10 Todo List App result dashboard

a) Add Task Bar

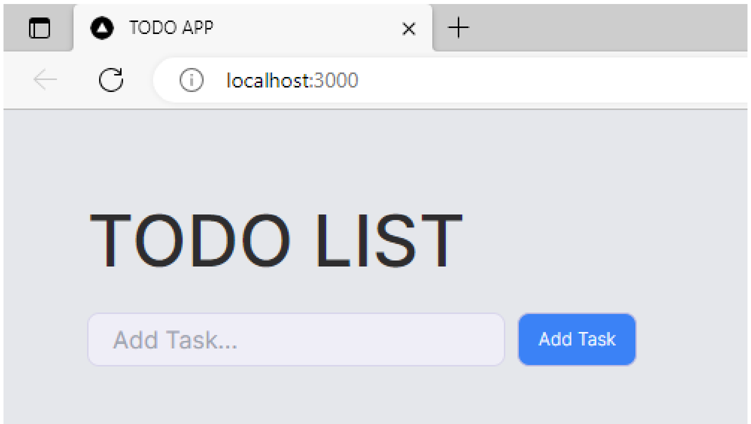


Figure 3. 11 Add Task Bar feature

Front and center on the dashboard, the "Add Task" bar beckons users to swiftly input their tasks. Inspired by the simplicity and accessibility of Trello's task creation, this feature aims to streamline the process. Upon entering task details, users are presented with the option to categorize tasks, set due dates, and assign labels, ensuring tasks are organized and easily identifiable.

b) Task Cards:

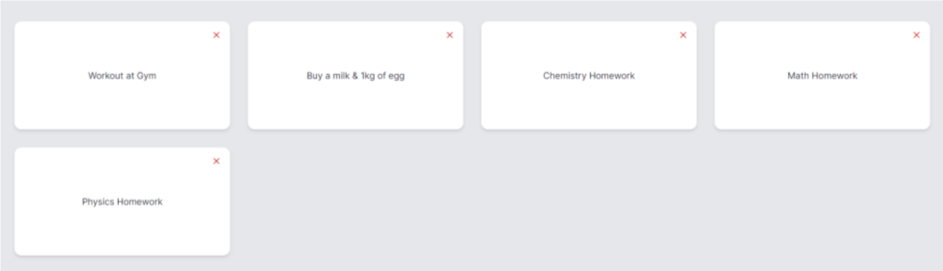
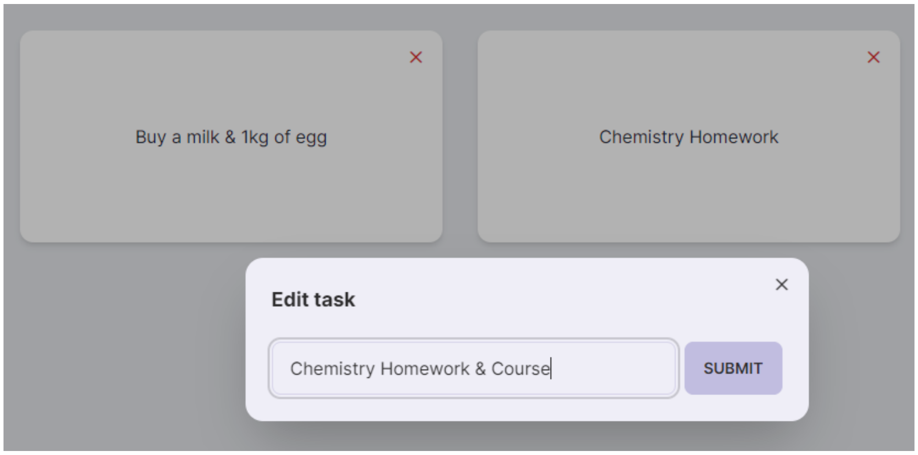


Figure 3. 12 Task card feature

The heartbeat of the dashboard, the task cards mirror Trello's card concept, enriched with our unique touch. Each task materializes as a distinct card, vividly displaying task titles, descriptions, due dates, and labels. This design echoes ClickUp's focus on detail and empowers users with comprehensive task information at a glance.

c) Edit Functionality:



Chemistry Homework & Course

Figure 3. 13 Edit functionality feature

Clicking on a task card opens a realm of possibilities with the integrated "Edit" function. Just as Trello empowers users to modify task attributes, our application enables seamless edits to task titles, descriptions, due dates, and labels. This fluidity echoes our commitment to enhancing user control and adaptability.

d) Delete Functionality:

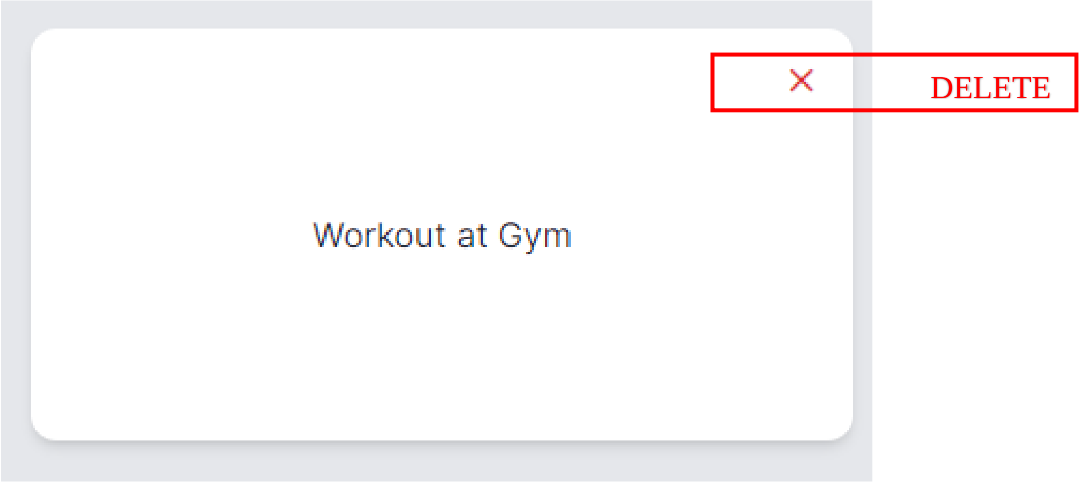


Figure 3. 14 Delete functionality feature

Embracing the efficiency of Trello's card removal, the "Delete" function empowers users to declutter their task list. With a simple click, users can eliminate tasks that have been completed or are no longer relevant, aligning with Trello's philosophy of maintaining a clean task environment.

3.1.4.2 Advanced Case Reference

This section unveils a curated selection of exceptional projects that exemplify NextJS's capacity to create captivating, interactive, and visually striking websites. Within these projects, writers witness NextJS's ability to seamlessly blend innovation with user engagement, offering a glimpse into the diverse applications of this powerful web development framework. Through the exploration of these real-world examples, writers gain insights into how NextJS transforms e-commerce experiences, elevates interactive media platforms, enhances educational landscapes, and amplifies corporate online presence. Each project serves as a testament to NextJS's role in shaping the future of web development, fostering seamless, dynamic, and engaging online interactions across a multitude of industries.

A. Pricing Calculator Website

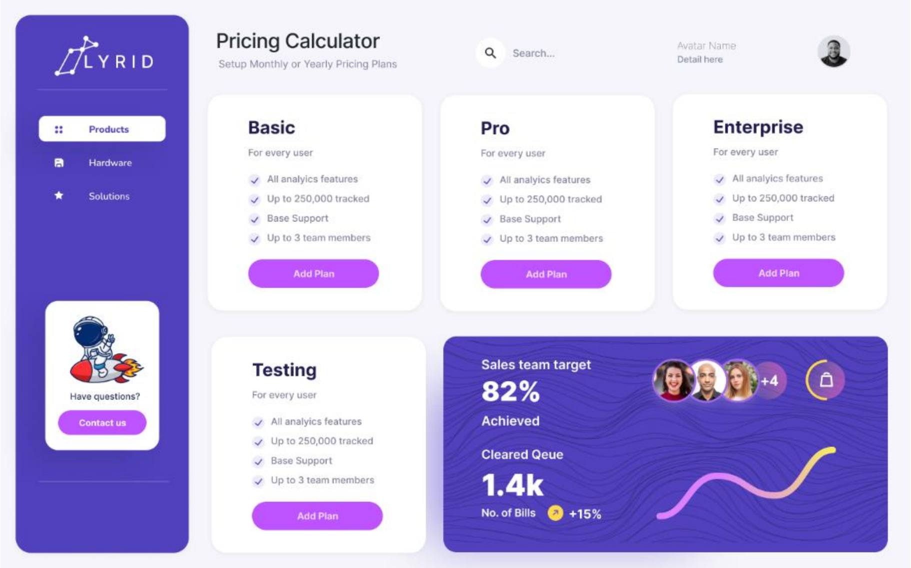


Figure 3. 15 Pricing calculator website

B. Initial Coin Offering Website

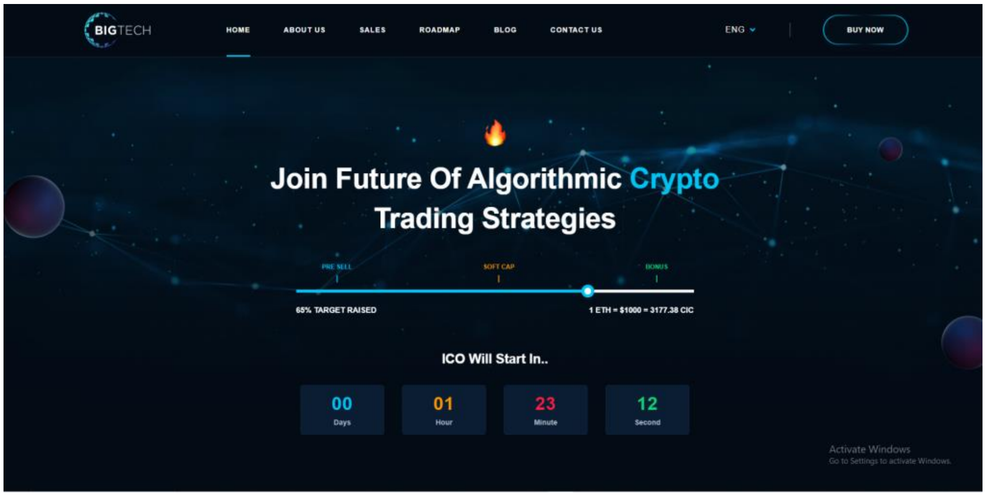
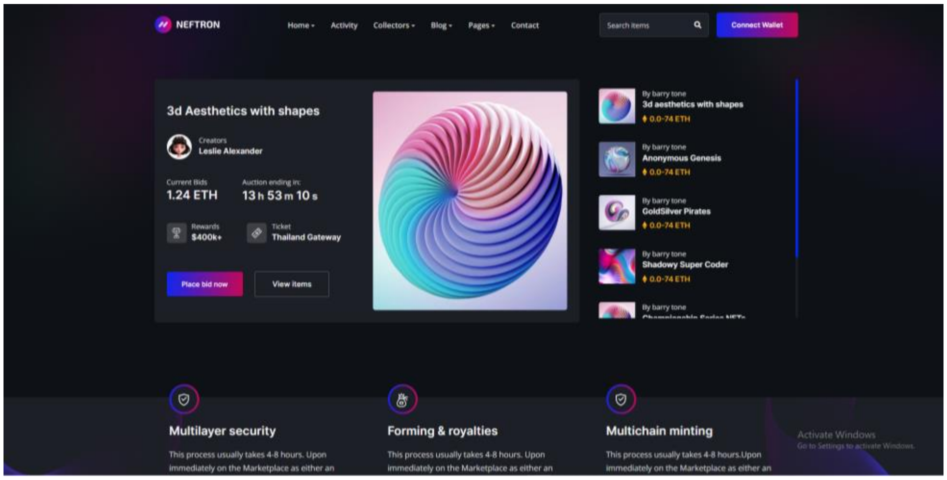


Figure 3. 16 Initial Coin Offering website

C. NFT & Crypto Marketplace



This process usually takes 4-8 hours. Upon immediately on the Marketplace as either an

This process usually takes 4-8 hours. Upon immediately on the Marketplace as either an

This process usually takes 4-8 hours.Upon immediately on the Marketplace as either an

Figure 3. 17 NFT & crypto marketplace

3.2 Exploring ASP.NET

Embarking on a journey through the intricate landscape of web development, writers now turn our attention to the formidable framework known as ASP.NET. Within this section, writers delve into the depths of ASP.NET, peeling back its layers to reveal a world of possibilities for crafting web applications that are not only robust and efficient but also dynamically responsive. As writers navigate through this exploration, our aim is to meticulously dissect the features, advantages, and strategic applications that position ASP.NET as a preeminent choice for the development of comprehensive and high-performing web solutions. Through the lens of ASP.NET, writers delve into the nuanced realm of web development, shedding illuminating insights on how this framework enriches the process of web application creation, catering to the distinct needs of both expansive enterprises and the rapidly evolving digital landscape.

3.2.1 What is ASP.NET

As the writer steps into the world of ASP.NET, the journey ahead unveils the layers of this dynamic framework. This section serves as a preliminary guide, shedding light on the significance and versatility of ASP.NET within the realm of web development. By delving into the intricacies of ASP.NET, this segment provides a foundational understanding, setting the stage for a comprehensive exploration of its features and advantages. Through this initial perspective, the groundwork is laid for a profound comprehension of how ASP.NET shapes the landscape of modern web application development.

3.2.1.1 ASP.NET Definition & Variant

Within this sub-subsection, the discourse unfurls around the intricate domain of the ASP.NET framework, illuminating its underlying architectural intricacies and distinct variant forms. ASP.NET, a revered open-source web development framework, stands as a testament to innovation and adaptability in the realm of web application creation. Its purpose is to craft robust, dynamic, and efficient web applications, presenting a wide spectrum of possibilities for developers to manifest their creative visions.

ASP.NET manifests in two distinct variants, each meticulously fashioned to address particular development paradigms and architectural requisites. The first, ASP.NET Web Forms, encapsulates a component-based approach that empowers developers to meticulously construct visually captivating user interfaces, harnessing an array of reusable controls. This variant excels in expediting application development, simplifying complex tasks, and facilitating a streamlined developmental trajectory. On the other end of the spectrum, ASP.NET MVC (Model-View-Controller) adheres to the architectural pattern from which its nomenclature derives. This variant delineates a meticulously structured division of roles within the application, fostering a methodical approach to development. ASP.NET MVC is particularly tailored for projects that demand precision control over components and their interactions.

ASP.NET's pivotal potency resides in its ability to adapt to the distinct requisites of developers and the projects they undertake. ASP.NET Web Forms exemplifies innate simplicity, accelerating development, while ASP.NET MVC champions a meticulously separated and maintainable codebase. As writers delve further into this section, the discourse unveils the architectural intricacies of these variants, furnishing an encompassing comprehension of their distinct attributes, use cases, and contributions to the ever-evolving landscape of web development.

3.2.1.2 Advantages and Disadvantagese

In the discourse that follows, writer undertake a comprehensive evaluation of ASP.NET, dissecting both its commendable advantages and inherent drawbacks. As a versatile web development framework, ASP.NET offers a pantheon of merits that contribute to its prominence in the development community. Its object-oriented nature, seamless integration with Windows systems, and the vast ecosystem of libraries and tools elevate its standing as a preferred choice for crafting robust web applications.

A. Advantages

1) **Robust Integratio**n: ASP.NET seamlessly integrates with various Microsoft technologies, enabling developers to create applications that effortlessly interact with other Microsoft tools and systems.

2) **Security Measures**: ASP.NET provides robust security features, including form authentication and role-based access control, enhancing the protection of web applications and sensitive data.

3) **Modular Architecture**: The modular nature of ASP.NET enhances maintainability, allowing developers to work on specific components without disrupting the entire application, thereby streamlining development workflows.

4) **Vast Ecosystem**: The extensive collection of libraries, frameworks, and tools associated with ASP.NET empowers developers with a rich resource pool to expedite development and enhance functionality.

5) **Object-Oriented Approach**: ASP.NET's object-oriented paradigm facilitates cleaner and more organized code, contributing to improved code readability and maintainability.

B. **Disadvantages**

1) **Learning Curve**: Mastering ASP.NET might require a steeper learning curve, particularly for developers new to the Microsoft ecosystem, which can lead to a longer onboarding process.

2) **Proprietary Nature**: While ASP.NET is open-source, it is closely associated with Microsoft's proprietary technologies, potentially limiting the flexibility and freedom of developers who prefer open-source alternatives.

3) **Performance Variation**: Although ASP.NET boasts impressive performance metrics, such as handling up to 7 million requests per second, this efficiency is largely optimized for Windows environments. Performance may vary when applied to non-Windows platforms.

4) **Resource Consumption**: ASP.NET applications may consume more resources than lightweight alternatives, potentially impacting scalability and cost considerations.

3.2.1.3 The Company who Implemented ASP.NET

Within this sub-subsection, writer delve into the realm of industry giants that have harnessed the power of ASP.NET to bolster their digital presence and streamline their operations. The utilization of ASP.NET by these renowned entities underscores its significance as a formidable framework in the world of web development. Here are ten prominent companies that have chosen ASP.NET to power their digital initiatives:

1. **Microsoft**: As the progenitor of ASP.NET, Microsoft employs its brainchild to fuel its own web applications, reinforcing the framework's robustness and adaptability.
2. **Stack Overflow**: The popular online community for programmers and developers, Stack Overflow, utilizes ASP.NET to maintain a responsive and interactive platform for knowledge sharing and discussions.
3. **Dell**: The multinational computer technology corporation, Dell, leverages ASP.NET to power its online store and facilitate seamless customer interactions.
4. **GoDaddy**: A leading domain registrar and web hosting company, GoDaddy, harnesses ASP.NET to offer users a user-friendly interface for domain registration and web hosting management.

5. **Verizon**: The telecommunications giant, Verizon, employs ASP.NET to deliver a streamlined and engaging user experience across its web-based customer services and products.
6. **Siemens**: The global conglomerate, Siemens, utilizes ASP.NET to develop sophisticated web applications for its diverse portfolio of products and services.
7. **The Home Depot**: A retail behemoth, The Home Depot, embraces ASP.NET to maintain an efficient online store that caters to the diverse needs of its customers.
8. **General Electric (GE**): The multinational conglomerate, GE, harnesses ASP.NET to create web applications that support its wide array of industries, from aviation to healthcare.
9. **Intuit**: The financial software company, Intuit, employs ASP.NET to offer users intuitive and feature-rich applications for managing their finances and businesses.
10. **Walmart**: The retail giant, Walmart, leverages ASP.NET to ensure a seamless online shopping experience for its millions of customers, showcasing the framework's scalability and performance.

These prominent entities stand as a testament to ASP.NET's adaptability and prowess in catering to the diverse needs of large-scale corporations. Through their utilization of ASP.NET, they underscore the framework's capability to power sophisticated web applications that align with their business goals and user expectations.

3.2.1.4 Vast Implementation of ASP.NET

The realm of ASP.NET extends far beyond the boundaries of traditional web development, encompassing an array of applications that span industries and technological domains. This sub-subsection delves into the diverse and expansive implementations of ASP.NET, showcasing its adaptability and versatility across various sectors.

1. **Enterprise Resource Planning (ERP) Systems**: ASP.NET finds extensive use in developing ERP systems that streamline business processes, integrate data, and enhance operational efficiency. Its robust architecture and scalability make it an ideal choice for managing complex business operations.
2. **E-Commerce Platforms**: Beyond standard online stores, ASP.NET powers intricate e-commerce platforms that encompass diverse features like inventory management, order processing, and customer relationship management, offering a comprehensive solution for businesses of all sizes.
3. **Healthcare Management**: ASP.NET facilitates the creation of robust healthcare management systems that centralize patient information, appointment scheduling, and medical record management, contributing to more efficient healthcare services.
4. **Financial Services**: ASP.NET is employed to develop secure and efficient financial applications, including online banking portals, investment platforms, and trading systems, catering to the stringent security and performance requirements of the financial sector.
5. **Education and E-Learning**: ASP.NET supports the development of e-learning platforms, allowing educational institutions to offer online courses, manage student progress, and provide interactive learning experiences.
6. **Robotics and Automation**: ASP.NET's capabilities extend to robotics and automation, powering the interfaces of robotic systems used in manufacturing, logistics, and research.
7. **Gaming and Entertainment**: The gaming industry leverages ASP.NET for building online gaming platforms, virtual reality experiences, and

interactive entertainment applications that engage users across various platforms.

8. **Internet of Things (IoT)**: ASP.NET is harnessed to create IoT applications that facilitate communication and data exchange between connected devices, enabling real-time monitoring, control, and automation.
9. **Supply Chain Management**: ASP.NET supports the development of supply chain management systems that optimize logistics, inventory, and distribution processes, enhancing overall operational efficiency.
10. **Government and Public Services**: ASP.NET is utilized to build secure and user-friendly government portals, digital citizen services, and administrative systems, promoting transparent and efficient interactions between citizens and government agencies.
11. **Aerospace and Defense**: The aerospace and defense industries benefit from ASP.NET's capabilities in developing applications for flight simulation, mission planning, and data analysis.
12. **Energy and Utilities**: ASP.NET contributes to applications that monitor and control energy and utility networks, enhancing resource utilization and reliability.
13. **Telecommunications**: ASP.NET supports the creation of telecom management systems that enable efficient network monitoring, billing, and customer management.
14. **Automotive**: ASP.NET is adopted in developing applications for automotive diagnostics, vehicle management, and connected car solutions.
15. **Environmental Monitoring**: ASP.NET can be used to develop applications that monitor environmental data and contribute to environmental conservation efforts.

The breadth of ASP.NET's implementation highlights its flexibility in catering to a wide range of industries and technological domains. Its robustness, scalability, and adaptability make it an invaluable tool in creating diverse applications that address specific business needs and technological advancements.

3.2.2 ASP.NET Abstraction Programming Technique Overview

Abstraction programming techniques form the bedrock of ASP.NET's development paradigm, enabling the creation of modular, maintainable, and adaptable applications. In this section, writers delve into a diverse array of abstraction methods that underpin the framework's architecture. These methods include delegates, events, generics, extension methods, lambda expressions, LINQ, and asynchronous programming.

Each technique serves as a building block, contributing to the overall structure, efficiency, and extensibility of ASP.NET applications. Through a deep exploration of these techniques, writers gain insight into how ASP.NET empowers developers to craft code that aligns with modern software development principles, fostering robustness and scalability. As writers navigate through each abstraction method, their unique roles in enhancing code organization and promoting flexible application design become increasingly apparent.

3.2.2.1 Abstraction Definition

Abstraction serves as a cornerstone of software engineering, offering a systematic approach to managing the intricate nature of complex systems. In essence, abstraction involves the process of simplifying the representation of real-world entities, highlighting essential attributes while suppressing less relevant details. By encapsulating intricate functionalities within higher-level constructs, abstraction empowers developers to work with a distilled version of reality that facilitates efficient design, implementation, and maintenance.

Within the context of software development, abstraction operates on multiple levels. At its core, it involves the creation of abstract data types, classes, or modules that encapsulate a specific set of functionalities. These abstractions serve as building blocks for constructing larger systems, allowing developers to focus on high-level design and logical flow, rather than grappling with the minutiae of every implementation detail.

Abstraction's paramount objective is to offer a clear separation between what an entity does and how it accomplishes it. This separation not only simplifies the development process but also fosters code reusability. By exposing only the essential methods, properties, and behaviors of an entity while hiding the internal intricacies, abstraction promotes a modular and maintainable codebase. Additionally, abstraction enables developers to create interfaces that act as standardized contracts, enabling different components to interact seamlessly without necessitating knowledge of each other's internal workings.

In the realm of ASP.NET development, abstraction plays a vital role in promoting the separation of concerns and enabling developers to tackle complex projects in a structured manner. By abstracting away the complexities of lower-level functionalities, ASP.NET empowers developers to focus on creating innovative solutions while ensuring a cohesive and efficient development process.

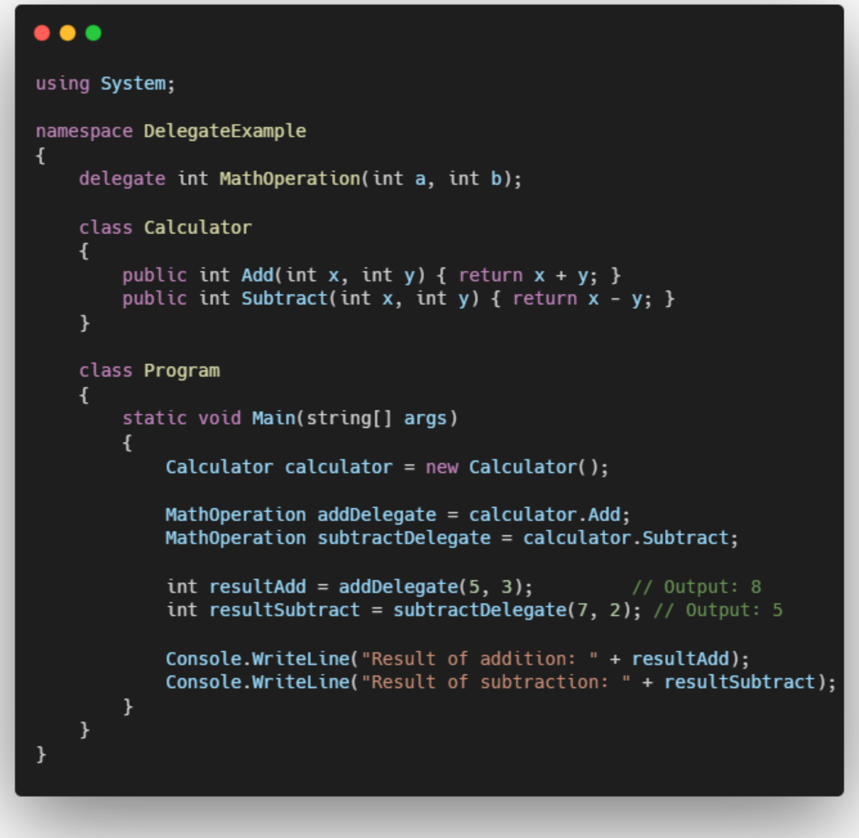
Throughout this section, our exploration of abstraction techniques will delve into the intricacies of delegates, events, generics, extension methods, lambda expressions, LINQ, and asynchronous programming. These techniques harness the concept of abstraction to create concise, reusable, and adaptable code structures, allowing developers to harness the full potential of ASP.NET's capabilities. By examining how each technique leverages abstraction, writers gain a deeper understanding of ASP.NET's versatility in addressing diverse development challenges while maintaining code clarity and modularity.

3.2.2.2 Delegates

Delegates stand as a cornerstone of abstraction within the ASP.NET framework, facilitating the implementation of dynamic behavior in a concise and organized manner. At its core, a delegate is a type that defines a contract for method signatures, allowing developers to treat functions as first-class entities. This abstraction enables the encapsulation of method calls within a delegate instance, allowing functions to be passed as parameters, returned from other functions, and stored as variables.

A delegate is essentially a reference to a method, allowing developers to invoke methods indirectly through the delegate instance. This proves particularly useful when constructing applications that require dynamic invocation of functions or when implementing design patterns such as the Observer pattern. Delegates empower developers to create flexible, modular code by decoupling the calling code from the specific method implementation.

Consider the following example to comprehend the application of delegates within ASP.NET:



```
using System;

namespace DelegateExample
{
    delegate int MathOperation(int a, int b);

    class Calculator
    {
        public int Add(int x, int y) { return x + y; }
        public int Subtract(int x, int y) { return x - y; }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Calculator calculator = new Calculator();

            MathOperation addDelegate = calculator.Add;
            MathOperation subtractDelegate = calculator.Subtract;

            int resultAdd = addDelegate(5, 3);         // Output: 8
            int resultSubtract = subtractDelegate(7, 2); // Output: 5

            Console.WriteLine("Result of addition: " + resultAdd);
            Console.WriteLine("Result of subtraction: " + resultSubtract);
        }
    }
}
```

Figure 3. 18 Delegates programming techniques

In this example, writers define a delegate **‘MathOperation’** that encapsulates methods with the same signature. The ‘**Calculator’** class contains methods for addition and subtraction. Through delegates, writers create instances that reference these methods and invoke them dynamically.

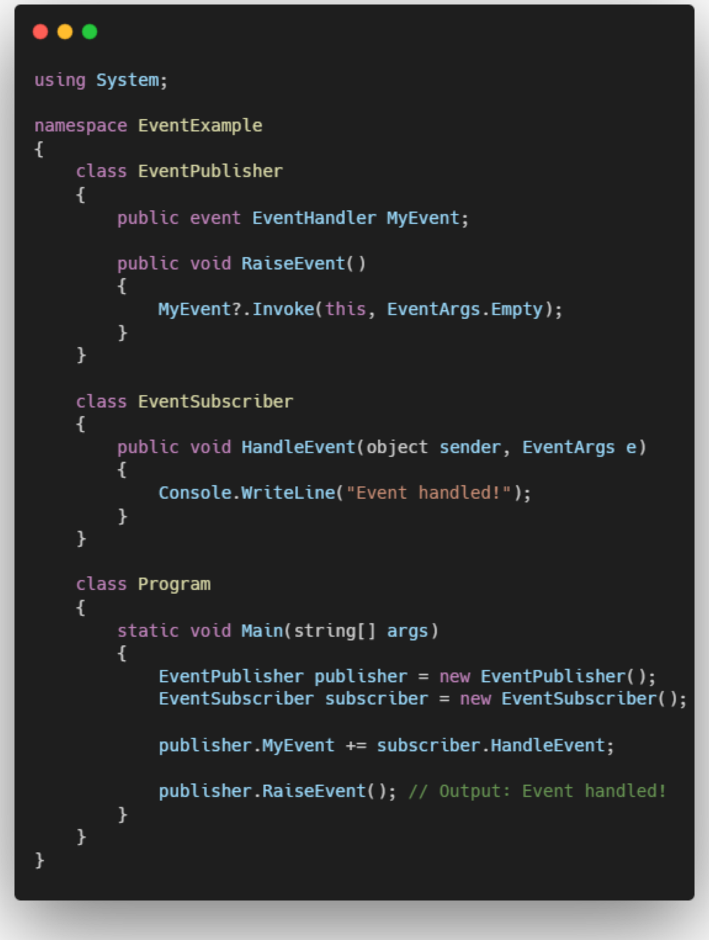
By leveraging delegates, ASP.NET developers can craft more flexible and adaptive code structures. Delegates play a pivotal role in events, callback mechanisms, and LINQ expressions, enhancing the extensibility and modularity of ASP.NET applications. This intricate abstraction technique exemplifies how ASP.NET empowers developers to build code that aligns with modern programming paradigms, offering both efficiency and maintainability.

3.2.2.3 Events

The concept of events in ASP.NET extends the realm of abstraction by providing a structured way to handle and notify changes, interactions, and state transitions within an application. An event is essentially a mechanism that enables an object to notify other objects when a specific action or state change occurs. This abstraction fosters decoupling between components, allowing them to interact without direct dependencies.

Events consist of two main entities: the event source and the event handler. The event source, typically an object, triggers the event when a specific condition is met. The event handler, on the other hand, is responsible for executing specific logic in response to the event. This separation of concerns promotes modular design, as objects can react to events without requiring a deep understanding of the event source's internals.

In ASP.NET, events are widely used to implement user interactions, data updates, and system notifications. They provide a mechanism for various components to communicate and respond to user actions or changes in data. For instance, consider a scenario where a button click in a web application triggers an event that updates a database record. Through event-driven programming, developers can encapsulate the logic for handling the button click and database update separately, resulting in a more maintainable and comprehensible codebase.



```
using System;

namespace EventExample
{
    class EventPublisher
    {
        public event EventHandler MyEvent;

        public void RaiseEvent()
        {
            MyEvent?.Invoke(this, EventArgs.Empty);
        }
    }

    class EventSubscriber
    {
        public void HandleEvent(object sender, EventArgs e)
        {
            Console.WriteLine("Event handled!");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            EventPublisher publisher = new EventPublisher();
            EventSubscriber subscriber = new EventSubscriber();

            publisher.MyEvent += subscriber.HandleEvent;

            publisher.RaiseEvent(); // Output: Event handled!
        }
    }
}
```

Figure 3. 19 Events programming techniques

In this example, an ‘**EventPublisher’** class defines an event named ‘**MyEvent**’. An ‘**EventSubscriber**’ class contains the event handler method ‘**HandleEvent**’. By subscribing to the event and raising it, the subscriber class reacts to the event by executing the handler method.

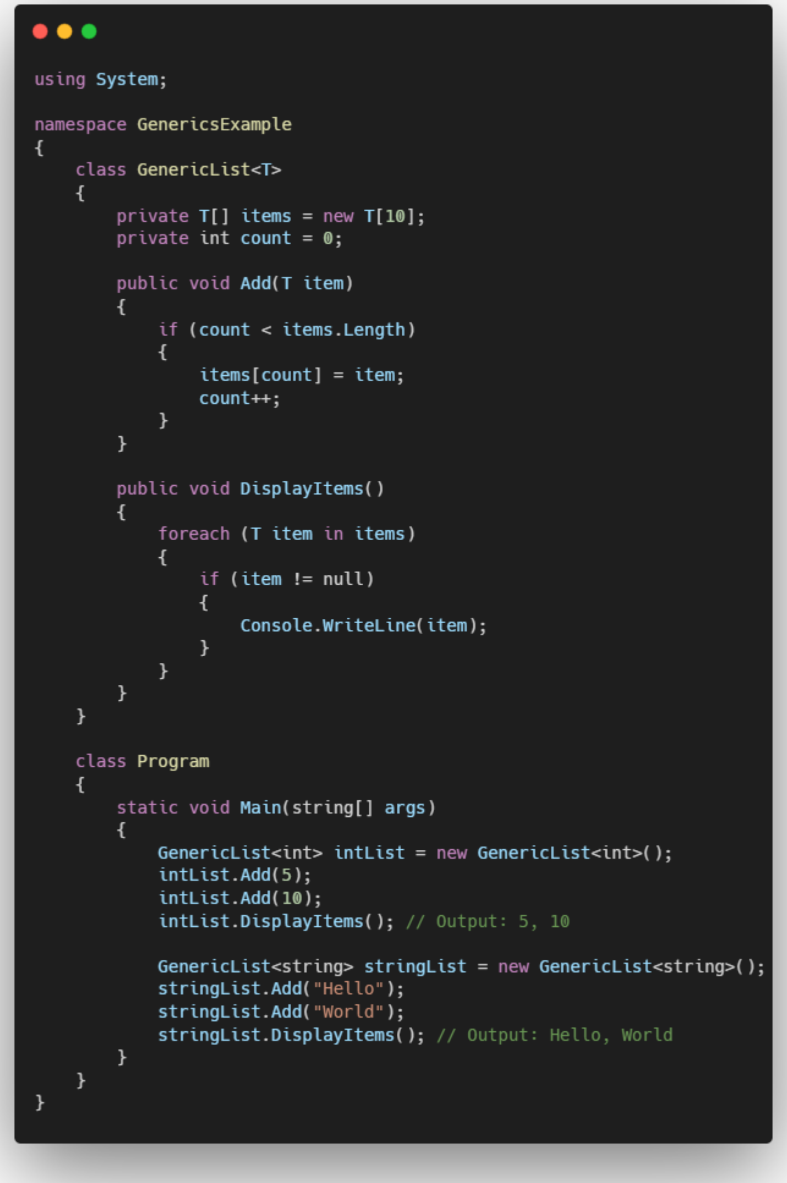
Events within ASP.NET exemplify the power of abstraction in promoting loose coupling and modular design. They enable developers to create responsive and interactive applications by establishing structured communication between components. This abstraction technique not only enhances the readability of code but also contributes to the overall scalability and maintainability of ASP.NET applications.

3.2.2.4 Generics

Within the realm of ASP.NET, the concept of generics introduces a versatile form of abstraction that significantly enhances code reusability and type safety. Generics empower developers to create classes, interfaces, methods, and delegates that can operate on various data types without sacrificing compile-time type checking.

At its core, generics enable the creation of components that can work with a wide range of data types while maintaining strong type safety. This abstraction technique is particularly valuable in scenarios where the same logic can be applied to different data types without resorting to casting or duplicated code.

Consider the following example to understand the utilization of generics in ASP.NET:



```
using System;

namespace GenericsExample
{
    class GenericList<T>
    {
        private T[] items = new T[10];
        private int count = 0;

        public void Add(T item)
        {
            if (count < items.Length)
            {
                items[count] = item;
                count++;
            }
        }

        public void DisplayItems()
        {
            foreach (T item in items)
            {
                if (item != null)
                {
                    Console.WriteLine(item);
                }
            }
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            GenericList<int> intList = new GenericList<int>();
            intList.Add(5);
            intList.Add(10);
            intList.DisplayItems(); // Output: 5, 10

            GenericList<string> stringList = new GenericList<string>();
            stringList.Add("Hello");
            stringList.Add("World");
            stringList.DisplayItems(); // Output: Hello, World
        }
    }
}
```

Figure 3. 20 Generics programming techniques

In this example, the ‘**GenericList’** class utilizes generics to create a container that can store various data types. The ‘**Add’** method is generic and can accept different types, ensuring type safety while offering code reusability.

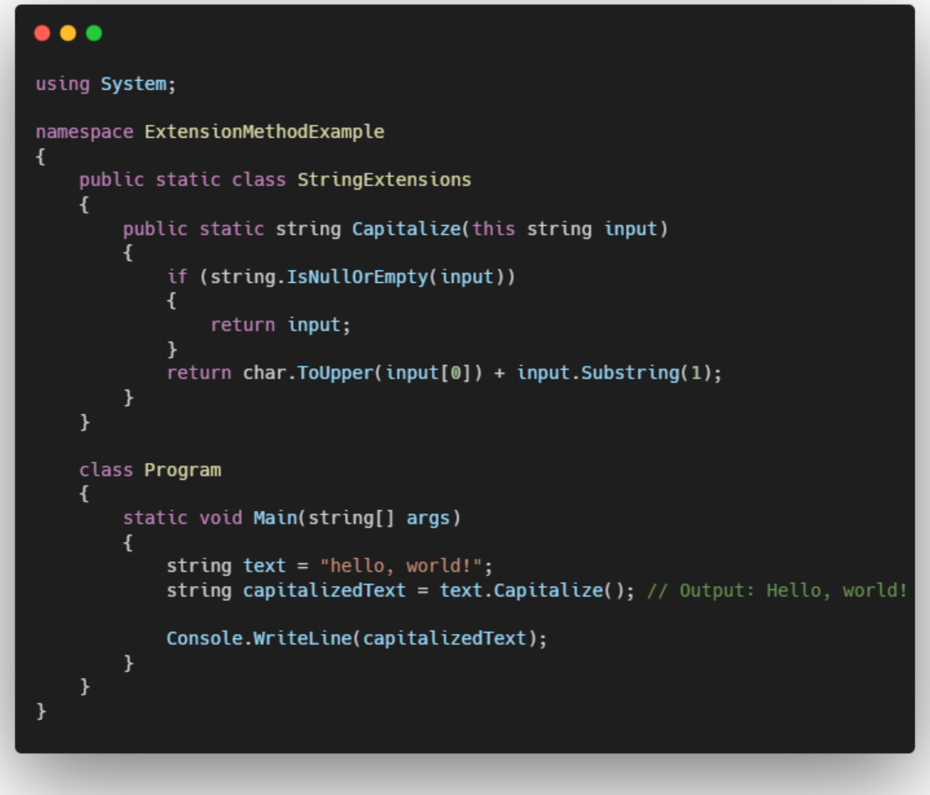
Generics within ASP.NET streamline code development by enabling the creation of versatile and flexible components that can accommodate a multitude of data types. Whether working with collections, database operations, or custom data structures, generics exemplify abstraction's role in simplifying complex tasks while maintaining a high level of type safety and code integrity. This abstraction technique within ASP.NET forms a fundamental building block for constructing efficient and adaptable applications.

3.2.2.5 Extensions Method

The utilization of extension methods within the ASP.NET framework introduces a form of abstraction that enables developers to enhance the functionality of existing classes without modifying their source code. Extension methods enable the addition of new methods to types that are already defined, including built-in and user-defined types. This abstraction empowers developers to augment existing classes with new behavior, improving code readability and promoting modular design.

Extension methods are static methods that are contained within static classes. These methods are decorated with the ‘**this’** keyword followed by the type being extended, allowing them to be invoked as if they were instance methods of that type. This abstraction fosters the illusion of native functionality while promoting the separation of concerns and preventing class modification.

Consider the following example to illustrate the implementation of extension methods within ASP.NET:



```
using System;

namespace ExtensionMethodExample
{
    public static class StringExtensions
    {
        public static string Capitalize(this string input)
        {
            if (string.IsNullOrEmpty(input))
            {
                return input;
            }
            return char.ToUpper(input[0]) + input.Substring(1);
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            string text = "hello, world!";
            string capitalizedText = text.Capitalize(); // Output: Hello, world!

            Console.WriteLine(capitalizedText);
        }
    }
}
```

Figure 3. 21 Extensions method programming techniques

In this example, the ‘**StringExtensions**’ class defines an extension method named ‘**Capitalize**’, which capitalizes the first letter of a string. By utilizing the ‘**this**’ keyword, the extension method can be invoked directly on a string instance.

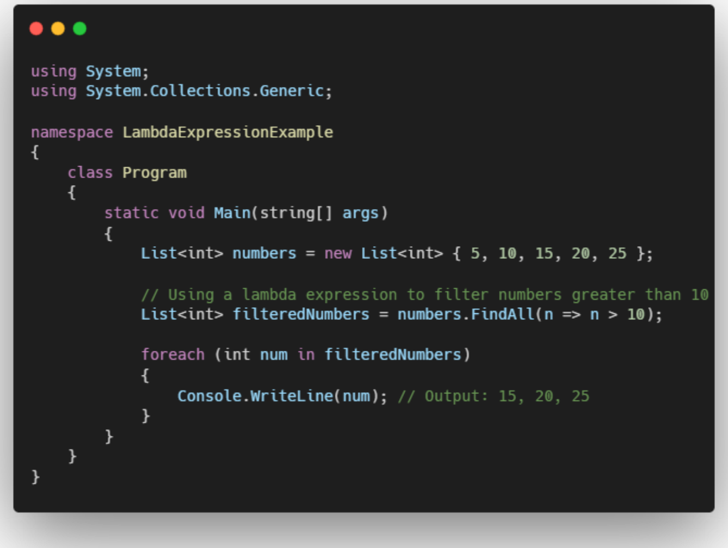
Extension methods within ASP.NET offer an elegant way to enrich the capabilities of existing types, making code more expressive and concise. This abstraction technique is particularly valuable when working with third-party libraries or framework classes, allowing developers to seamlessly integrate custom functionality without altering the original class structure. By employing extension methods, ASP.NET developers can elevate code reusability, readability, and maintainability, thereby showcasing the role of abstraction in fostering flexible and extensible applications.

3.2.2.6 Lambda Expressions

Lambda expressions within the ASP.NET framework introduce a concise and expressive form of abstraction that simplifies the implementation of anonymous methods or delegates. Lambda expressions enable developers to define small, inline functions without the need to explicitly declare a separate method or delegate. This abstraction technique streamlines code development by promoting a more functional programming style and enhancing readability.

Lambda expressions consist of a parameter list, a lambda operator **(‘=>’),** and a body. The body can be a single expression or a block of statements. Lambda expressions are particularly useful in scenarios where a simple function or delegate is needed, such as for sorting, filtering, or transforming data collections.

Consider the following example to illustrate the usage of lambda expressions within ASP.NET:



```
using System;
using System.Collections.Generic;

namespace LambdaExpressionExample
{
    class Program
    {
        static void Main(string[] args)
        {
            List<int> numbers = new List<int> { 5, 10, 15, 20, 25 };

            // Using a lambda expression to filter numbers greater than 10
            List<int> filteredNumbers = numbers.FindAll(n => n > 10);

            foreach (int num in filteredNumbers)
            {
                Console.WriteLine(num); // Output: 15, 20, 25
            }
        }
    }
}
```

Figure 3. 22 Lambda expressions programming techniques

In this example, the ‘**FindAll’** method uses a lambda expression to define a filter condition. The lambda expression ‘**n => n > 10**’ represents a function that checks whether a number is greater than 10. This concise notation enhances code readability and reduces the need for defining separate methods or delegates.

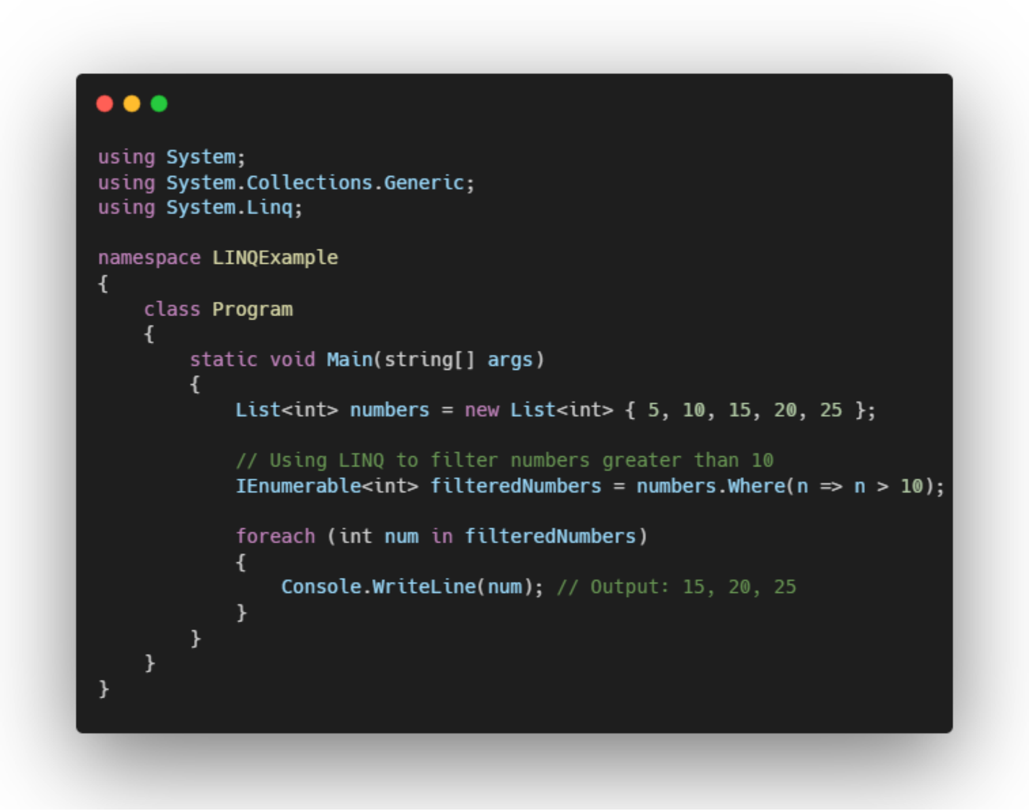
Lambda expressions within ASP.NET offer a succinct way to encapsulate simple operations, making code more compact and comprehensible. By embracing lambda expressions, developers can adopt a more functional programming approach, thereby promoting modular and efficient code development. This abstraction technique demonstrates ASP.NET's adaptability to modern programming paradigms and the role of abstraction in promoting efficient code structures.

3.2.2.7 LINQ

The Language Integrated Query (LINQ) within the ASP.NET framework introduces a powerful abstraction that enables developers to perform data querying and manipulation tasks using a unified syntax. LINQ provides a consistent querying approach regardless of the data source, such as collections, databases, XML, or other data formats. This abstraction simplifies the querying process and enhances code maintainability by eliminating the need for different query languages for different data sources.

LINQ allows developers to write queries directly in C# or other .NET languages, providing a familiar and intuitive way to retrieve, transform, and filter data. This abstraction technique leverages deferred execution, optimizing performance by executing queries only when results are actually needed.

Consider the following example to showcase the usage of LINQ within ASP.NET:



```
using System;
using System.Collections.Generic;
using System.Linq;

namespace LINQExample
{
    class Program
    {
        static void Main(string[] args)
        {
            List<int> numbers = new List<int> { 5, 10, 15, 20, 25 };

            // Using LINQ to filter numbers greater than 10
            IEnumerable<int> filteredNumbers = numbers.Where(n => n > 10);

            foreach (int num in filteredNumbers)
            {
                Console.WriteLine(num); // Output: 15, 20, 25
            }
        }
    }
}
```

Figure 3. 23 LINQ programming techniques

In this example, the '**Where**' method employs LINQ to filter numbers greater than 10 from the **'numbers'** collection. The LINQ query is expressed using lambda expressions, providing a concise and readable syntax for data manipulation.

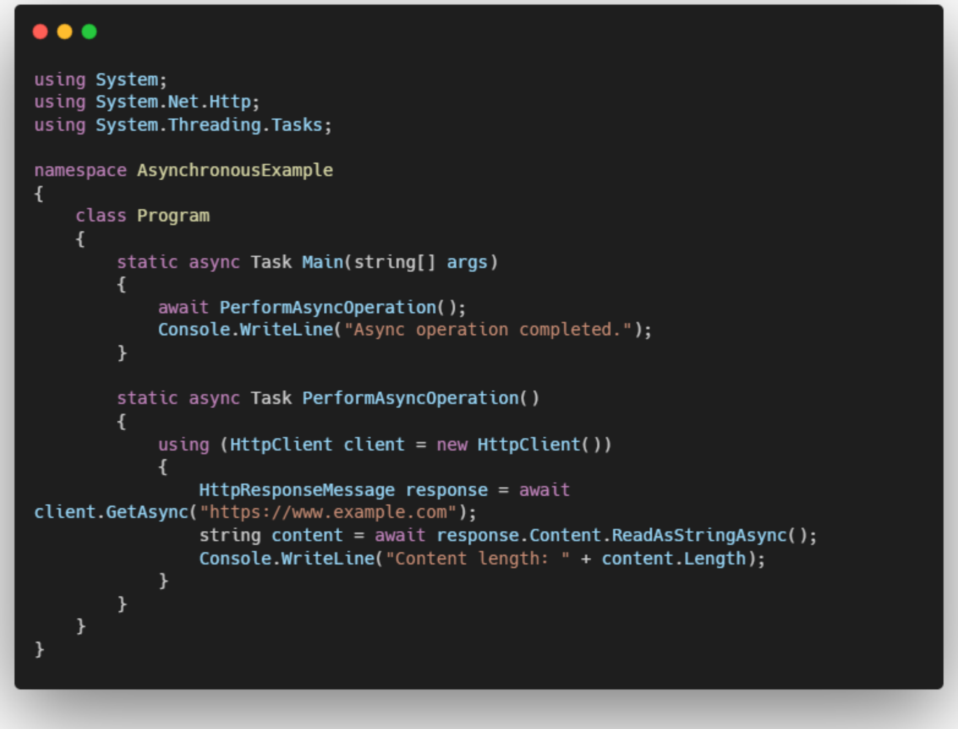
LINQ within ASP.NET offers a streamlined approach to querying and manipulating data, promoting code readability and maintainability. By unifying the querying syntax and providing a wide range of query operators, LINQ simplifies complex data-related tasks, such as sorting, filtering, grouping, and joining. This abstraction technique underscores ASP.NET's commitment to enhancing developer productivity while facilitating efficient and expressive code development.

3.2.2.8 Asyncronous Programming

Asynchronous programming within the ASP.NET framework introduces a vital abstraction that enables developers to create more responsive and efficient applications by allowing tasks to be executed concurrently. Asynchronous operations are particularly beneficial for scenarios involving I/O-bound tasks, such as database queries, web requests, and file operations. This abstraction enhances application responsiveness by preventing blocking and allowing the program to continue executing other tasks while waiting for the completion of asynchronous operations.

In ASP.NET, asynchronous programming is achieved using the **‘async’** and **‘await’** keywords. The **‘async’** modifier signifies that a method contains asynchronous operations, while the **‘await’** keyword pauses the method's execution until the awaited task is completed. This abstraction technique ensures that the application's user interface remains responsive even when performing time-consuming operations.

Consider the following example to illustrate the usage of asynchronous programming within ASP.NET:



```
using System;
using System.Net.Http;
using System.Threading.Tasks;

namespace AsynchronousExample
{
    class Program
    {
        static async Task Main(string[] args)
        {
            await PerformAsyncOperation();
            Console.WriteLine("Async operation completed.");
        }

        static async Task PerformAsyncOperation()
        {
            using (HttpClient client = new HttpClient())
            {
                HttpResponseMessage response = await
client.GetAsync("https://www.example.com");
                string content = await response.Content.ReadAsStringAsync();
                Console.WriteLine("Content length: " + content.Length);
            }
        }
    }
}
```

Figure 3. 24 Asyncronous programming techniques

In this example, the **'PerformAsyncOperation'** method demonstrates asynchronous programming by utilizing the **'async'** and **'await'** keywords. The method performs an asynchronous HTTP request to fetch content from a website while allowing the program to continue execution without waiting for the network operation to complete.

Asynchronous programming within ASP.NET enhances application responsiveness, scalability, and overall performance. By allowing tasks to run concurrently and minimizing blocking, developers can create more fluid user experiences, particularly in applications that involve time-consuming operations. This abstraction technique reflects ASP.NET's adaptability to modern software architecture paradigms and its commitment to optimizing application responsiveness in various scenarios.

CHAPTER 4

CONCLUSION & SUGESTION

4.1 Conclusions

In the contemporary arena of web development, the fusion of modern technologies and well-established frameworks has paved the way for innovative solutions that cater to both user expectations and technical rigor. Through our exploration of Next.js, Tailwind CSS, and ASP.NET, writers have unveiled a spectrum of possibilities that illuminate the path to effective and captivating web development.

Writers journey into the world of Next.js and Tailwind CSS has showcased their prowess in expediting development cycles, optimizing user experiences, and fostering agility in design. The symbiotic relationship between Next.js's server-side rendering and Tailwind CSS's utility-driven design methodology presents developers with an arsenal that not only streamlines development but also crafts interfaces that are aesthetically pleasing and seamlessly interactive.

Meanwhile, our comprehensive dive into the realm of ASP.NET has revealed a robust foundation enriched by academic excellence. ASP.NET's expansive landscape extends far beyond web development, reaching into realms like robotics and enterprise applications. The abstraction programming techniques, including delegates, events, and generics, amplify code efficiency and maintainability, creating a dynamic ecosystem adaptable to diverse demands. The asynchronous programming model and LINQ enhance data processing capabilities, underpinning the creation of responsive and efficient applications.

As writers bridge these modern trends and academic strength, writers underline the significance of blending innovation with tried-and-true frameworks. This synthesis empowers developers to deliver sophisticated solutions that cater to both evolving industry expectations and the precision demanded by academic rigors. The amalgamation of these approaches underscores our commitment to holistic, strategic, and user-centric web development.

In conclusion, our exploration of Next.js, Tailwind CSS, and ASP.NET serves as a testament to the dynamic nature of the web development landscape. By embracing the possibilities offered by modern tools and fusing them with the robustness of established frameworks, writers can architect web solutions that resonate with the present while accommodating the complexities of the future. This balance between innovation and academic prowess propels us to create digital experiences that are not only visually appealing but also technically resilient, shaping the digital landscape in an impactful and meaningful way.

4.2 Sugestions

As writers conclude the exploration of Next.js, Tailwind CSS, and ASP.NET, writers can reflect on the potential avenues for further enhancement and growth in the realm of web development. While these technologies have proven to be instrumental in driving innovation and efficiency, there are always areas where refinement and evolution can continue to contribute to the advancement of the field.

First and foremost, staying attuned to the ever-evolving landscape of web technologies is crucial. As new frameworks, libraries, and methodologies emerge, it's imperative to invest time in researching and understanding how they can augment the development process. Regularly updating skill sets and adopting new tools will enable developers to remain relevant and adaptable in a dynamic field.

Secondly, fostering collaboration and community engagement is paramount. Engaging in online forums, attending web development meetups, and contributing to open-source projects can provide exposure to diverse perspectives and novel ideas. Collaborative efforts often lead to innovative breakthroughs and the sharing of best practices that elevate the collective skill level of the community.

Moreover, continuous improvement in user experience should remain a focal point. User expectations are ever-evolving, and as such, consistently gathering feedback and refining interfaces will ensure that the applications developed align with user needs. Employing user testing and feedback loops can provide insights that drive user-centric design decisions.

Lastly, in the context of ASP.NET, further exploration of advanced programming techniques and leveraging its full ecosystem can be immensely beneficial. From machine learning integration to Internet of Things (IoT) applications, tapping into the diverse applications of ASP.NET can yield innovative solutions that transcend traditional web development boundaries.

In conclusion, the path forward in web development involves a harmonious balance between embracing emerging technologies and continuously refining existing practices. By remaining vigilant in monitoring trends, fostering collaboration, prioritizing user experience, and delving into the depth of frameworks like ASP.NET, developers can navigate the intricate web development landscape with prowess and innovation. Through these suggestions, writers are poised to navigate the future of web development with acumen and ingenuity.

REFERENCES

PT. IDX Consulting (2023). IDX Partners. Accessed on August 18th 2023 https://idxpartners.com/

Grabski, T., 18 great examples of nextjs websites. Pagepro website.accessed on August 18th 2023 https://pagepro.co/blog/nextjs-websites-examples/

NextJS documentation. Nextjs website. Accessed on August 18th 2023 https://nextjs.org/docs

ASP.NET Core documentation. ASP.NET website. Accessed on August 18th 2023 https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0&WT.mc_id=dotnet-35129-website

Nath, P., Create reusable react components the right way. Medium website. Accessed on 18th August 2023 https://javascript.plainenglish.io/create-reusable-react-components-the-right-way-c9b42578a187

Tapha. The main advantage of TailwindCSS. Planflow website. Accessed on 18th August 2023 https://planflow.dev/blog/the-main-advantage-of-tailwindcss

UNIVERSITAS INDONESIA CEP-CCIT FAKULTAS TEKNIK	**CEP - CCIT** **FAKULTAS TEKNIK** **UNIVERSITAS INDONESIA**	No. Form	011/UN2.F4.D2.1.2/AKM/2016 – Form 02
		Revisi	00
		Tanggal	**19 Agustus 2023**
		Halaman	1 dari 1

FORMULIR KONSULTASI PROYEK AKHIR

Nama : RIKI AWAL SYAHPUTRA

Kelas : 4ISA3 Program: ~~TI SE/ TI WD/~~ TI ISA*
Proyek : ~~Software Engineering~~/Integrated System*

Lembaga : PT IDX CONSULTING

Judul : Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET

Pembimbing 1 : LISTYO EDI PRABOWO, S.T.,M.T.

Konsultasi :

No	Tanggal	Uraian	Paraf
1	8/16/2023	Judul:Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET	
2	8/17/2023	Konten : Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET	
3	8/18/2023	Konten : Practical Approaches with Next.js & Tailwind CSS: Exploring Modern Web Development Trends Alongside an Academic Overview of ASP.NET	
PERSETUJUAN PRESENTASI AKHIR			

* coret yang tidak perlu