PROJECT FINAL REPORT

FOR GROUP PROJECT

THE TASK LIST APPLICATION

Group - F1

Information Technology (2020/2021)

Faculty of Applied Science

University of Vavuniya

Date: *December 2023*

Group member details :

Name	Registration Number
P.R.Naveen	2020/ICT/23
L.P.I.Madhumali	2020/ICT/40
T.Lanojika	2020/ICT/50
G.L.S.K.Liyanage	2020/ICT/51
P.M.D.P.R.P.Gunarathna	2020/ICT/85
M.A.S.S Kulathunga	2020/ICT/92

Table of Contents

1.Introduction

1.1 Project Title

The Task List Application (Your Roadmap to Effortless Productivity and Success)

1.2 Project Background

In a fast-paced world where time is of the essence and productivity is paramount, our task list application emerges as the solution to streamline and enhance the way individuals and teams manage their daily activities. Recognizing the growing need for a user-friendly, efficient, and connected task management system, our team embarked on the mission to create an intuitive platform that seamlessly integrates into users' lives.

This task list application aims to transform the traditional approach to task management by providing a centralized hub for organizing, prioritizing, and tracking tasks. With a user-centric design, it caters to the diverse needs of professionals, students, and anyone seeking a tool to optimize their workflow.

As we embark on this journey of redefining task management, our goal is to empower users to take control of their time, reduce stress, and achieve their goals with unparalleled efficiency. Together, let's embrace a future where productivity meets simplicity.

1.3 Project Scope

The scope of the Task List Management Application encompasses the development of a feature-rich, user-friendly platform aimed at optimizing task organization and productivity. The application will allow users to create, prioritize, and categorize tasks, set due dates, and receive timely reminders. It will incorporate user authentication for personalized task management and foster collaboration by enabling users to share and work on task lists collectively. The technology stack will include a chosen programming language, framework, and database, ensuring a secure environment through robust authentication, encryption, and adherence to web application security standards. The project will emphasize comprehensive testing, documentation, and user training, with deployment on a web server or cloud platform to guarantee accessibility across various devices. Ongoing maintenance, support, and scalability considerations will ensure the application's longevity and ability to adapt to evolving user needs. Integration with other tools and compliance with relevant regulations will be explored, and a projected timeline will guide the efficient development and deployment of the application. This scope envisions a holistic solution that revolutionizes task management for individuals and collaborative teams alike.

1.4 Aims and Objectives

Aim :

The primary aim of the Task List Management Application is to revolutionize and elevate the efficiency of task organization and productivity for individuals and collaborative teams. By providing a user-friendly platform equipped with robust features, the application seeks to enhance the way users create, prioritize, and manage their tasks.

Objectives :

1. Task Creation and Editing:

   - Enable users to efficiently create, edit, and delete tasks, providing a seamless interface for managing their to-do lists

2. Due Dates and Reminders:

   - Integrate a calendar feature for setting due dates on tasks and implement a reminder system to ensure users stay informed about upcoming deadlines.

3. User Authentication and Personalization:

   - Establish a secure user authentication system to safeguard user data and personalize the task management experience based on individual preferences.

4. Security Measures:

   - Implement robust security measures, including data encryption and protection against common web application vulnerabilities, to ensure the confidentiality and integrity of user information.

5. Testing and Quality Assurance:

   - Conduct comprehensive testing, including unit testing and user acceptance testing, to identify and address any bugs or issues and ensure a stable and reliable application.

6. Scalability and Future Enhancements:

   - Design the application architecture to be scalable, accommodating potential user growth, and consider future enhancements to meet evolving user needs.

7. User Training and Support:

   - Develop training materials to assist users in maximizing the application's potential and establish channels for user support to address queries and issues promptly.

8. Feedback Mechanism:
   - Implement a feedback system to gather user insights, allowing continuous improvement and refinement of the application based on user experiences and suggestions.

1.5 Final Outcome of the Project

The final outcome of the task list application is a highly user-friendly and versatile tool that streamlines task management with its intuitive interface and comprehensive features. Users can effortlessly create, organize, and prioritize tasks, leveraging options like due dates, priority levels, and tags. The application supports seamless collaboration, enabling users to share and edit tasks with team members. Reminders and notifications ensure that deadlines are never missed, while integration with calendars enhances overall scheduling. With cross-platform compatibility and offline functionality, users can access and update their tasks anytime, anywhere. Robust security measures prioritize data privacy, and regular updates, coupled with responsive customer support, ensure a continually evolving and reliable resource for effective task management. The application not only boosts individual productivity but also facilitates teamwork, making it an indispensable tool for efficient and organized task handling.

2.Project Description

2.1Functional Requirements:

1. User Authentication:
   - Users must be able to create accounts and log in securely to access the task list application.

2. Task Creation:
   - Users should be able to create new tasks with a title, description, due date, priority level, and tags.

3. Task Organization:
   - The application must support the organization of tasks into projects or categories for better management.

4. Task Editing and Deletion:
   - Users should have the ability to edit and delete tasks to accommodate changes in priorities or details.

5. Reminders and Notifications:
   - Users should receive timely reminders and notifications for upcoming task deadlines.

6. Cross-Platform Compatibility:
   - The application must be accessible and functional across various platforms, including web browsers, iOS, Android, and desktop applications.

9. Analytics and Insights:
   - The application should offer analytics and insights into completed tasks, pending items, and productivity trends.

10. Security Measures:
 - Implement robust security measures to protect user data and ensure the privacy of sensitive information.

2.2 Non-Functional Requirements:

1. Performance:
 - The application should have low latency, ensuring a responsive user experience even during periods of high usage.

2. Scalability:
 - The system should be scalable to accommodate an increasing number of users and tasks without compromising performance.

3. Reliability:
 - The application must be reliable, with minimal downtime for maintenance and updates.

4. Usability:
 - The user interface should be intuitive and easy to navigate, requiring minimal training for new users.

5. Compatibility:
 - It should be compatible with a variety of devices, browsers, and operating systems.

6. Data Backup and Recovery:
 - Regular data backups should be performed, and there should be mechanisms in place for quick data recovery in case of system failures.

7. Maintainability:

- The codebase and infrastructure should be well-documented and easily maintainable for future updates and improvements.

8. Accessibility:

- The application should adhere to accessibility standards, ensuring it is usable by individuals with disabilities.

3. Methodology

The RAD methodology is chosen for the development of the Task List application due to its emphasis on rapid prototyping and iterative development. This approach aligns well with the dynamic nature of task management requirements, allowing for quick adjustments and enhancements based on user feedback.

1. Planning:

   -Identify Stakeholders

   - Build Cross-Functional Team

   - Conduct Initial Workshops

   - Develop High-Level Project Plan:

2. User Feedback and Prototyping:

   Develop quick and functional prototypes of the core features of the task list application. Conduct regular feedback sessions with end-users to gather input on the prototypes. Based on user feedback, iterate on the prototypes, refining and enhancing features as needed.

3. Development:

   Implement the features based on the finalized prototypes.Encourage parallel development activities, allowing different teams to work on distinct features simultaneously. Use modular and reusable code to facilitate rapid development and future iterations.

4. Integration and Testing:

   Integrate individual components and features into the complete task list application.Conduct thorough testing, including unit testing, integration testing, and user acceptance testing.Address and resolve any issues identified during testing promptly.

5. Deployment:

Release regular updates and new features to end-users in a controlled and staged manner. Provide user support and monitor user feedback during and after deployment. - Ensure seamless integration with different platforms (web browsers, mobile devices, desktop applications).

6. Evaluation and Continuous Improvement:

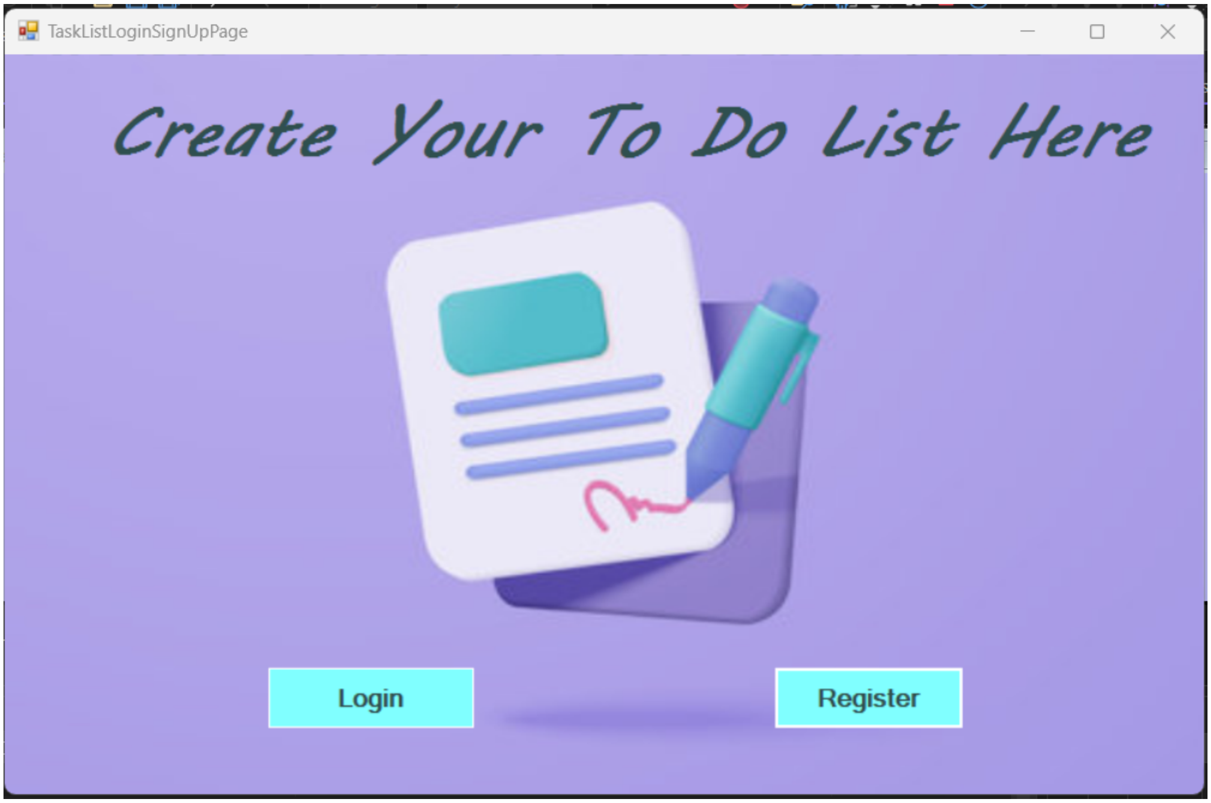
Gather post-implementation feedback from end-users.Analyze application usage and performance data. Identify areas for improvement and plan for subsequent iterations.

7. Iterative Development:

Iterate through the development phases for subsequent releases, incorporating user feedback and making continuous improvements. Be agile and adaptive to changing requirements and priorities throughout the development process.

4.Implementation

1. Landing Page



1. Header Section:

- Create Your To-Do List Here: A prominent heading indicating the main purpose of the page, encouraging users to create and manage their to-do lists.

2. Login Section:

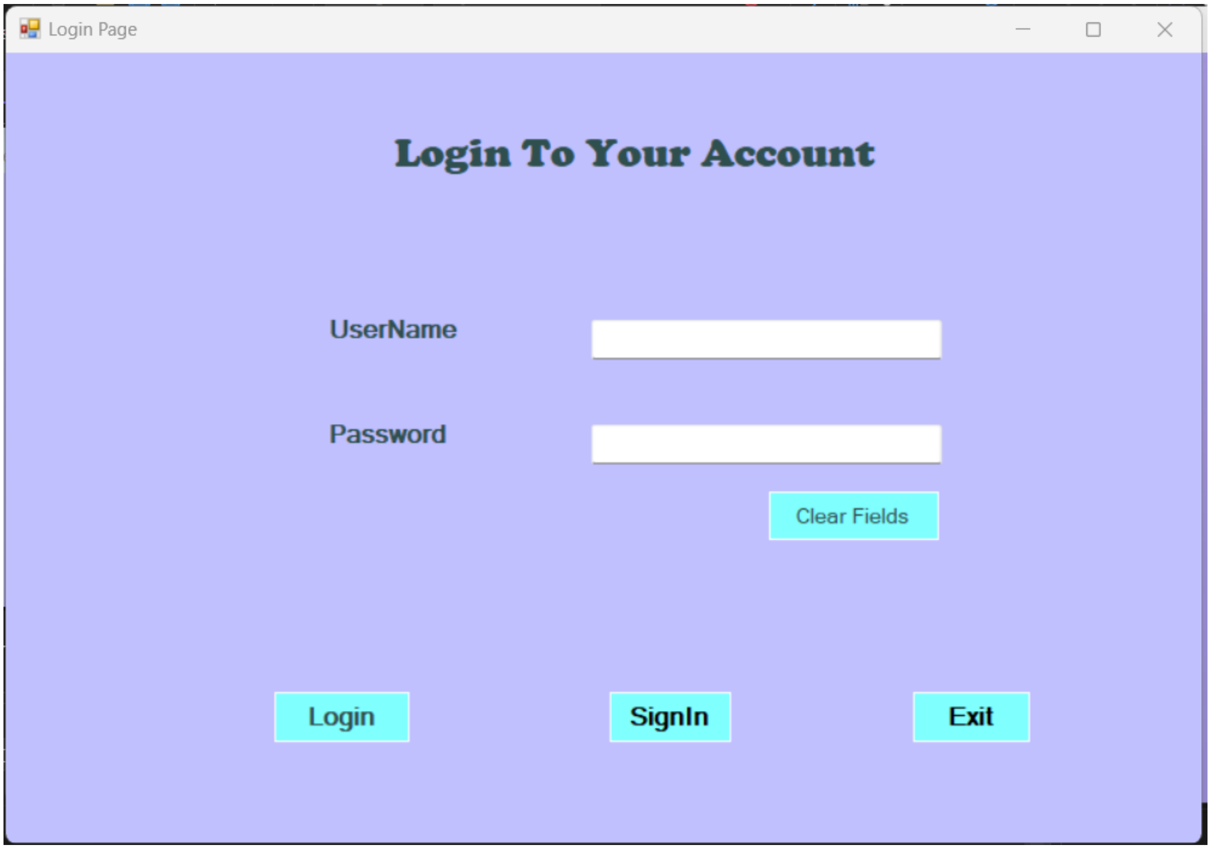
- Login Form: A section where existing users can log in.

3. SignUp Section:

- Register Form: A section for new users to sign up for an account.

The page is designed to serve as a gateway for users who want to access a task management or to-do list application. It provides options for both existing users to log in and new users to register for an account. The central theme is creating and managing to-do lists, emphasizing a user-friendly experience for organizing tasks.

2. Login Page



1. Page Title: Login Page
   - Clearly states the purpose of the page, which is to facilitate user logins.

2. Login Message:
   - "Login To Your Account": A message indicating the action the user is expected to take.

3. Input Fields:

- UserName: Field for users to input their username.
- Password: Field for users to input their password.

4. Actions:

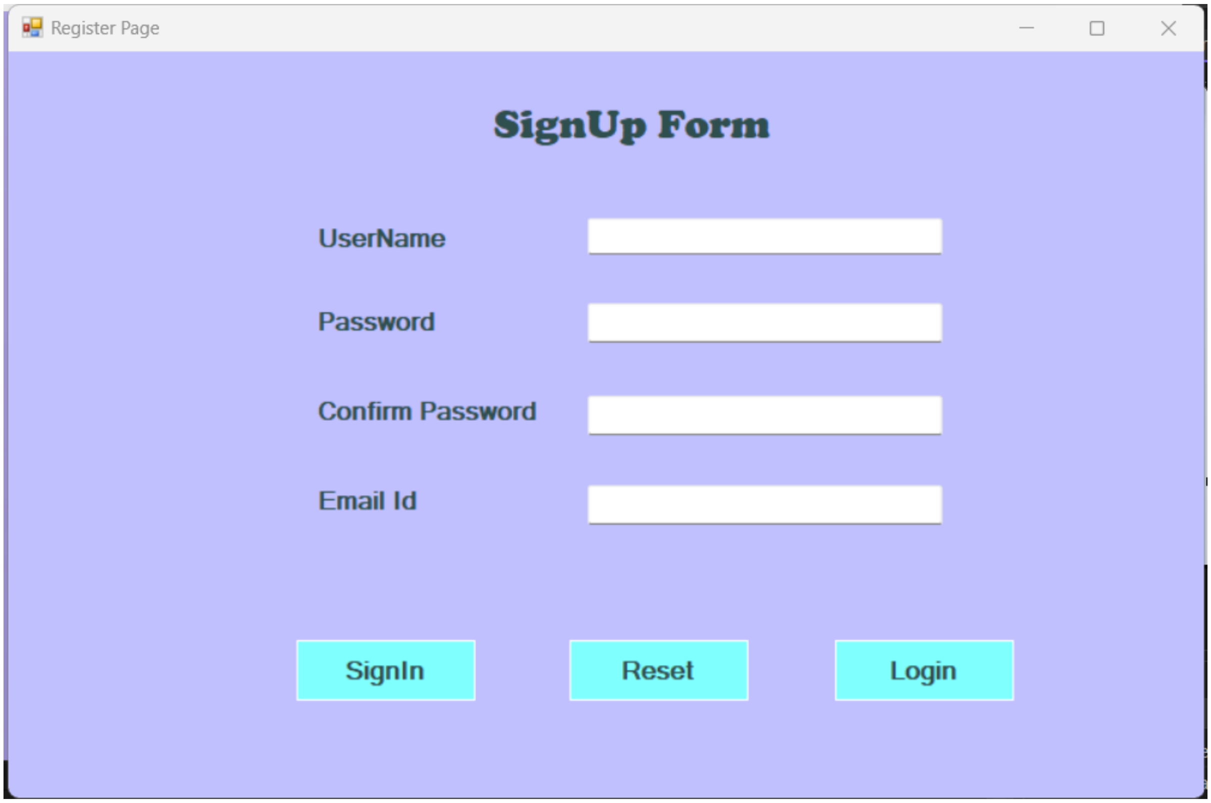
- Clear Fields:- A button that clears the input fields, allowing users to start over if needed.
- Login or SignIn:Button(s) to initiate the login process.
- These buttons are likely meant to perform the same action, but the presence of both might depend on design preferences or user familiarity.

5. Exit:

- Exit Button: A button that allows users to exit or navigate away from the page.

The login page serves the purpose of user authentication. Users are prompted to enter their username and password. The option to clear fields allows for easy correction of input, and the login or sign-in buttons initiate the authentication process. Additionally, an exit button provides users with a convenient way to leave the login page if needed.

3. SignUp Page



1. Page Title: Register Page
   - Clearly indicates the purpose of the page, which is to facilitate user registration.

2. SignUp Form:
   - Indicates the section where users can fill in their registration information.

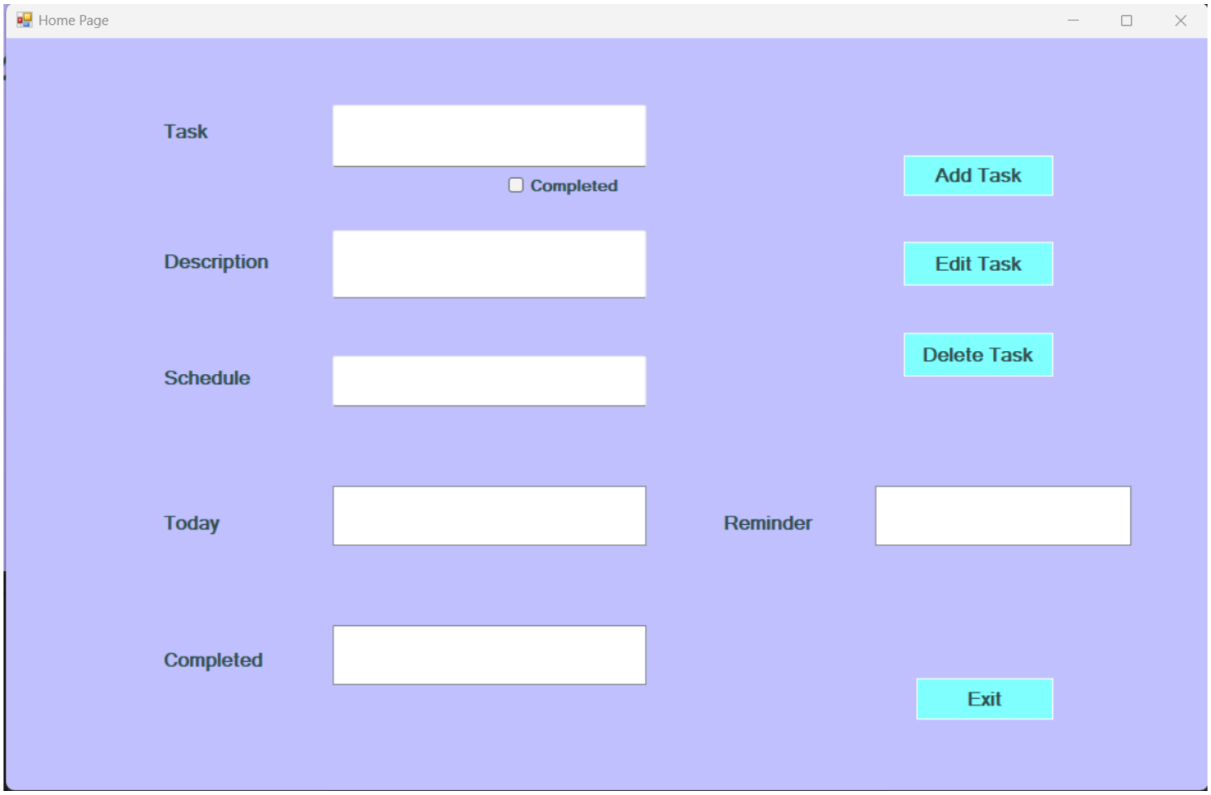
3. Input Fields:
   - UserName: Field for users to input their desired username.
   - Password: Field for users to input a password.
   - Confirm Password: Field for users to re-enter the password to ensure accuracy.
   - Email Id: Field for users to input their email address.

4. Actions:
   - SignIn: Button to submit the registration information and create a new account.
   - Reset: Button that clears all input fields, allowing users to start over if needed.

5. Navigation Links:
   - Login: A link or button that directs users to the login page, providing an option to switch to an existing account.

The register page is designed for user registration. Users can input their desired username, password, confirm the password, and provide an email address. The page features actions like "SignIn" to complete the registration process and create a new account. The "Reset" button allows users to clear all input fields if they want to start over. Additionally, a link or button to "Login" is provided, allowing users to switch to the login page and access an existing account if needed.

4. Home Page



1. Page Title: Home Page

   - Clearly identifies the main landing page or hub of the application.

2. Task Section:

   - Task: Likely displays a list or summary of current tasks that need attention.

3. Completed Section:

   - Completed: Displays a list or summary of tasks that have been marked as completed.

4. Task Management Actions:

   - Add Task: Button or action allowing users to add a new task to their list.
   - Edit Task: Option or button to modify the details of an existing task.
   - Delete Task: Option or button to remove a task from the list.

5. Task Details:

- Description: Section or option to view additional details or descriptions for a selected task.

6. Task Organization:

- Schedule: Section or option to set or view the schedule for tasks.

8. Task Reminders:

- Reminder: Feature or section for setting reminders related to tasks.

9. Date Specific:

- Today: Option or filter for viewing tasks scheduled for the current day.

10. Completed Tasks:

- Completed: Allows users to view a list of tasks that have been marked as completed.

11. Exit:

- Exit: Button or option to leave the home page or log out of the application.

The home page serves as the central hub for managing tasks. Users can view current tasks, completed tasks, and have access to various actions like adding, editing, and deleting tasks. Task details, scheduling, and reminders contribute to a comprehensive task management experience. The inclusion of a date-specific filter (e.g., "Today") enhances the user's ability to organize and prioritize tasks. The "Exit" option allows users to leave the application or log out. Overall, the design prioritizes task management functionality and user-friendly navigation.

5.Resources

1. Visual Studio IDE:-

   [Visual Studio Community edition] (https://visualstudio.microsoft.com/vs/community/): Visual Studio is a powerful integrated development environment (IDE) for building applications with various languages, including C#. The Community Edition is free for individual developers.

2. C# Programming Basics:

   The official Microsoft C# documentation provides a comprehensive guide for beginners and advanced users alike.

3. Windows Forms Application:

   Learn how to create desktop applications using Windows Forms in C#.

4. Database Integration:

   Entity Framework Core is a lightweight, extensible, and cross-platform version of the popular Entity Framework data access technology.

5. User Interface Design:

   If you prefer a more modern and flexible UI, you might want to explore WPF for creating rich desktop applications.

6. ASP.NET for Web Development:

   If you plan to create a web-based task list application, ASP.NET Core is a modern, open-source, and cross-platform framework.

7. Learning Platforms:

[Pluralsight](https://www.pluralsight.com/), [Udemy](https://www.udemy.com/), and [Coursera](https://www.coursera.org/) offer various courses on C# development, Visual Studio, and application development.

8. GitHub Repositories:

Explore open-source projects on platforms like [GitHub](https://github.com/) to see how other developers structure their C# projects. You can find existing task list projects or similar applications for reference.

9. Community Forums:

Participate in forums such as [Stack Overflow](https://stackoverflow.com/) to ask questions, seek advice, and learn from the experiences of other developers.

10. Books:

"C# 9 and .NET 5 – Modern Cross-Platform Development" by Mark J. Price provides a comprehensive guide to modern C# development.

11. Official Microsoft Tutorials:

[Microsoft Learn](https://learn.microsoft.com/en-us/) offers a variety of tutorials, including step-by-step guides on C# development and using Visual Studio.

6. Individual Contribution

1. Landing Page : The landing page is the initial web page visitors see when they arrive at the site.It is designed and developed by
   - T.Lanojika (2020/ICT/50)

2. Login Page : The login page is where users provide their credentials (username and password) to access their personalized accounts.It is designed and developed by
   - P.M.D.P.R.P.Gunarathna (2020/ICT/85)

3. SignUp Page : The sign-up page is where new users can create accounts on the platform.It is designed and developed by
   - G.L.S.K.Liyanage (2020/ICT/51)
   - M.A.S.S Kulathunga (2020/ICT/92)

4. Home Page : The home page is the central hub of a website or application.It is designed and developed by
   - L.P.I.Madhumali (2020/ICT/40)
   - P.R.Naveen (2020/ICT/23)

7.References

1. w3schools : C# Get Started
2. Youtube : https://www.youtube.com/
3. Stack Overflow: https://stackoverflow.com/
4. javatpoint : https://www.javatpoint.com/c-sharp-tutorial
5. Microsoft :

   https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022