

# **TO-DO List Web application**

**Submitted by: Nikhil Dewangan**

**Class: B.Sc. 6th Semester**

**Project Title : TO-DO List  
Web application**

**Date: 08 May 2025**

## Certificate

This is to certify that the project report titled  
" Web Application of To-Do List "  
submitted by Nikhil Dewangan B.Sc. 6th Semester, in  
partial fulfillment of the requirements for the award  
of the degree of Bachelor of Science, is a record of  
an original work carried out by him under my  
supervision. To the best of my knowledge, the matter  
embodied in this report has not been submitted for  
award of any other degree.

Supervisor: \_\_\_\_\_

Signature :

## Acknowledgments

I express my deep gratitude to my project supervisor, Mr. Prashant Vaishanav for his valuable guidance, constructive criticism, and constant encouragement throughout this project. I am thankful to the Department of Computer Science, Guru Ghasidas Vishwavidyalaya , for providing the necessary resources and an environment conducive to research and development. I would also like to acknowledge my parents and friends for their unwavering support and motivation. Finally, I appreciate the feedback from peers and testers that helped refine the website and this report.

# Table of Content

## Chapter 1

1.1 Introduction

1.2 General Organization of the report

## Chapter 2

2.1 Problem Defination

2.2 Existing System

2.3 Proposed system

2.4 system requirements

## Chapter 3

3.1 Architecture Design

3.2 Database Design

## Chapter 4

4.1 Project Description

## Chapter 5

5.1 Language and Tool

5.2 pseudo Code

## Chapter 6

6.1 Unit Testing

6.2 Integration Testing

6.3 Acceptance Testing

6.4 Validation Testing

# Chapter 7

## 7.1 User Manual

# Chapter 8

## 8.1 Sample Screenshot



# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 IMPORTANCE OF THE PROJECT**

The To-Do list application is an essential tool that can help individuals from all walks of life to prioritize and manage their tasks efficiently. It is particularly valuable for people who are busy and often have many things on their plate, such as students, professionals, and entrepreneurs.

One of the primary benefits of the To-Do list application is that it enables users to create, edit, and delete tasks easily. This means that they can keep track of all their responsibilities and quickly make changes as needed. The application also allows users to set due dates for their tasks, which helps them to prioritize their work and ensure that they meet deadlines.

Another advantage of the To-Do list application is that it allows users to track their progress. This feature is essential because it helps individuals to monitor their work and ensure that they are making progress towards their goals. By tracking their progress, users can identify any areas where they are falling behind and make adjustments as needed.

Moreover, the To-Do list application is a centralized solution. This feature is particularly valuable for people. With the ability to access their tasks anytime, users can stay on top of their responsibilities and avoid missing deadlines.

The user-friendly interface of the To-Do list application is another significant advantage. It makes it easy for individuals who may not have experience with task management software to use the application. The ability to switch between different task categories, view all incomplete, complete, today, monthly and yearly tasks in one location, and easily navigate the application makes it a straightforward solution for managing tasks.

In summary, the To-Do list application is an invaluable tool that helps individuals manage their tasks and time more efficiently. Its ability to enable users to create, edit, and delete tasks, set due dates, track progress, access tasks from any location, and its user-friendly interface make it a suitable solution for personal and professional use.

### **1.2 GENERAL ORGANIZATION OF THE REPORT**

In Chapter 2 of the study, the researchers identified a problem with the test structure, which is not aligned with the ideal model derived from applying similitude laws to the prototype online websites. The deviations in the test structure and their potential impact on the correlation between the performance of the test structure and the prototype are outlined in detail. The researchers aim to address this issue and improve the performance of the test structure by aligning it with the ideal model derived from similitude laws.

In Chapter 3, the researchers discussed the system design, which encompasses various subcategories such as Architectural Design, Database Design, and GUI Design. The design process starts with the Data Flow Diagram (DFD), which provides an overview of the system's flow, illustrating how data moves between different components of the system. The Entity Relationship Diagrams (ER) depict the relationships between different entities in the system, showing how data is related to each other and how it is stored in the database. Finally, the Graphical User Interface (GUI) model provides a visual representation of the system's user interface, showcasing how users interact with the system's functionalities.

In Chapter 4, the researchers described the project and its scope. They provided an overview of the system's functionalities and discussed the features that will be implemented.

Chapter 5 discussed the system development process, which included the Language/Tool and Pseudo Code used in developing the system. The researchers discussed the programming languages and software tools used in developing the system and provided sample code snippets to illustrate how the system was developed.

In Chapter 6, the researchers discussed the System Testing and Validation process, which included Unit Testing, Integration Testing, Acceptance Testing, and Validation. They described how the testing was performed and discussed the results of the testing process.

Chapter 7 provided a User Manual for the system, which included instructions for users to operate the system. The researchers discussed the system's functionalities and provided step-by-step instructions on how to use the system.

In Chapter 8, the researchers discussed the System Deployment process, which included the deployment of the system and its installation on the server.

## **CHAPTER 2**

### **SYSTEM ANALYSIS**

#### **2.1 PROBLEM DEFINITION**

The passage highlights the problem that the To-Do list application seeks to address. The modern-day lifestyle is characterized by a fast-paced and demanding routine, making it challenging for individuals to keep track of their tasks and manage their time effectively. It is in this context that the To-Do list application comes in handy.

The application provides a centralized platform for managing and tracking tasks, allowing users to stay organized and focused. The application's user-friendly interface ensures that users can navigate through the application with ease and can add, edit, and delete tasks without any hassle. Additionally, the application is accessible on multiple devices, ensuring that users can manage their tasks from anywhere, anytime, and on any device.

The need for a centralized and accessible platform to manage tasks is essential to ensure that individuals can maintain a work-life balance and prevent burnout. By utilizing the To-Do list application, users can prioritize their tasks and ensure that they are completing them on time, thus reducing stress and improving productivity. In summary, the To-Do list application is an indispensable tool in the modern-day, providing a simple, efficient, and effective solution for managing and tracking tasks.

#### **2.2 EXISTING SYSTEM**

The current landscape of task management applications is characterized by a prevalent lack of integration and synchronization across different devices. Many of these applications are designed to operate in a standalone manner, meaning that they are not interconnected with other devices or applications. This makes it difficult for users to seamlessly access and manage their tasks from multiple devices, as they often need to manually update their progress and status on each individual device.

Furthermore, while some existing systems may offer rudimentary task management features, they often lack the advanced tracking and management capabilities that many users require. This can limit the effectiveness of these applications in helping users stay organized, productive, and on top of their various responsibilities. As a result, there is a growing demand for more comprehensive and integrated task management solutions that can provide users with a seamless and efficient experience across all of their devices.

Some of the Existing system examples are

#### **Google Tasks**

A simple task management application that is integrated with other Google products. It is available on multiple devices, but its functionality is limited, and it does not provide advanced tracking and management capabilities.

## **Microsoft To Do**

A standalone application that allows users to manage their tasks and to-do lists. While it is available on multiple devices, its functionality is limited, and it does not provide advanced tracking and management capabilities.

### **Pros**

- Some existing systems have a user-friendly interface
- Cross-device synchronization is a feature offered by some systems

### **Cons**

- Complex features may make it harder for users to understand and learn how to use the system
- Incorporating too many features can defeat the simplicity of the user interface

## **2.3 PROPOSED SYSTEM**

In response to the challenges posed by the current task management landscape, the To-Do list application has been designed to offer a straightforward and comprehensive solution for users. The application seeks to provide users with the essential features required for effective task management, including the ability to add, edit, and delete tasks, set reminders, and monitor progress. By emphasizing these core functionalities, the To-Do list application aims to offer users a streamlined and efficient task management experience.

Furthermore, the To-Do list application distinguishes itself from other systems by offering a clean and intuitive user interface. Rather than overwhelming users with a cluttered interface, the application prioritizes simplicity and ease of use, making it straightforward for users to manage their tasks and stay organized.

Overall, the To-Do list application aims to provide a simple yet effective solution to the challenges posed by the current task management landscape. By focusing on essential features, cross-device synchronization, and a clean user interface, the application aims to help users stay productive and on top of their responsibilities.

Some of the abstracts proposed are

### **Timely**

The To-Do list application is a task management tool that focuses on simplicity and efficiency. With essential features such as adding, editing, and deleting tasks and tracking progress, users can easily stay organized and productive. The application's user-friendly interface ensures ease of use, making it ideal for anyone who needs to manage tasks effectively.

### **Trac**

Manage your tasks with ease and simplicity using the To-Do list application. This tool lets you add, edit, and delete tasks as well as track your progress. Its intuitive design makes it suitable for anyone who wants to stay on top of their tasks and achieve their goals.

## Taskaid

The To-Do list application is a simple and effective way to organize your tasks. You can create, modify, and remove tasks as well as monitor your progress. The application has a user-friendly interface that allows you to use it without any hassle. Whether you need to manage personal or professional tasks, this tool can help you get things done.

### Pros

- Provides a simple, yet comprehensive solution for managing tasks
- Focuses on essential features like adding, editing, and deleting tasks, setting reminders, and tracking progress
- Prioritizes a clean and intuitive design for easy navigation and task management
- Aims to address the limitations of existing systems by providing a simple and efficient solution for managing tasks.

### Cons

- No Email Connectivity
- No support for multiple platforms (as of planned)

## 2.4 SYSTEM REQUIREMENT

### 2.4.1 HARDWARE REQUIREMENT:

- X86 Processor 2.0 GHz or above
- 2 GB Ram or above
- Minimum of 20 GB of storage space

### 2.4.2 SOFTWARE REQUIREMENT:

- Microsoft windows 7/8/10/11
- XAMPP server (Html, PHP, MySql)
- VS code or any other code/text editor
- Browser

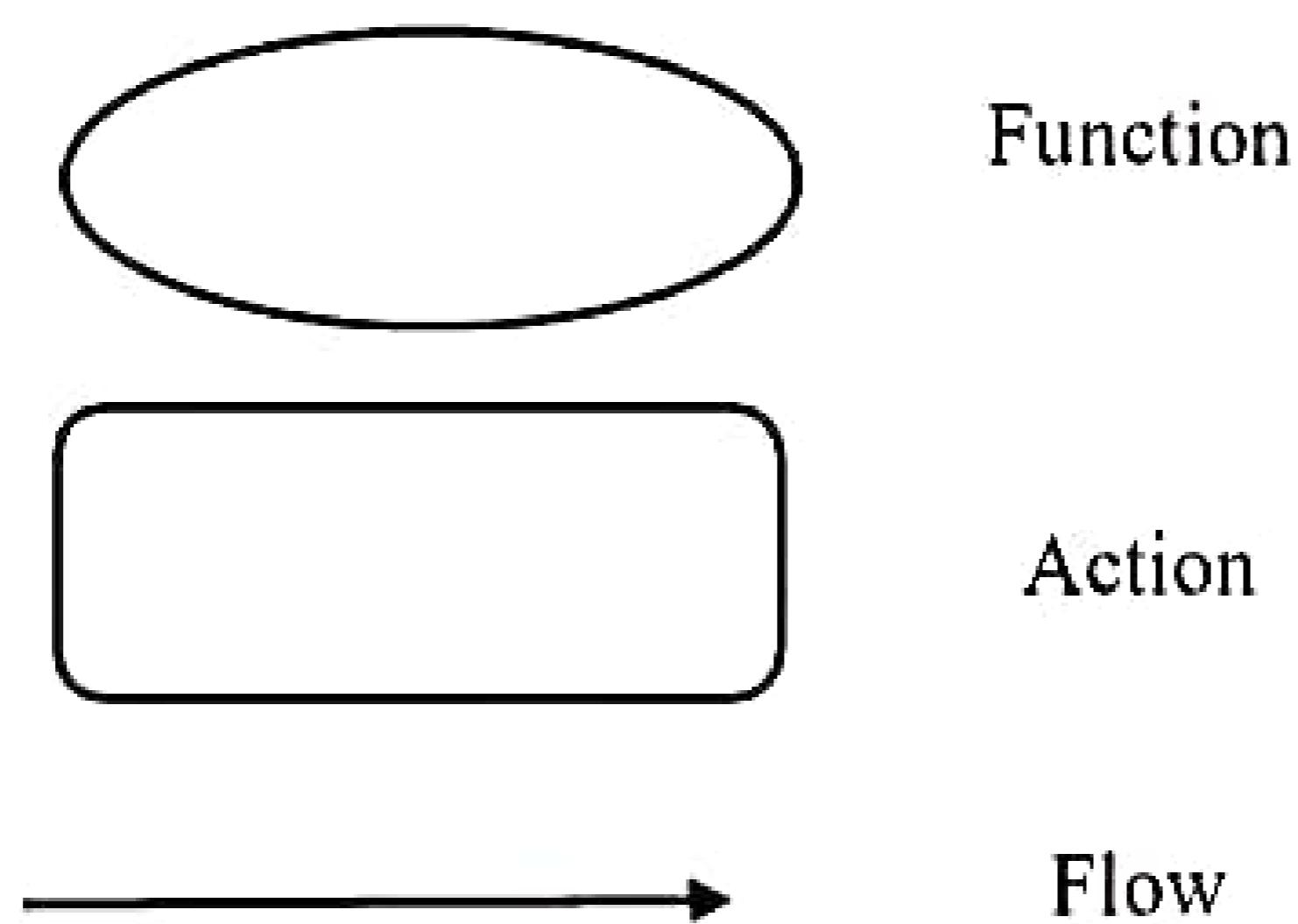
# CHAPTER 3

## SYSTEM DESIGN

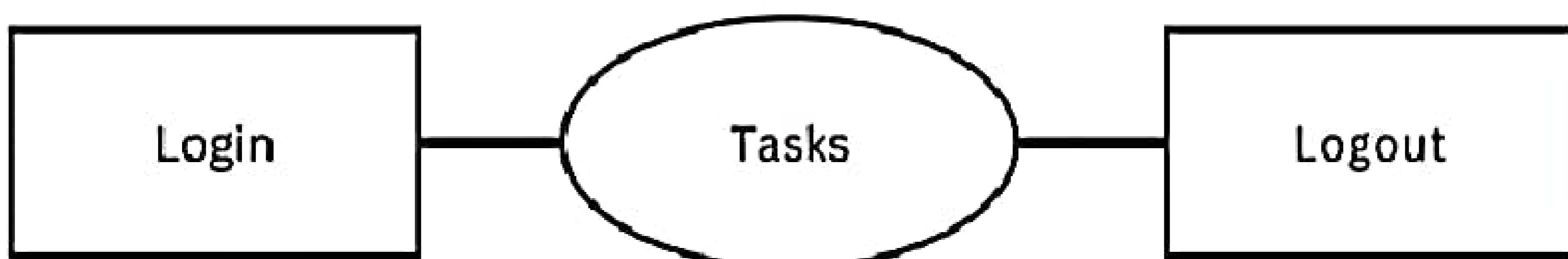
### 3.1 ARCHITECTURAL DESIGN

#### Data flow diagram (DFD)

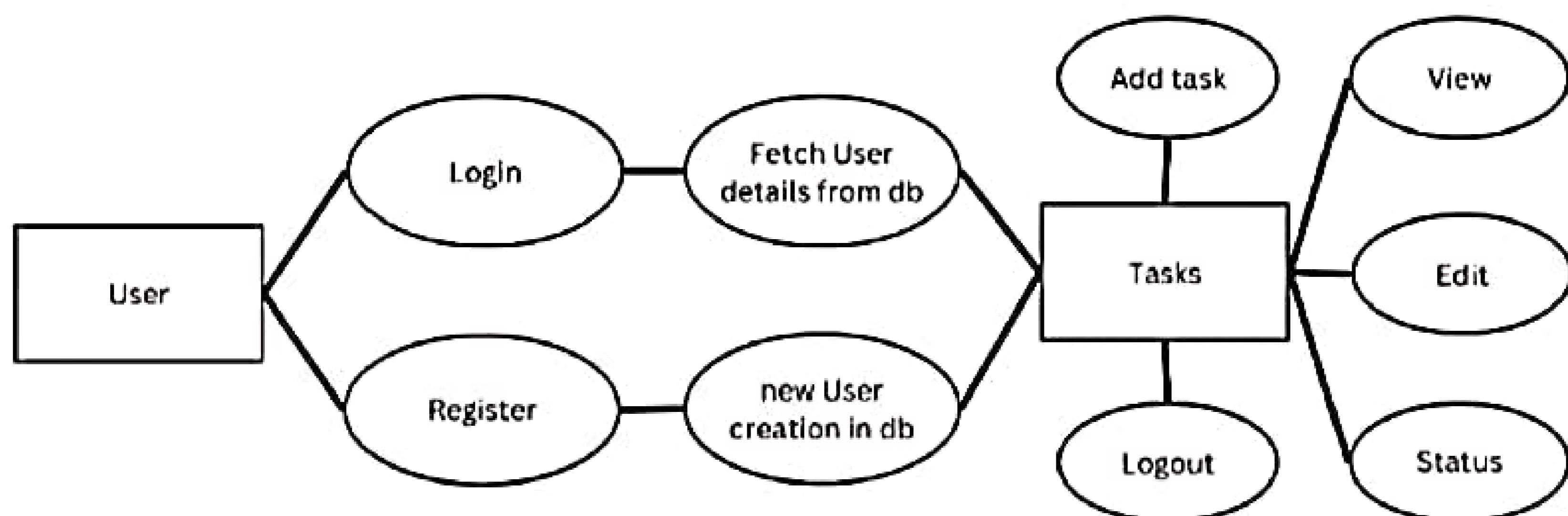
A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFD's can also be used for the visualization of data processing (structured design).



#### LEVEL 0



#### LEVEL 1



### 3.2 DATABASE DESIGN

#### Entity relationship diagram (ERD)

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

#### ER DIAGRAM

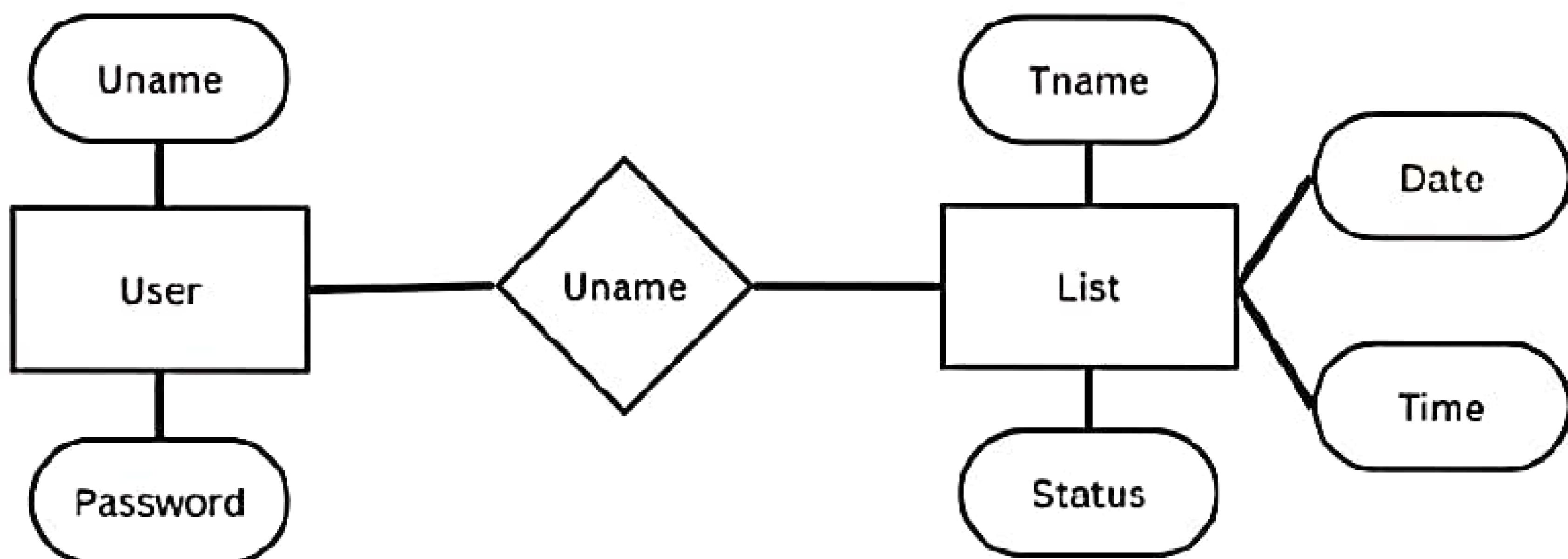


Table name : user

Primary key : uname

Description : To store details of the users (password and username)

Serial. No	Fields	Datatype	Description
1	uname	varchar(25)	Primary key, not null
2	password	varchar(25)	not null

Table name : list

Foreign key : uname

Description : To store details of the Tasks(tname, date, time, status) created by the users

Serial. No	Fields	Datatype	Description
1	uname	varchar(25)	Foreign key, not null
2	tname	varchar(255)	not null
3	date	date	
4	time	time	
5	status	varchar(10)	not null

## **CHAPTER 4**

### **PROJECT DESCRIPTION**

The To-Do List Application is a web-based project that helps users to manage their daily tasks and track their progress.

Overall, this project aims to provide users with a user-friendly and efficient way to manage their daily tasks. By using this application, users can prioritize their tasks, track their progress, and achieve their goals efficiently.

There are several modules in the proposed system, which will help the end user to interact with the system with ease. The below specified are key modules which are integrated in the project.

1. Login
2. Registration
3. Tasks
4. Add Task
5. View Tasks
6. Task Status

#### **Login**

The User Login module is an essential part of the To-Do List application that allows registered users to access their accounts. The module provides a standard login interface for the users where they can enter their username and password to authenticate themselves.

Upon successful login, the user is redirected to the main Tasks page, where they can view, edit, delete, and add new tasks. The user's login credentials are securely stored in the application's database.

The User Login module can also include additional features such as password reset functionality, multi-factor authentication, and social media login integration to enhance the user's login experience. The password reset functionality allows the user to reset their password in case they forget it or suspect a security breach. The multi-factor authentication provides an additional layer of security by requiring the user to provide a second factor, such as a one-time password sent to their phone, to access their account. The social media login integration allows the user to log in to their account using their social media accounts, such as Facebook or Google, making the login process more convenient for the user.

In summary, the User Login module is a crucial component of the To-Do List application that ensures secure and convenient access for registered users to manage their daily tasks efficiently.

#### **Registration**

The Registration of New User module is an integral part of the To-Do List application that enables new users to create an account and start using the application. The module provides a

simple and easy-to-use interface where new users can register by providing their basic details, such as their username and password.

Upon registration, the user's information is securely stored in the application's database, and a unique user ID is generated for the new user. The user ID and password are required for the user to access their account and start using the To-Do List application.

To ensure the user's privacy and security, the Registration of New User module may include additional features such as email verification, captcha verification, and password strength validation. The email verification feature sends a verification email to the user's email address, which they need to verify to activate their account. The captcha verification feature prevents automated bots from creating fake accounts, and the password strength validation feature ensures that the user's password is strong enough to protect their account from unauthorized access.

In summary, the Registration of New User module is a critical component of the To-Do List application that enables new users to create their account and start using the application. The module ensures the user's privacy, security, and convenience while registering, making the process simple and straightforward for the new users.

## Tasks

The To-Do list application offers a range of features that enable users to manage their tasks and time more efficiently. One of the most crucial functions of the application is the ability to add tasks easily. This feature is essential because it allows users to keep track of all the tasks they need to complete. Users can add new tasks to their To-Do List by entering the task name and description, along with the optional date and time.

To make it easier for users to navigate their tasks, the application offers view options. Once users have added their tasks, they can choose to view them based on their time frame. Users can choose to view their tasks for the day, week, month or year by selecting the appropriate option from the drop-down box.

Another useful feature of the To-Do list application is the Tasks Status option. Upon selecting this button, users are shown a page where they can view their tasks based on their status. They can view tasks that are incomplete, completed, or overdue. This feature allows users to track their progress and ensure that they are meeting their deadlines.

Finally, the application provides a logout button for users to log out of the main page. This feature is particularly useful for individuals who share devices or are using public computers. By logging out of the application, users can ensure that their tasks and data remain private and secure.

In conclusion, the To-Do list application provides a range of features that enable users to manage their tasks efficiently. The ability to add tasks easily, view options, Tasks Status, and logout option all contribute to making the application user-friendly and straightforward.

## Add Tasks

The To-Do list application provides users with a variety of tools to create and manage their tasks. One of the key features is the task name textbox, which allows users to enter the name of their task or a short description. This textbox serves as a placeholder for users to save their tasks and retrieve them later.

Once users have entered their task name or description, they can then set a due date using the task date field. This feature is particularly useful for individuals who have multiple tasks to manage and need to prioritize their workload. Users can then set a due time for their task, if necessary, using the task time field.

To add the task to the database, users can click on the Add Task button. However, this button only works when all the required fields for the task details are filled. This ensures that users do not accidentally save incomplete tasks, which can lead to confusion and missed deadlines.

To navigate back to the home page, users can click on the Home button. This button serves as a quick and easy way to return to the main page of the application. It is particularly useful for individuals who are managing multiple tasks and need to switch between different screens quickly.

In conclusion, the task name textbox, task date field, task time field, Add Task button, and Home button are all essential features of the To-Do list application. They provide users with the necessary tools to create and manage their tasks efficiently, ensuring that they can stay on top of their workload and meet their deadlines.

## Edit Tasks

The To-Do List application provides users the ability to edit their tasks like the task name, due date, time, etc.

The drop down box provides users with all the tasks present in their To-Do List, the users can choose a task and edit it on the go.

In conclusion, the Edit Tasks option provide users with a comprehensive ability to alter their tasks, allowing them to manage their time more efficiently. The Home button provides users with a convenient way to navigate back to the main page and access other features and options of the application.

## View Tasks

The To-Do list application provides users with various options to view their tasks based on their due dates. One of the options available to users is Today's Tasks, which displays a list of incomplete tasks that are due on the current day. This feature is useful for individuals who want to plan their schedule for the day or ensure that they do not miss any important deadlines.

Similarly, the This Week's Tasks option displays a list of incomplete tasks that are due within the current week. This feature allows users to plan their schedule for the week and ensure that they are on track to complete all of their tasks.

The This Month's Tasks option displays a list of incomplete tasks that are due within the current month. This feature provides users with a broader view of their workload, allowing them to plan their schedule for the month and prioritize their tasks accordingly.

The This Year's Tasks option displays a list of incomplete tasks that are due within the current year. This feature is particularly useful for individuals who have long-term goals or projects that span multiple months or even years.

To return to the home page, users can click on the Home button. This button serves as a quick and easy way to navigate back to the main page of the application and access other features and options.

In conclusion, the Today's Tasks, This Week's Tasks, This Month's Tasks, and This Year's Tasks options provide users with a flexible and customizable way to view and manage their tasks based on their due dates. The Home button serves as a convenient way to navigate back to the main page of the application and access other features and options.

### **Tasks Status :**

The To-Do list application provides users with options to view both their incomplete and completed tasks. The In-Completed Tasks option displays a list of all the tasks that have not yet been completed, allowing users to plan their schedule and prioritize their tasks accordingly. Users can mark specific tasks as completed as they complete them, which will move them to the Completed Tasks list.

The Completed Tasks option displays a list of all the tasks that have been marked as completed by the user. This feature allows users to review their completed tasks and gain a sense of accomplishment as they see the progress they have made.

The Home button is available on every page of the application, allowing users to quickly return to the main page. This button provides users with a convenient way to navigate back to the home page and access other features and options.

In conclusion, the In-Completed Tasks and Completed Tasks options provide users with a comprehensive view of their tasks, allowing them to manage their time more efficiently. The Home button provides users with a convenient way to navigate back to the main page and access other features and options of the application.

## **CHAPTER 5**

### **SYSTEM DEVELOPMENT**

#### **5.1 LANGUAGE / TOOL**

##### **HTML and CSS**

HTML and CSS are essential technologies for building web pages and web applications. HTML, which stands for Hypertext Markup Language, provides the structure and content of web pages. CSS, which stands for Cascading Style Sheets, is used for page layout, formatting, and design.

##### **HTML**

- Stands for Hypertext Markup Language.
- Provides the structure and content of web pages.
- Allows designers to publish online documents with text, tables, images, and other media.
- Offers several features for designers and developers, including the ability to create interactive forms, retrieve information from remote services, and integrate multimedia elements like audio and video clips.

##### **CSS**

- Stands for Cascading Style Sheets.
- Used for page layout, formatting, and design.
- Used in the To-Do list project to style the web application, including layout, colours, and fonts.
- Allows the designer to adjust the presentation of the web application to different types of devices, including small screens, large screens, and printers.

##### **Combination**

- The combination of HTML and CSS is fundamental to building modern, responsive web applications that provide a rich user experience.
- Used in the To-Do list project to create the user interface and structure of the web application.
- Also utilized additional features of HTML and CSS, such as the ability to create forms and navigate between pages.

Overall, HTML and CSS are essential technologies for building web pages and web applications. They offer several features for designers and developers, allow for the creation of interactive and multimedia-rich web pages, and are fundamental to building modern, responsive web applications that provide a rich user experience.

## **PHP**

PHP, short for hypertext pre-processor, is a server-side scripting language primarily designed for web development, but it is also used for general-purpose programming. Originally created in 1994 by Rasmus, it was initially called "Personal Home Page." Today, the reference implementation of PHP is produced by the PHP Group. PHP is a widely-used language with over 240 million websites and 2.1 million web servers using it as of January 2013.

One of the key features of PHP is its ability to be combined with various web frameworks and templating engines, allowing developers to build complex web applications. The language is often mixed with HTML code to dynamically generate web pages. The PHP code is generally processed by a PHP interpreter, which can be executed as a native module of the web server or as a common gateway interface (CGI) executable.

### **Functionality**

- Ability to be combined with various web frameworks and templating engines to build complex web applications.
- Often mixed with HTML code to dynamically generate web pages.
- Processed by a PHP interpreter, which can be executed as a native module of the web server or as a common gateway interface (CGI) executable.
- Standard PHP interpreter is powered by the Zend Engine, which is free software released under the PHP license.
- Multiple versions of PHP available, with version 5.5 being used for the current application.

### **Usage**

- Used in the current project to handle server-side processing, including database interactions and user authentication.
- Ability to work seamlessly with HTML made it a natural choice for dynamically generating web pages.
- Project made use of various PHP frameworks and libraries to streamline development and improve code efficiency.

Overall, PHP is a powerful and versatile language that plays a vital role in modern web development. Its ability to handle server-side processing and work seamlessly with HTML make it an essential tool for building dynamic and interactive web applications.

## **MySql**

SQL stands for Structured Query Language, which is a programming language used to manage and manipulate relational databases. MySQL is a popular open-source Relational Database Management System (RDBMS) that is widely used for web applications and is a crucial component of the LAMP (Linux, Apache, MySQL, PHP/Perl/Python) software stack. MySQL

is utilized by various software applications, including WordPress, Joomla and many others. It is also used by large-scale websites like Google, YouTube, Facebook, Twitter, and Flickr.

MySQL is written in C and C++ and was created by a Swedish company. The first version of MySQL was released on May 23, 1995. MySQL has several versions, and the version used in this project is 5.6.17.

## MySQL Features

- Scalability: can handle large amounts of data and users without sacrificing performance
- Security: provides various security mechanisms to protect data from unauthorized access and ensure data privacy
- ACID Compliance: ensures that transactions are atomic, consistent, isolated, and durable (ACID) to maintain data integrity
- High Availability: supports various replication techniques that enable the database to be available even in case of system failures

## MySQL in Project

- Used as a database to store and manage the application data
- Used to store user data, task data, and various other data related to the application.

## 5.2 PSEUDO CODE

Pseudo code is a term which is often used in programming and algorithm-based fields. It is a methodology that allows the programmer to represent the implementation of an algorithm. Simply, we can say that it's the cooked-up representation of an algorithm.

### Login

1. Start
2. Create HTML file named loginpage.html
3. Create a form using HTML and CSS
  - add input field for username and password
  - add button to submit form data
  - add a link to registration page
4. Create a PHP file named login.php
  - Start session
  - Connect to MySQL database
  - If connection fails, display error message
  - If the form is submitted, retrieve the values of the username and password fields
  - Run a SQL query to select a user with matching username and password
  - If there is exactly one matching row in the result set:
  - Store the username in a session variable

- Redirect to the tasks.php page
  - Otherwise, redirect to the registration page
5. End

## Registration

1. Start
2. Create HTML file named registerpage.html
  - Create a form for registration
  - Add input fields for username and password
  - Add submit button
3. Create a PHP file named register.php
  - Connect to MySQL database
  - If connection fails, display error message
  - If form is submitted, retrieve the values of the username and password fields
  - Check if the username already exists in the database
  - If it exists, display an alert message and redirect to registration page
  - If it doesn't exist, insert the new user information into the database
  - If successful, redirect to the login page
  - If unsuccessful, display error message
4. End

## Tasks

1. Start session
2. Connect to database
3. Get logged in user's username
4. Echo welcome message with logged in user's username
5. Display task operation section with Add Task, View Tasks, Status, and Log-Out button
6. Echo About Tasks section with information about the project
7. End

## Add Task

1. Start session
2. Connect to database
3. Get logged in user's username
4. Echo add task page title with logged in user's username
5. Display form for adding task with input fields for task name, date and time, and submit button
6. If form is submitted with non-empty task name, get task name, date and time from form input
7. Insert task into list table with logged in user's username, task name, date, and time

8. Close database connection
9. End

## View Tasks

1. Start session
2. Connect to database
3. Get logged in user's username
4. Echo View tasks page title with logged in user's username
5. Display the drop box to select other view options
6. Echo home button
7. Display the tasks based on the selected view option
8. Close database connection
9. End

## Tasks Status

1. Start session
2. Connect to database
3. Retrieve the username from the session
4. If the "completed" button is pressed: a. Retrieve the task name from the form b. Update the task status in the database to "completed" for the corresponding user and task name
5. Retrieve all incomplete tasks for the user from the database, ordered by date and time
6. Display the incomplete tasks in a table with columns for task name, date, time, status, and an action column with a "Complete" button for each task
7. If the "incomplete" button is pressed: a. Retrieve the task name from the form b. Update the task status in the database to "incomplete" for the corresponding user and task name
8. Retrieve all completed tasks for the user from the database, ordered by date and time
9. Display the completed tasks in a table with columns for task name, date, time, status, and an action column with an "Incomplete" button for each task.
10. Close database connection
11. End

## **CHAPTER 6**

### **SYSTEM TESTING AND VALIDATIONS TESTING**

Testing is the process of executing a program or system with the intent of finding errors. Or, it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Software is not unlike other physical processes where inputs are received, and outputs are produced. Where software differs is in the way it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all the different failure modes for software is generally infeasible.

Testing is one of the most important stages in software development which can prove whether the implementation and the requirements are in fidelity with each other. Testing can also confirm if the software specifications are complete and consistent.

One of the main goals of testing is to have a minimum number of test cases that will find most implementation errors.

**Types of Testing:**

- Unit testing
- Integration Testing
- Acceptance Testing
- Validation
- System Testing
- Functional Testing
- Smoke Testing
- Regression Testing
- Performance Testing
- Security Testing
- User Acceptance Testing

#### **6.1 Unit Testing**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing involves only those characteristics that are vital to the performance of the unit under test. The units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system many modules at various levels are needed. Unit testing focuses on the modules independently of one another, to locate errors. Unit testing can be time-consuming and tedious. It demands patience and thoroughness on the part of the development team. Rigorous documentation must be maintained. Unit testing must be done with an awareness that it may not be possible to test a unit for every input scenario that will occur when the program is run in a real-world environment. Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

## **Implementation**

### **login and registration system**

For the login and registration system, we need to test that the user can create an account, log in, and log out without any errors. We should also test that the application handles incorrect login credentials, such as an invalid email or password.

### **main task page**

For the main task page, we need to test that the user can view their tasks actions buttons

### **add task page**

For the add task page, we need to test that the user can create a new task and that the task is added to the main task page. We should also test that the user cannot create a task without entering the required information, such as the task name or due date.

### **edit task page**

For the edit task page, we need to test that the user can edit an existing task and that the changes are saved correctly. We should also test that the user cannot edit a task that belongs to another user.

### **view task page**

For the view task page, we need to test that the user can view the Tasks in specified time periods

### **task status page**

For the task status page, we need to test that the user can change the status of a task, such as from incomplete to complete, and that the changes are reflected on the view task page.

Overall, unit testing is crucial for ensuring that a to-do list project is working as intended and providing a seamless user experience. By testing each feature thoroughly, we can catch any errors or bugs early on in the development process and ensure that the application meets the user's needs.

## **6.2 Integration Testing**

Integration testing is an important aspect of ensuring that all the features of a to-do list project work seamlessly together. This project includes a login and registration system, a main task page, and additional pages for adding, editing, viewing, and changing the status of tasks. Integration testing should be performed to verify that each of these features interacts with one another properly.

## **Implementation**

### **login and registration system**

For the login and registration system, we need to test that user accounts are created successfully and that the login system grants access to the correct user accounts. We should also test that the system rejects invalid login credentials and that logging out clears the session data.

### **main task page**

For the main task page, we need to test that the user can view their tasks actions buttons

### **add task page**

For the add task page, we need to test that new tasks are added to the database correctly, and that the page updates to reflect the new task. We should also test that adding a new task preserves any data that was entered previously.

### **edit task page**

For the edit task page, we need to test that changes made to an existing task are saved properly, and that the task is updated correctly on the main task page. We should also test that attempting to edit a task that belongs to another user results in an error message.

### **view task page**

For the view task page, we need to test that users can view details of a specific timed tasks

### **task status page**

For the task status page, we need to test that changing the status of a task from incomplete to complete or vice versa is reflected correctly on the main task page. We should also test that changing the status of a task does not affect any other tasks.

Overall, integration testing helps ensure that all the features of a to-do list project work seamlessly together to provide the best possible user experience. By testing how different features interact with each other, we can catch any issues that might arise and ensure that the application meets the user's needs.

## **6.3 Acceptance Testing**

Acceptance testing is a critical stage of software development that aims to verify whether the to-do list project meets the user's requirements and expectations. In this to-do list project, we need to conduct acceptance testing to ensure that the system functions correctly, and the user interface is intuitive and easy to use. Acceptance testing should involve end-users, who will use the application to evaluate whether it satisfies their needs and expectations.

## **Implementation**

### **login and registration system**

For the login and registration system, we need to ensure that the user interface is user-friendly, and users can register, log in, and recover their password easily. The acceptance testing should focus on verifying that the login and registration system meets the user's security and privacy expectations.

### **main task page**

For the main task page, we need to test that the user can view their tasks actions buttons

### **add task page**

For the add task page, we need to ensure that users can add tasks easily and quickly. Acceptance testing should focus on verifying that the form fields are intuitive and easy to use, and that users can enter data quickly and without any errors.

### **edit task page**

For the edit task page, we need to ensure that users can edit tasks quickly and easily. Acceptance testing should focus on verifying that the form fields are intuitive and easy to use, and that users can edit tasks without any errors or confusion.

### **view task page**

For the view task page, we need to ensure that users can view tasks easily and access all details about a task, including its date status sorted through the specified time period.

### **task status page**

For the task status page, we need to ensure that users can change the status of a task quickly and easily. Acceptance testing should focus on verifying that users can mark tasks as complete or incomplete without any errors or confusion.

In conclusion, acceptance testing is an important step in ensuring that the to-do list project meets the user's requirements and expectations. By involving end-users in the testing process, we can obtain feedback and make any necessary changes to ensure that the system functions correctly and provides an intuitive user experience.

## **6.4 Validation:**

Validation is a crucial aspect of developing a to-do list project that contains a login, register, main tasks, add task, edit task, view task, and task status pages. Validation aims to ensure that users can input data accurately and prevent them from entering invalid data that could lead to errors or security issues.

## **Implementation**

### **login and registration system**

For the login and registration system, we need to validate the user's input to ensure that they provide valid data, such as a valid string of username and password. We also need to validate the user's input for the password recovery process to prevent unauthorized access to the system.

### **main task page**

For the main task page, we need to validate the proper functioning of the provided buttons

### **add task page**

For the add task page, we need to validate the user's input to ensure that they provide valid data for all required fields. We should also validate any optional fields to ensure that the user provides valid data and to prevent any errors or issues with the task's details.

### **edit task page**

For the edit task page, we need to validate the user's input to ensure that they provide valid data for all required fields. We should also validate any optional fields to ensure that the user provides valid data and to prevent any errors or issues with the task's details.

### **view task page**

For the view task page, we need to validate the user's input to ensure that they can view tasks correctly, including their due date, status depending on their specified time slot.

### **task status page**

For the task status page, we need to validate the user's input to ensure that they can update the status of a task correctly and prevent them from entering invalid data that could lead to errors or issues with the task's details.

In conclusion, validation is critical for ensuring that the to-do list project functions correctly and provides an excellent user experience. By validating the user's input, we can prevent errors and issues, improve the system's performance, and ensure that users can complete tasks effectively and efficiently.

## **CHAPTER 7**

### **USER MANUAL**

#### **Introduction**

Welcome to my To-Do List project, a tool designed to help you organize and manage your tasks efficiently. This user manual will guide you through the features and functionalities of the system, including login, registration, main tasks, add task, edit task, view task, and task status pages.

#### **Login**

To access the system, you need to have an account. If you already have one, enter your username and password in the login page and click "login." If you don't have an account yet, click on "register" to create a new account.

#### **Registration**

To register for an account, click on the "register" button on the login page. You will be prompted to enter your name, email address, and password. Once you have entered the information, click "submit," and your account will be created.

#### **Main Tasks**

After you have logged in, you will be directed to the main tasks page. This page displays an array of buttons to navigate to the functions offered and a short description of the To-Do List

#### **Add Task**

To add a new task, click on the "add task" button on the main tasks page. You will be prompted to enter the task's title, description, due date, priority, and status. Once you have entered the information, click "submit," and the new task will be added to your list.

#### **Edit Task**

To edit an existing task, click on the "edit" button next to the task you want to modify. You can update the task's title, description, due date, priority, and status. Once you have made the changes, click "submit," and the task will be updated.

#### **View Task**

To view a task's details, click on the task's title in the main tasks list. This will open a new page displaying the task's description, due date, priority, and status.

## **Task Status**

To update a task's status, click on the "edit" button next to the task you want to modify. You can change the task's status from "in progress" to "completed" or "cancelled." Once you have made the changes, click "submit," and the task's status will be updated.

## **Conclusion**

My To-Do List project is an effective tool for managing and organizing your tasks. By following this user manual, you can take full advantage of the system's features and functionalities to keep track of your tasks and increase your productivity.

## APPENDIX B

### SAMPLE SCREENSHOT

#### LOGIN PAGE

To access the system, you need to have an account. If you already have one, enter your username and password in the login page and click "login." If you don't have an account yet, click on "register" to create a new account.



#### REGISTRATION PAGE

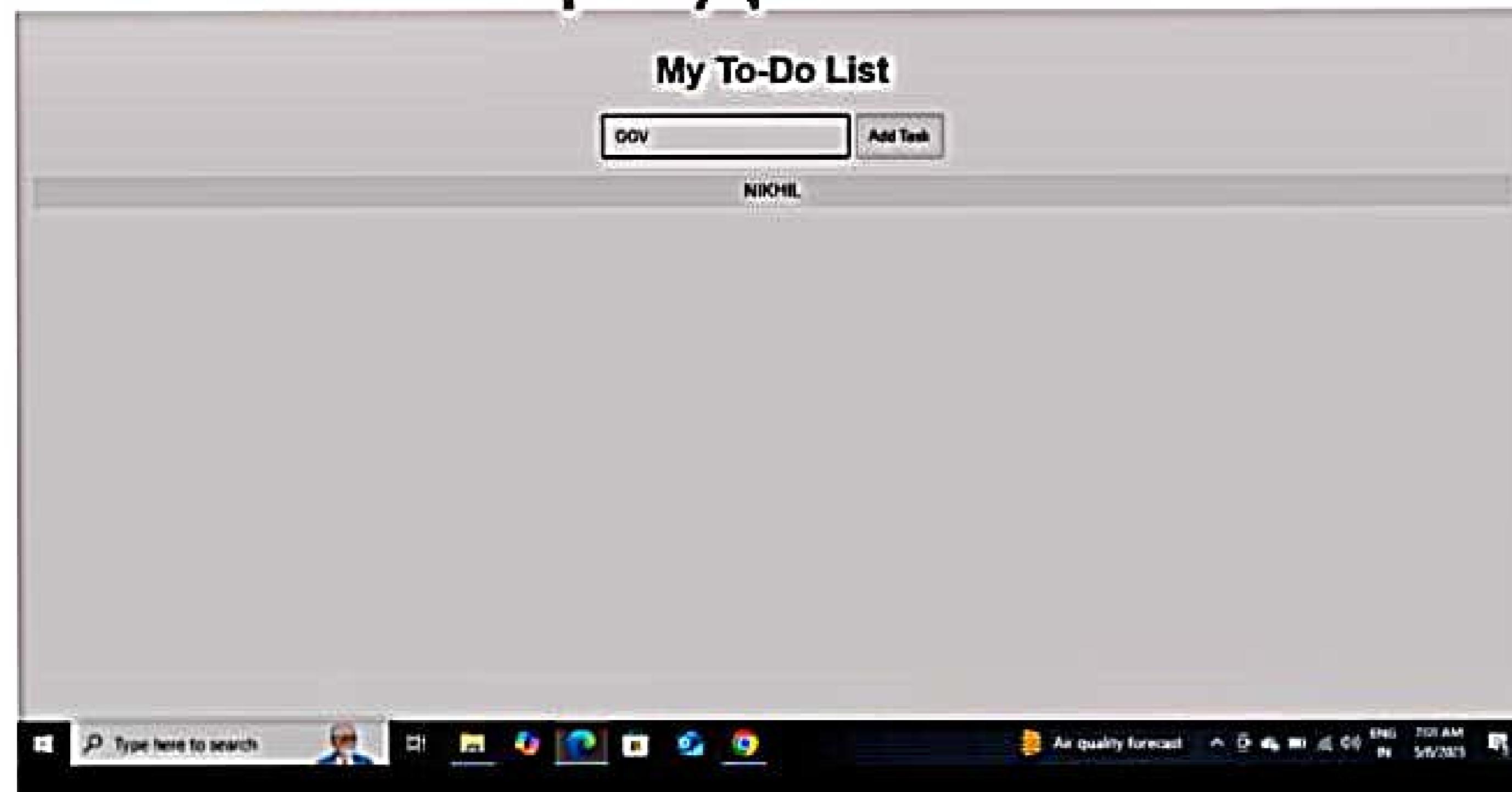
To register for an account, click on the "register" button on the login page. You will be prompted to enter your name, email address, and password. Once you have entered the information, click "submit," and your account will be created.



# Task main page



# Add task page



# Add view page



The screenshot shows a code editor interface with the following details:

- Title Bar:** todo-app
- File Menu:** File Edit Selection View Go ...
- Toolbar:** Includes icons for back, forward, search, and file operations.
- Explorer:** Shows a folder structure under TODO-APP:
  - index.html (selected)
  - script.js
  - style.css
- Code Editor:** Displays the content of index.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6   <title>My To-Do App</title>
7   <link rel="stylesheet" href="style.css" />
8 </head>
9 <body>
10  <h1>My To-Do List</h1>
11  <input type="text" id="taskInput" placeholder="Enter a task" />
12  <button onclick="addTask()">Add Task</button>
13  <ul id="taskList"></ul>
14
15  <script src="script.js"></script>
16 </body>
17 </html>
```
- Terminal:** Shows the command PS C:\Users\DR\OneDrive\Documents\New folder (2)\todo-app>
- Status Bar:** Includes file statistics (Ln 17, Col 8, Spaces 2, UTF-8, CRLF), file type (HTML), and system information (CPU 655 AM, 5/6/2025).

A screenshot of a Windows desktop environment. In the center is a Microsoft Edge browser window titled "todo-app". The address bar shows "PS C:\Users\DR\OneDrive\Documents\View folder (2)\todo-app>". The main content area of the browser displays a simple todo application with a header "Todos", a text input field, and a button labeled "Add". Below the input field is a list of todos: "Buy milk", "Buy bread", and "Buy eggs".

To the left of the browser is a Visual Studio Code editor window. The title bar says "todo-app". The editor has three tabs open: "index.html", "# style.css", and "script.js". The "# style.css" tab is active and contains the following CSS code:

```
# style.css > body
1 body {
2   font-family: Arial, sans-serif;
3   text-align: center;
4   margin-top: 50px;
5 }
6
7 input {
8   padding: 10px;
9   width: 200px;
10 }
11
12 button {
13   padding: 10px;
14 }
15
16 ul {
17   list-style: none;
18   padding: 0;
19 }
```

The status bar at the bottom of the VS Code window shows "Ln 5, Col 4" and "38°C Sunny". The taskbar at the bottom of the screen includes icons for File Explorer, Task View, Edge, File Explorer, Camera, and Task View again.

Edit Selection View Go ... ← → ⌂ todo-app ⌂

EXPLORER  
TODO-APP  
index.html  
# style.css

script.js

```
script.js > addTask > onclick
1 function addTask() {
2     const input = document.getElementById("taskInput");
3     const task = input.value;
4
5     if (task === "") {
6         alert("Please enter a task!");
7         return;
8     }
9
10    const ul = document.getElementById("taskList");
11    const li = document.createElement("li");
12    li.textContent = task;
13
14    // Add delete when clicked
15    li.onclick = function () {
16        ul.removeChild(li);
17    };
18
19    ul.appendChild(li);
20    input.value = ""; // clear input
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\VK\OneDrive\Documents\Vue folder (2)\todo-app>

Ln 16, Col 26 Spaces: 2 UTF-8 CRLF () JavaScript

to search

File Explorer Home Recent View Taskbar