**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

**PNR NO.:------------------- Roll no:---------------------**

**1.Name of the student**

Shraddha Chandrakant Khandare

**2. Title of the Project**

To-Do List

**3. Name of the Guide**

Mrs. Minakshi Dhande and Mrs. Sheetal Vekhande

**4. Is this your first submission? Yes No**

**Signature of the student Signature of the Guide**

**Date: ………………… Date: ………………..**

**Signature of the Coordinator**

**Date: …………………**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TO-DO LIST APPLICATION**

**A Project Report**

Submitted in partial fulfillment of the requirement for the award of the Degree of

**BACHLOR OF SCIENCE (COMPUTER SCIENCE)**

**By**

Miss. Shraddha Chandrakant Khandare

Seat Number

**Under the esteemed guidance of**

**Mrs. Minakshi Dhande**

**Mrs. Sheetal Vekhande**

**Designation**

****

**DEPARTMENT OF COMPUTER SCIENCE**

**MODEL COLLEGE OF SCIENCE AND COMMERC**

**(Affiliated to university of Mumbai)**

**KALYAN-MAHARASHTRA-421306**

**2021-2022**

**MODEL COLLEGE OF SCIENCE AND COMMERCE**

**DEPARTMENT OF COMPUTER SCIENCE**

**(Affiliated to university of Mumbai)**

**KALYAN-MAHARASHTRA-421306**

****

**CERTIFICATE**

This is to certify that the project entitled,”**To-Do List** ”,is bonafied work of “**Shraddha Chandrakant Khandare**” baring seat no;( ) submitted in partial fulfillment of the requirement for the award of degree of BACHLOR OF SCIENCE in COMPUTER SCIENCE from University of Mumbai.

**Internal Guide Coordinator**

**External Examiner**

**Date: College Seal**

**ABSTRACT**

TO-DO List is a project which aims to remind an individual of the planned activities to be don.TO-DO List are the lists that we generally use to maintain our day to day tasks or list of everything that we have to do, with the most important tasks at the top of the list, and the least important tasks at the bottom. It is helpful in planning our daily schedules. We can add more tasks any time and delete a task which is completed. The four major tasks that we can perform in a TODO list are:

**1) Add tasks and save tasks**

**2) Update tasks or Edit Tasks**

**3) Read tasks**

**4) Mark as complete your tasks**

**5) Delete tasks or Delete all list**

Let us see what is there in the application:

* User can add the tasks that are to be done in a descriptive way.
* User will be able to add as many tasks as they have.
* Once the task is completed, user will be able to remove it by clicking on “Task Completed” button.

Following are the user interface components that will be shown in the application-

* There will be an add task button and delete all button on the right side of the application
* There will be a button “Complete,Edit and Delete” for each task added, so user will be able to mark as complete and edit it and delete it.
* The tasks that will be added by the users are stored in the list view.

**FUTURE SCOPE**

* Nowha-days youth glued to their phones and internet all the time most of them have account on social media platform online marketers can easily find audience through the social network.
* Now a days most of everything is online and computerized in this situation people is not able to maintain all the tasks record in a book Or sticky notes Or piece of paper. This computerized system/software Or application will help to maintain the tasks of the user which will reduce to worry about losing sticky notes or piece of paper.
* To-do lists are essential if user are going to beat work overload. When user don’t use them effectively, they will appear unfocused and unreliable to the people around you.
* By laying out everything that they have to do in order, it’s easier to focus on one activity at a time. This reduces “noise” and can lower anxiety about having too much to do.
* A to-do list provides an easy-to-follow structure for their day. You don’t have to worry about interdependencies. They simply go down the list in order. This helps people avoid multitasking and being sidetracked.
* To-do lists allow user to see what they’ve accomplished. Every time you complete a task, they get to check it off, which can help them feel more productive and successful. It can even boost their mood.

**ACKNOWLEDGEMENT**

We thank the almighty for giving us the courage & perseverance in completing the project. This project itself is an acknowledgement for all those who have given us their heart-felt-co-operation in making it a grand success.

We are greatly indebted to Mrs. Minakshi Dhande and Mrs. Sheetal Vekhande, for providing valuable guidance at every stage of this project work.

It is a pleasure to express our deep and sincere gratitude to the project Guide and are profoundly grateful towards the unmatched help rendered by her. Our special thanks to all the lectures of Information Technology, for their valuable advises at every stage of this work.

Last but not the least; we would like to express our deep sense and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the project. We would also like to thank our friends, whose direct or indirect help has enabled us to complete this work successfully.

**Shraddha Chandrakant Khandare**

**DECLARATION**

I hereby declare that the project entitled, **”TO-DO LIST”** done at **MODEL COLLEGE OF SCIENCE AND COMMERCE,** Rajbhernager, chinchpada road , katemanivali Kalyan€,421306,has not been case duplicated to submit to any other university for the award of any degree .To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (COMPUTER SCIENCE)** to be submitted as final semester project as part of our curriculum.

**Shraddha Chandrakant Khandare**

**Name and signature of the student**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Reno.** |  | **Topic** | **Page no.** |
| **1** |  | **CHAPTER-1** |  |
| **1** | **INTRODUCTION** |  |
| 1.1 | What is a ToDo List? | **1** |
| 1.2 | The Benefits of Using a To Do List | **1** |
| 1.3 | What is a ToDo List in business and why is it important? | **2** |
| 1.4 | What Makes a Great To Do List App? | **3** |
| **2** |  | **CHAPTER-2** |  |
| **2** | **OBJECTIVES AND SCOPE OF THE SYSTEM** |  |
| 2.1 | Objectives Of the Proposed System | **4** |
| 2.2 | Scope of the Project | **4** |
| 2.3 | Background of the project | **5** |
|  |  | **CHAPTER-2.1** |  |
|  | **2.1** | **SYSTEM ANALYSIS** |  |
|  | 2.1.1 | Description of System | **6** |
|  | 2.1.2 | Feasibility Study | **7** |
|  | 2.1.3 | Project Analysis | **9** |
|  | 2.2.1 | Hardware and software requirement | **10** |
|  | 2.2.2 | Software Specification | **11** |
|  | 2.2.3 | Features of JavaScript | **12** |
|  | 2.2.4 | Gantt chart | **13** |
| **3** |  | **CHAPTER-3** |  |
| **3** | **PROJECT DESIGN** |  |
| 3.1 | Design Document | **14** |
| 3.2 | Data Flow Diagram | **15** |
| 3.3 | Activity Diagram | **20** |
| 3.4 | UML | **21** |
| 3.5 | Use Case | **22** |
| 3.6 | ERD | **23** |
| **4** |  | **CHAPTER-4** |  |
|  | 4.1 | SOURCE CODE | **25** |
|  | 4.2 | SYSTEM LAYOUT | **37** |
| **5** |  | **CHAPTER-5** |  |
|  | 5 | PROJECT TESTING | **48** |
| **6** |  | **CHAPTER-6** |  |
|  | 6 | CONCLUSION | **49** |
| **7** |  | **CHAPTER-7** |  |
|  | 7 | REFERENCE | **50** |

**CHAPTER-1:**

**INTRODUCTION**

**1.1)What is a ToDo List?**

The definition is a simple one. It’s a list of tasks user need to complete, or things that user want to do.

Most typically, they’re organised in order of priority. Traditionally, they’re written on a piece of paper or post it notes and acts as a memory aid. As technology has evolved we have been able to create a todo lists with excel spreadsheets, word documents, email lists, todo list apps, microsoft to do and google to do list to name a few. User can use a to do list in their home and personal life, or in the workplace.

Having a list of everything user need to do written down in one place means they shouldn’t forget anything important. By prioritising the tasks in the list they plan the order in which they’re going to do them and can quickly see what needs their immediate attention and what tasks they can leave until a little later.

**1.2)The Benefits of Using a To Do List**

One of the most important reasons user should use a to do list is that it will help them stay organised. When they write all their tasks in a list, they seem more manageable. When they’ve got a clear outline of the tasks they’ve got to do and those they’ve completed, it helps them stay focused. While freeing up space in their mind for other more creative tasks.

When user complete a task, they can cross it off their list. This gives them a sense of progress and achievement, something they will lack if they're always rushing from one task to the next. If user feel a sense of achievement, it spurs them on and motivates them to keep moving forward.

But that’s not the only benefits of a to do list. Here are a few more:

**Improves user's memory:** A to do list acts as an external memory aid. It’s only possible to hold a few pieces of information at one time. Keep a to do list and they'll be able to keep track of everything, rather than just a few of the tasks user need to do. User’s to do list will also reinforce the information, which makes it less likely they are going to forget something.

**Increases productivity:** A to do list allows user to prioritize the tasks that are more important. This means user don’t waste time on tasks that don’t require their immediate attention. Their list will help them stay focused on the tasks that are the most important.

**Helps with motivation:** To do lists are a great motivational tool because user can use them to clarify their goals. User can divide their long-term goal into smaller, more achievable short-term goals and as user tick each one off their list, their confidence will increase.

**1.3)What is a ToDo List in business and why is it important?**

It seem such a simple solution by putting pen to paper and taking time out of their day to create a to do list, a plan for their day helps define their challenges and goals. Preventing time being wasted trying to identify what is the next most important task to tackle next and even more important makes sure user don’t forget to do something important.

To-do lists offer a way to increase productivity, stopping user forgetting things, helps prioritise task, manage tasks effectively, use time wisely and improve time management as well as workflow.

**1.4)What Makes a Great To Do List App?**

* + Tasks should be fast to add and organise
  + Ability to plan user's workflow
  + Setting priorities
  + Allocation of tasks if using it for task management with a team. Team to-do list allows user to assign to the best person for the job.
  + Able to synch between different platforms
  + All it takes is just a few minutes every day to keep a to do list up to date. With a to do list, user can complete goals without wasting time trying to figure out priorities. Their productivity will increase, user won’t forget things, their time management will improve and they will be able to manage their tasks more effectively.

**CHAPTER-2:**

**OBJECTIVE AND SCOPE OF THE SYSTEM**

**2.1) OBJECTIVES OF THE PROPOSED SYSTEM**

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

* Add tasks
* Update tasks or Edit Tasks
* Display and Read tasks
* Mark as complete user’s tasks
* Delete tasks or Delete all list

**2.2) SCOPE OF THE PROJECT:**

TO-DO List application uses an JavaScript local storage to store tasks records of user in list.

To start with, it is better to mention that a typical to-do list (also called as a list of things, a list of errands, things-to-do list, etc.) is just a list of tasks for a certain period of time, a day or a week.

Initially, to-do list was created for the purpose to be a well-structured work plan helps every person to see his current load and execute all the necessary tasks. Now it becomes a common practice almost for everyone, and it solves many problems concerning the high workload.

**To-do List Has Its Following Functions:**

* Organization of a proper work with an accurate planning of every task with its short description
* Simplify the workflow and a timely execution of necessary work scope
* Improvement of every employee’s productivity and time management skills

**2.3)BACKGROUND OF PROJECT**

TO-DO LIST is an application which refers to maintain all the tasks records in a computerized application or system. To-do lists are evil from a productivity perspective, it’s much more effective to schedule time.

To-do list software is designed to make the process of task management easy. These are cloud-based applications that typically offer mobile and tablet (iOS and Android) apps for quick access to user's tasks.The key benefit of this software is that there’s no learning curve.

A to-do list is simply a list of all of the tasks user need to complete during a particular period of time. For instance, at the beginning of the day, they might jot down on a sticky note all of the tasks they want to complete before the end of the day. That becomes their to-do list.

With to-do list functionality built into their project management software, user can easily create, manage, edit, and share their lists. Plus, user don’t have to worry about losing sticky notes or pieces of paper. If user chosen software has a mobile application, user can keep the list with them wherever they go.

By adding the most important tasks at the top of the list, and the least important tasks at the bottom of the list. It is helpful in planning their daily schedules. User can add more tasks any time and delete a task which is completed. The four major tasks that we can perform in a TODO list are:

* Add tasks & save tasks
* Update tasks or Edit Tasks
* Read tasks
* Mark as complete user's tasks
* Delete tasks or Delete all list

**2.1: SYSTEM ANALYSIS**

**2.1.1) DESCRIPTION OF PROPOSED SYSTEM:**

As the name specifies "**TO-DO LIST**"is a software developed to add tasks in the task list

As the strength of the working people is increasing it is not able to maintain the all the tasks records in a book Or sticky notes or piece of paper. This computerized system/software or application will help to maintain the tasks of the user which will reduce to worry about losing sticky notes or piece of paper.

This particular project deals with the problems on managing the records of tasks.Having a list of everything user need to do written down in one place means user shouldn’t forget anything important. By prioritising the tasks in the list user plan the order in which user were going to do them and can quickly see what needs their immediate attention and what tasks user can leave until a little later.

One of the most important reasons user should use a to do list is that it will help him/her stay organised. When user write all their tasks in a list, they seem more manageable. When user have got a clear outline of the tasks user have got to do and those user have completed, it helps user to stay focused. While freeing up space in their mind for other more creative tasks.

To-Do List project is an application specially built to keep track of errands or tasks that need to be done. This application will be like a task keeper where the user would be able to enter the tasks that they need to do. Once they are done with their tasks they can also remove them from the list.

Following are the user interface components that will be shown in my application-

* There will be an add task button and delete all button on the right side of the application.
* There will be a button “Complete,Edit and Delete” for each task added, so user will be able to mark as complete and edit it and delete it.
* The tasks that will be added by the users are stored in the list view.

**2.1.2) FEASIBILITY STUDY**

**Feasibility:** A measure of how beneficial or practical the development of information system would be to an organization can be termed as Project Feasibility.

* Feasibility should be measured throughout the life cycle. The scope and complexity of an apparently feasible project can change after the initial problems and opportunities are fully analyzed and after the system has been designed. Thus, a project that is feasible at one point may become infeasible later.
* The check points for feasibility study are: -
* A survey phase checkpoint
* A study phase checkpoint.
* A definition phase checkpoint
* A selection phase checkpoint.
* Generally there are following areas of risk for a new system that are considered when confirming project feasibility:

1. **Technical Feasibility**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn’t want to try to put Star Trek’s transporters in their building—currently, this project is not technically feasible.

1. **Economic Feasibility**

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

1. **Legal Feasibility**

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let’s say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization’s ideal location isn’t zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

1. **Operational Feasibility**

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development

1. **Scheduling Feasibility**

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

* **Internal Project Constraints:** Technical, Technology, Budget, Resource, etc.
* **Internal Corporate Constraints:** Financial, Marketing, Export, etc.
* **External Constraints:** Logistics, Environment, Laws, and Regulations, etc

**2.1.3) PROJECT ANALYSIS**

**Existing System:-**

Present system is manual. Many people find to-do lists helpful for prioritizing and completing tasks. There are three reasons for this:

* + By laying out everything that user have to do in order, it’s easier to focus on one activity at a time. This reduces “noise” and can lower anxiety about having too much to do.
  + A to-do list provides an easy-to-follow structure for user’s day. User don’t have to worry about interdependencies. User simply go down the list in order. This helps people avoid multitasking and being sidetracked.
  + To-do lists allow user to see what they’ve accomplished. Every time user complete a task, user get to check it off, which can help user feel more productive and successful. It can even boost their mood.

**2.2.1)SYSTEM AND DESIGN USER REQUIREMENT**

**DEVELOPMENT TOOL USED**

**SOFTWARE CONFIGURATION**

**OPERATING PLATFORM : REDMI NOTE 9 PRO**

**BACK END TOOL(Data storage) : JAVASCRIPT (ES6) VERSION**

**SOFTWARE : TREB EDIT**

**FRONT END TOOL : HTML & CSS**

**HARDWARE CONFIGURATION**

**RAM : 4GB**

**HARD DISK(STORAGE) : 16 GB**

**(MEMORY CARD)**

**LOCAL STORAGE : 128GB**

**2.2.2) SOFTWARE SPECIFICATIONS:**

**HTML:-**

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as “this is a heading”, “this is a paragraph”, “this is a link”, etc.

**JAVASCRIPT:-** ECMAScript 2015 commonly known as ES6 or ES2015 released on June 2015. ES5 was released on December 2009. It would then take almost six years for the next version of ECMAScript to be released. So, there are many exciting feature in ES6.

Javascript ES6 has been around for few years now, and it allows us to write code in a clever way which basically makes the code more modern and more readable.It’s fair to say that with the use of ES6 features we write less and do more, hence the term ‘write less, do more’ definitely suits ES6.

ES6 introduced several key features like const, let, arrow functions, template literals, default parameters and lot more.Let’s take a look at them one by one.

“const” and “let”

Before ES6 we mainly made use of the var keyword whenever we wanted to declare a variable.But it had some serious issues, also it was not developers favorite so in the ES6 version we were introduced to const and let keywords which allows us to store variables.They both have their own way to storing variables.

**2.2.3) FEATURES:**

**A) FEATURES OF JAVASCRIPT (ES6) VERSION:**

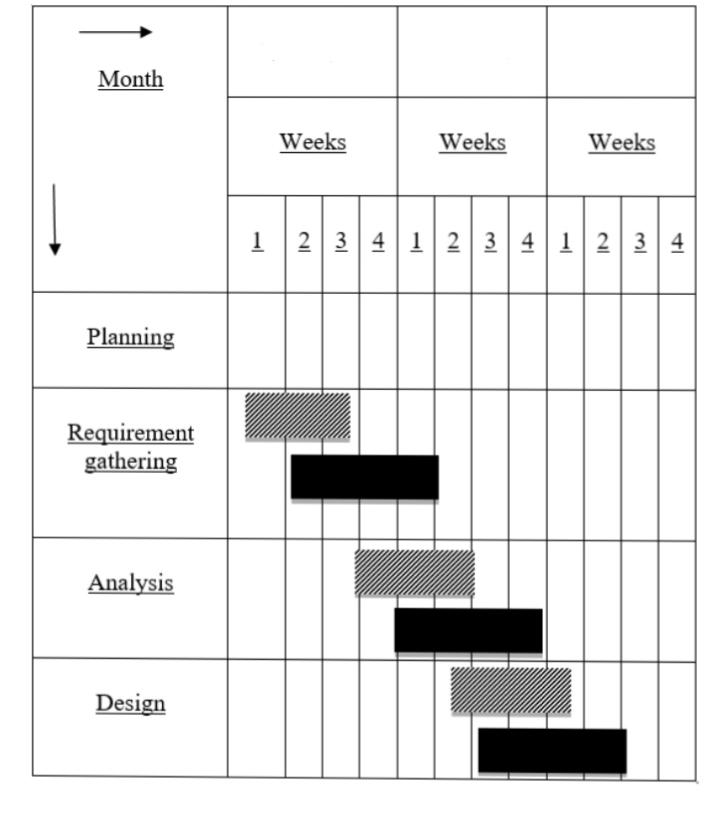
Here’s the list of the top 10 best ES6 features for a busy javascript developer(in no particular order):

1. Default Parameters in ES6
2. Template Literals in ES6
3. Multi-line Strings in ES6
4. Destructuring Assignment in ES6
5. Enhanced Object Literals in ES6
6. Arrow Functions in ES6
7. Promises in ES6
8. Block-Scoped Constructs Let and Const
9. Classes in ES6
10. Modules in ES6

**B) FEATURES OF HTML 5:**

1. Intro of audio and video 6) Progress Tag
2. Vector Graphics 7) Placeholder Attribute
3. Header and Footer 8) Email Attribute
4. Figure and Figcaption 9) Storage
5. Nav tag 10) Ease of Use

**2.2.4) GANTT CHART:-**



**September**

**July**

**August**

**CHAPTER-3:**

**PROJECT DESIGNIN**

**3.1)DESIGN DOCUMENT**

* + The entire system is projected with a physical diagram which specifics the actual storage parameters that are physically necessary for any database to be stored on to the disk. The overall systems existential idea is derived from this diagram.
  + The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
  + The content level DFD is provided to have an idea of the functional inputs and outputs that are achieved through the system. The system depicts the input and out put standards at the high level of the systems existence.

**3.2) DATA FLOW DIAGRAMS:**

Data flows are data structures in motion, while data stores are data structures. Data flows are paths or ‘pipe lines’, along which data structures travel, where as the data stores are place where data structures are kept until needed.

Data flows are data structures in motion, while data stores are data structures at rest. Hence it is possible that the data flow and the data store would be made up of the same data structure.

Data flow diagrams is a very handy tool for the system analyst because it gives the analyst the overall picture of the system, it is a diagrammatic approach.

A DFD is a pictorial representation of the path which data takes From its initial interaction with the existing system until it completes any interaction. The diagram will describe the logical data flows dealing the movements of any physical items. The DFD also gives the insight into the data that is used in the system i.e., who actually uses it is temporarily stored.

A DFD does not show a sequence of steps. A DFD only shows what the different process in a system is and what data flows between them.

The following are some DFD symbols used in the project

External entities

Process: A transaction of information that resides within the bounds of the system to be module.

DATAFLOWS

Datastore: A repository of data that is to be stored for use by one or more processes, may be as simple as buffer of queue or as a relational database.

**RULES FOR DFD:**

* Fix the scope of the system by means of context diagrams.
* Organize the DFD so that the main sequence of the actions reads left to right and top to bottom.
* Identify all inputs and outputs.
* Identify and label each process internal to the system with rounded circles process is required for all the data transformation and transfers. Therefore, never Connect adata store to a data source or the destinations or another data store with just a data flow arrow.
* Do not indicate hardware and ignore control information.
* Make sure the names of the processes accurately convey everything the process is done.
* There must not be unnamed process.
* Indicate external sources and destinations of the data, with squares.
* Number each occurrence of repeated external entities.
* Identify all data flows for each process step, except simple Record retrievals.
* Label data flow on each arrow.
* Use details flow on each arrow.
* Use the details flow arrow to indicate data movements.
* There can’t be unnamed data flow.
* A data flow can’t connect two external entities.

**LEVELS OF DFD:**

The complexity of the business system means that it is a responsible to represent the operations of any system of single data flow diagram. At the top level, an Overview of the different systems in an organization is shown by the way of context analysis diagram. When exploded into DFD

They are represented by:

* LEVEL-0 : SYSTEM INPUT/OUTPUT
* LEVEL-1:SUBSYSTEM LEVEL DATAFLOW FUNCTIONAL
* LEVEL-2 : FILE LEVEL DETAIL DATA FLOW.

The input and output data shown should be consistent from one level to the next.

**LEVEL-0: SYSTEM INPUT/OUTPUT LEVEL**

A level-0 DFD describes the system-wide boundaries, dealing inputs to and outputs from the system and major processes. This diagram is similar to the combined user-level context diagram.

**LEVEL-1: SUBSYSTEM LEVEL DATA FLOW**

A level-1 DFD describes the next level of details within the system, detailing the data flows between subsystems, which makeup the whole.

**LEVEL-2: FILE LEVEL DETAIL DATA FLOW**

All the projects are feasible given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Feasibility and the risk analysis are pertained in many ways. If project risk is great.

### 3.2.1) ZERO LEVEL DATAFLOW DIAGRAM

**DELETE TASK**

## ADD TASKS

**MARK AS COMPLETE**

## UPDATE/EDIT TASKS

## DELETE ALL Tasks

**Search Tasks**

**2nd Level DFDs:**

### DFD for User:

**User**

**User**

**Start App**

**Create the**

**Tasks list**

**Provide List**

**Most Important at the top & least important at bottom**

**Add Task**

**User**

**3.3) ACTIVITY DIAGRAM:**



Add Tasks



Save and Display Tasks

Mark as complete Tasks

Save Edited Tasks

Update or Edit Tasks

Search Tasks

Delete Specific Tasks



Delete All Tasks List

Show Confirm box while deleting



Exist



**3.4) Unified Modeling Language Diagrams(UML):**

* + The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.
  + A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

**User Model View**

* + 1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective**.**

**Structural model view**

◆In this model the data and functionality are arrived from inside the system.

◆ This model view models the static structures.

**Behavioral Model View**

◆ It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**Implementation Model View**

* + In this the structural and behavioral as parts of the system are represented as they are to be built.

**Environmental Model View**

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.UML is specifically constructed through two different domains they are

* + UML Analysis modeling, which focuses on the user model and structural model views of the system.

**3.5) USE CASE DIAGRAMS**

**User:** In the User Module**,** user Add Tasks, Update Tasks, search Tasks, Delete Tasks.

****

**User**

#### 3.6) ENTITY-RELATIONSHIP DIGRAMS (ERD)

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

## The symbols used in E-R diagrams are:

SYMBOL PURPOSE

Represents Entity sets.

Represent attributes.

Represent Relationship Sets.

Line represents flow

Structured analysis is a set of tools and techniques that the analyst.

To develop a new kind of a system:

The traditional approach focuses on the cost benefit and feasibility analysis, Project management, and hardware and software selection an personal considerations.

**USER*:***

**TASKS**

**USER**

**Exit**

**CHAPTER-4:**

**SOURCE CODE AND SYSTEM LAYOUT**

**4.1) SOURCE CODE:**

**1)Index.html**

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”UTF-8” />

<meta name=”viewport” content=”width=device-width, initial-scale=1.0” />

<title>Todo List</title>

<link rel=”stylesheet” href=”font-awesome1.min.css” />

<link rel=”stylesheet” href=”bootstrap1.min.css” />

<link rel=”stylesheet” href=<https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css>>

<link rel=”stylesheet” href=”style1.css” />

</head>

<body>

<header class=”bg-secondary”style=”background:linear-gradient(#f800ff,#2bd5ff)”>

<div class=”container”>

<div class=”row”>

<div class=”col-sm-12”>

<!—show Current date and time -- >

<body bgcolor=”white”>

<!-- <marquee> -->

<p class=”DTbox”><i> <b>Date & Time</b>: <span id=’date-time’></span>.</i></p>

<!-- </marquee> -- >

</body>

<!—opening of script in javascript -- >

<script>

Var dt = new Date();

Document.getElementById(‘date-time’).innerHTML=dt;

</script>

<!—Navigation bar -- >

<nav class=”navbar justify-content-between”>

<a class=”navbar-brand”>Todo List</a>

<form class=”form-inline”>

<input class=”form-control mr-sm-2” type=”search” placeholder=”Search” aria-label=”Search”

Id=”searchtextbox” />

</form>

</nav>

</div>

</div>

</div>

</header>

<section class=”todo-outer”>

<div class=”container”>

<div class=”row justify-content-md-center”>

<div class=”col-sm-12 col-md-8”>

<h1 align=”center” style=”color:#0057ff”>Welcome in Todo List Application</h1>

<div class=”todo-inner”>

<div class=”form-row”>

<div class=”col-8 mr-sm-2”>

<input type=”text” class=”form-control” placeholder=”Enter your task” id=”addtaskinput” />

<input type=”hidden” id=”saveindex”>

</div>

<button type=”button” class=”btn btn-success mr-sm-2” id=”addtaskbtn”>

Add Task

</button>

<button type=”button” class=”btn btn-success mr-sm-2” id=”savetaskbtn” style=”display: none;”>

Save Task

</button>

<button type=”button” id=”deleteallbtn” class=”btn btn-danger”>

Delete All

</button>

</div>

<div class=”to-do-output”>

<table class=”table table-striped mt-3 mb-0” id=”addedtasklist”>

</table>

</div>

</div>

</div>

</div>

</div>

</section>

<script src=”main1.js”></script>

</body>

</html>

**2) Style.css**

@font-face {

Font-family: ‘Roboto-Regular’;

Src: url(‘../fonts/Roboto-Regular.eot’);

Src: local(‘☺’), url(‘../fonts/Roboto-Regular.woff’) format(‘woff’), url(‘../fonts/Roboto-Regular.ttf’) format(‘truetype’), url(‘../fonts/Roboto-Regular.svg’) format(‘svg’);

Font-weight: normal;

Font-style: normal;

}

@font-face {

Font-family: ‘FontAwesome’;

Src: url(‘../fonts/fontawesome-webfont.eot’);

Src: local(‘☺’), url(‘../fonts/fontawesome-webfont.woff’) format(‘woff’), url(‘../fonts/fontawesome-webfont.ttf’) format(‘truetype’), url(‘../fonts/fontawesome-webfont.svg’) format(‘svg’);

Font-weight: normal;

Font-style: normal;

}

Body{font-family: ‘Roboto-Regular’;background:#f5f5f5;}

Header .navbar{padding: .5rem 0;margin-bottom:10px;}

Header nav.navbar .navbar-brand{color:#ffffff;}

Section.todo-outer {

Width: 100%;

Float: left;

Padding: 30px 0;

}

Section.todo-outer h1 {

Font-size: 40px;

Margin-bottom: 20px;

}

.todo-inner {

Width: 100%;

Float: left;

Padding: 40px;

Box-shadow: 0px 0px 10px #c1c1c1;

/\*background:#fff;\*/

Background:linear-gradient(140deg,#9180ff ,#ffffff ,#ff80dd);

/\*background: url(“”)\*/;

}

Table tr td button {

Background: no-repeat;

Border: none;

Padding: 0;

Cursor: pointer;

}

.completed{text-decoration: line-through;text-decoration-color:#097e00;text-decoration-style:double;}

Font: ;

/\* Media Queries: Tablet Landscape \*/

@media screen and (max-width: 1060px) {

#primary { width:1678%; }

#secondary { width:1300%; margin-left:63%;}

}

/\* Media Queries: Tabled Portrait \*/

@media screen and (max-width: 768px) {

#primary { width:15000%; }

#secondary { width:1000%; margin:0; border:none; }

}

.DTbox{background-color:#161616;

Background:linear-gradient(139deg,#ffaaf4 ,#ffffff ,#ffaaf4);

Color:#0003e2;

Border:double;

Border-color:#0018ff;

Margin-top:10px;}

**3) bootstrap.min.css**

<https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css>

**4) Font-awesome.min.css**

<https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css>

**5) bootstrap-icons.css**

<https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css>

**6) main.js**

Showtask();

Let addtaskinput = document.getElementById(“addtaskinput”);

Let addtaskbtn = document.getElementById(“addtaskbtn”);

Addtaskbtn.addEventListener(“click”, function(){

Addtaskinputval = addtaskinput.value;

If(addtaskinputval.trim()!=0){

Let webtask = localStorage.getItem(“localtask”);

If(webtask == null){

taskObj = [];

}

Else{

taskObj = JSON.parse(webtask);

}

taskObj.push({‘task\_name’:addtaskinputval, ‘completeStatus’:false});

// console.log(taskObj, ‘Ashendra’);

localStorage.setItem(“localtask”, JSON.stringify(taskObj));

addtaskinput.value = ‘’;

}

Showtask();

});

// showtask

Function showtask(){

Let webtask = localStorage.getItem(“localtask”);

If(webtask == null){

taskObj = [];

}

Else{

taskObj = JSON.parse(webtask);

}

Let html = ‘’;

Let addedtasklist = document.getElementById(“addedtasklist”);

taskObj.forEach((item, index) => {

if(item.completeStatus==true){

taskCompleteValue = `<td class=”completed”>${item.task\_name}</td>`;

}else{

taskCompleteValue = `<td>${item.task\_name}</td>`;

}

Html += `<tr>

<th scope=”row”>${index+1}</th>

${taskCompleteValue}

<td><button type=”button” onclick=”edittask(${index})” class=”text-primary”><I class=”bi bi-pencil-square blue”></i>Edit</button></td>

<td><button type=”button” class=”text-success” id=${index}><I class=”bi bi-check-circle-fill”></i>Complete</button></td>

<td><button type=”button” onclick=”deleteitem(${index})” class=”text-danger”><I class=”bi bi-trash-fill”></i>Delete</button></td>

</tr>`;

});

Addedtasklist.innerHTML = html;

}

// edittask

Function edittask(index){

Let saveindex = document.getElementById(“saveindex”);

Let addtaskbtn = document.getElementById(“addtaskbtn”);

Let savetaskbtn = document.getElementById(“savetaskbtn”);

Saveindex.value = index;

Let webtask = localStorage.getItem(“localtask”);

Let taskObj = JSON.parse(webtask);

Addtaskinput.value = taskObj[index][‘task\_name’];

Addtaskbtn.style.display=”none”;

Savetaskbtn.style.display=”block”;

}

// savetask

Let savetaskbtn = document.getElementById(“savetaskbtn”);

Savetaskbtn.addEventListener(“click”, function(){

Let addtaskbtn = document.getElementById(“addtaskbtn”);

Let webtask = localStorage.getItem(“localtask”);

Let taskObj = JSON.parse(webtask);

Let saveindex = document.getElementById(“saveindex”).value;

For (keys in taskObj[saveindex]) {

If(keys == ‘task\_name’){

taskObj[saveindex].task\_name = addtaskinput.value;

}

}

// taskObj[saveindex] = {‘task\_name’:addtaskinput.value, ‘completeStatus’:false} ;

// taskObj[saveindex][task\_name] = addtaskinput.value;

Savetaskbtn.style.display=”none”;

Addtaskbtn.style.display=”block”;

localStorage.setItem(“localtask”, JSON.stringify(taskObj));

addtaskinput.value=’’;

showtask();

})

// deleteitem

Function deleteitem(index){

Let webtask = localStorage.getItem(“localtask”);

Let taskObj = JSON.parse(webtask);

taskObj.splice(index, 1);

if (confirm(“Are you sure want to delete this item from list?”))

localStorage.setItem(“localtask”, JSON.stringify(taskObj));

showtask();

}

//complete task

/\* function completetask(index){

Let webtask = localStorage.getItem(“localtask”);

Let taskObj = JSON.parse(webtask);

taskObj[index] = ‘<span style=”text-decoration:line-through”>’ + taskObj[index] + ‘</span>’;

let addedtasklist = document.getElementById(“addedtasklist”);

addedtasklist.addEventListener(“click”, function€{

console.log(addedtasklist)

})

localStorage.setItem(“localtask”, JSON.stringify(taskObj));

showtask();

} \*/

// complete task

Let addedtasklist = document.getElementById(“addedtasklist”);

Addedtasklist.addEventListener(“click”, function€{

// console.log€;

// showtask();

Let webtask = localStorage.getItem(“localtask”);

Let taskObj = JSON.parse(webtask);

Let mytarget = e.target;

If(mytarget.classList[0] === ‘text-success’){

Let mytargetid = mytarget.getAttribute(“id”);

// let taskValue = taskObj[mytargetid][‘task\_name’];

Mytargetpresibling = mytarget.parentElement.previousElementSibling.previousElementSibling;

// let mynewelem = mytargetpresibling.classList.toggle(“completed”);

// taskObj.splice(mytargetid,1,mynewelem);

For (keys in taskObj[mytargetid]) {

If(keys == ‘completeStatus’ && taskObj[mytargetid][keys]==true){

taskObj[mytargetid].completeStatus = false;

// taskObj[mytargetid] = {‘task\_name’:taskValue, ‘completeStatus’:false};

}else if(keys == ‘completeStatus’ && taskObj[mytargetid][keys]==false){

taskObj[mytargetid].completeStatus = true;

//taskObj[mytargetid] = {‘task\_name’:taskValue, ‘completeStatus’:true};

}

}

//}

// showtask();

localStorage.setItem(“localtask”, JSON.stringify(taskObj));

showtask();

}

})

// deleteall

Let deleteallbtn = document.getElementById(“deleteallbtn”);

Deleteallbtn.addEventListener(“click”, function€{

Let savetaskbtn = document.getElementById(“savetaskbtn”);

Let addtaskbtn = document.getElementById(“addtaskbtn”);

Let webtask = localStorage.getItem(“localtask”);

Let taskObj = JSON.parse(webtask); e.preventDefault();

If (confirm(“Are you sure want to delete all list?”))

If(webtask == null){

taskObj = [];

}

Else{

taskObj = JSON.parse(webtask);

taskObj = [];

}

Savetaskbtn.style.display=”none”;

Addtaskbtn.style.display=”block”;

localStorage.setItem(“localtask”, JSON.stringify(taskObj));

showtask();

})

// serachlist

Let searchtextbox = document.getElementById(“searchtextbox”);

Searchtextbox.addEventListener(“input”, function(){

Let trlist = document.querySelectorAll(“tr”);

Array.from(trlist).forEach(function(item){

Let searchedtext = item.getElementsByTagName(“td”)[0].innerText;

Let searchtextboxval = searchtextbox.value;

Let re = new RegExp(searchtextboxval, ‘gi’);

If(searchedtext.match(re)){

Item.style.display=”table-row”;

}

Else{

Item.style.display=”none”;

}

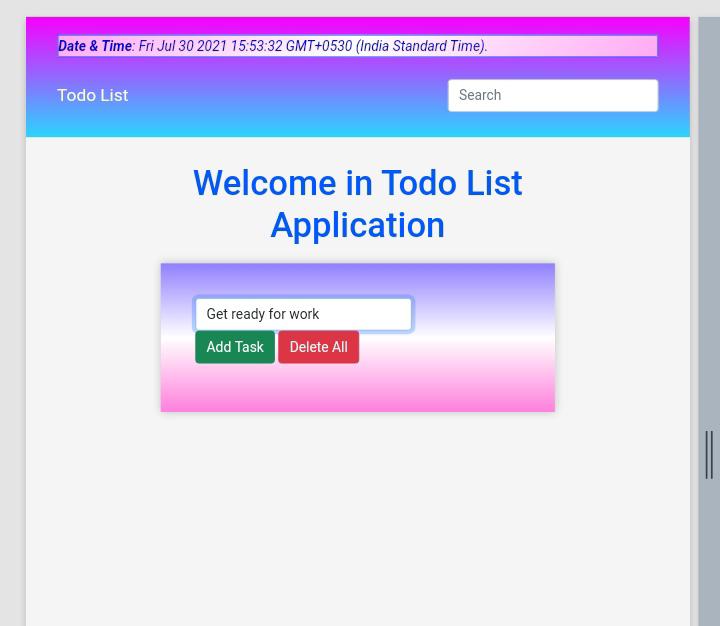
})

})

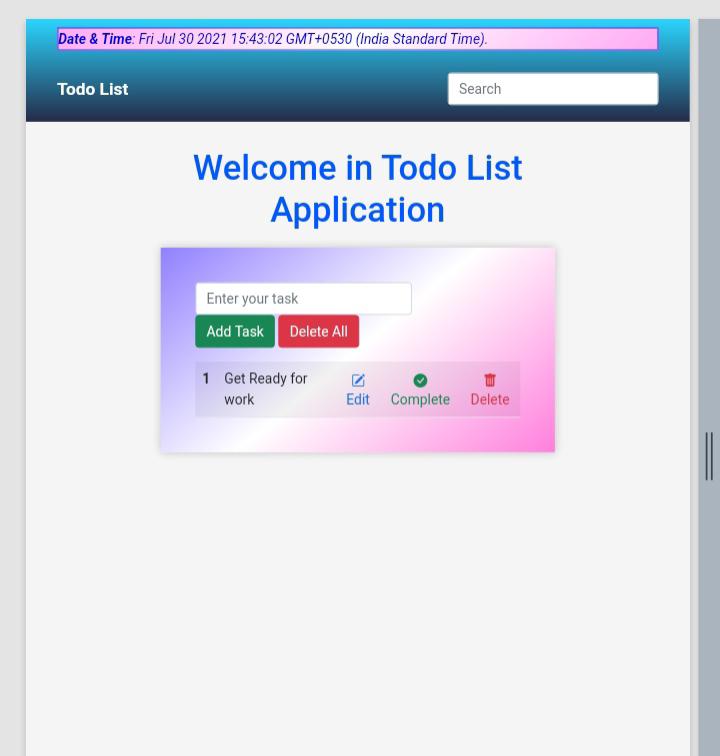
**4.2) SYSTEM LAYOUT**

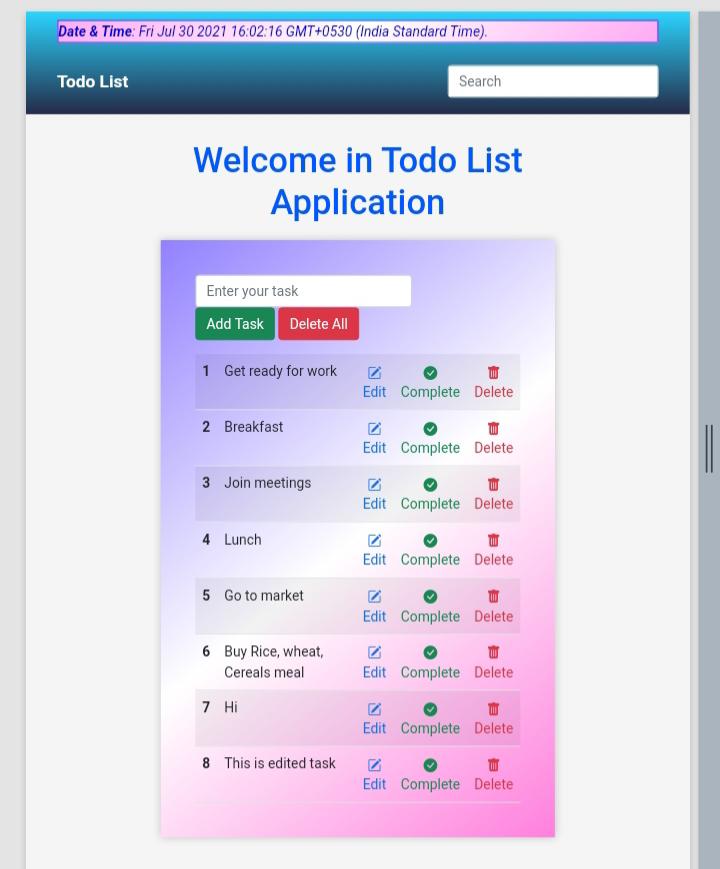
**OUTPUT / RESULT**

**1.Add Tasks**

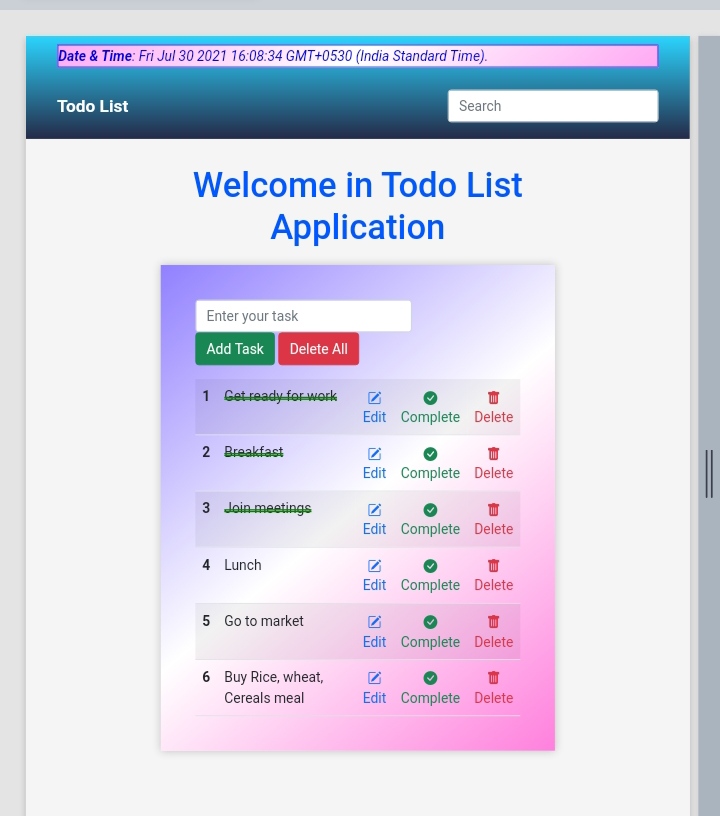


**2.Display Task**



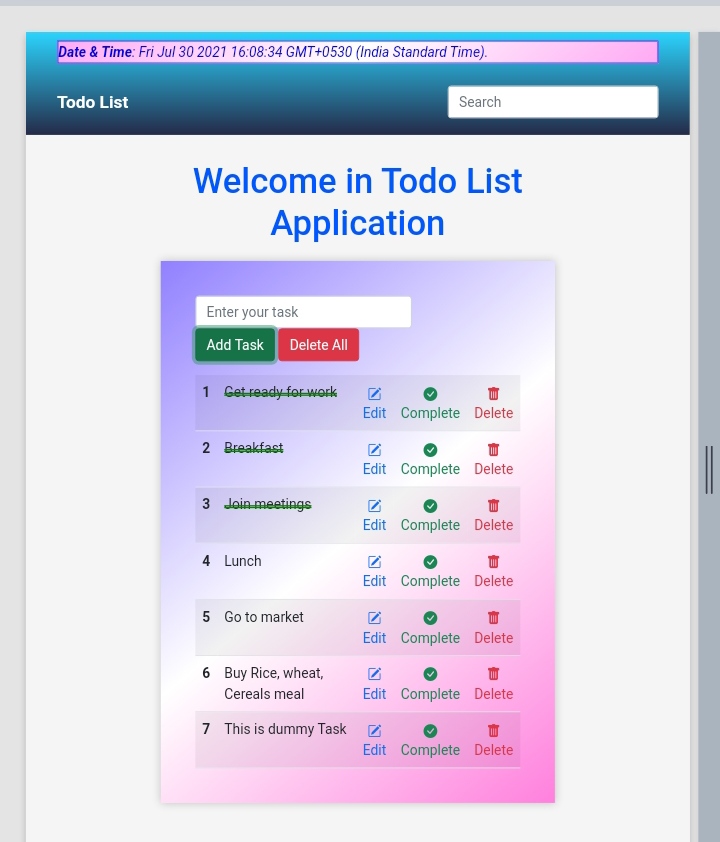
**3.Show Added list**

**4.Mark as complete task**

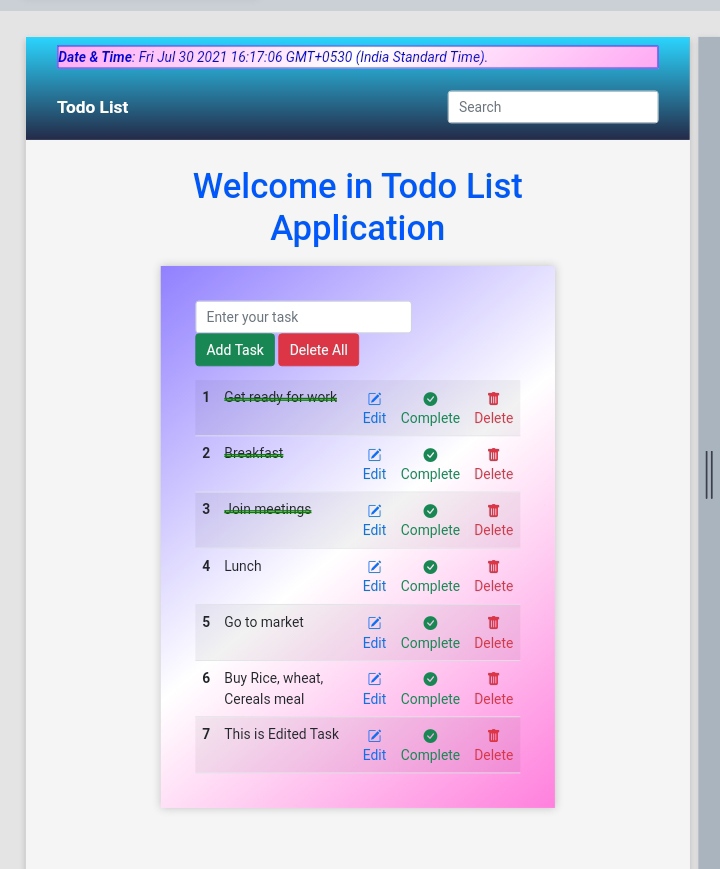


**5.Update Task Or Edit Task**

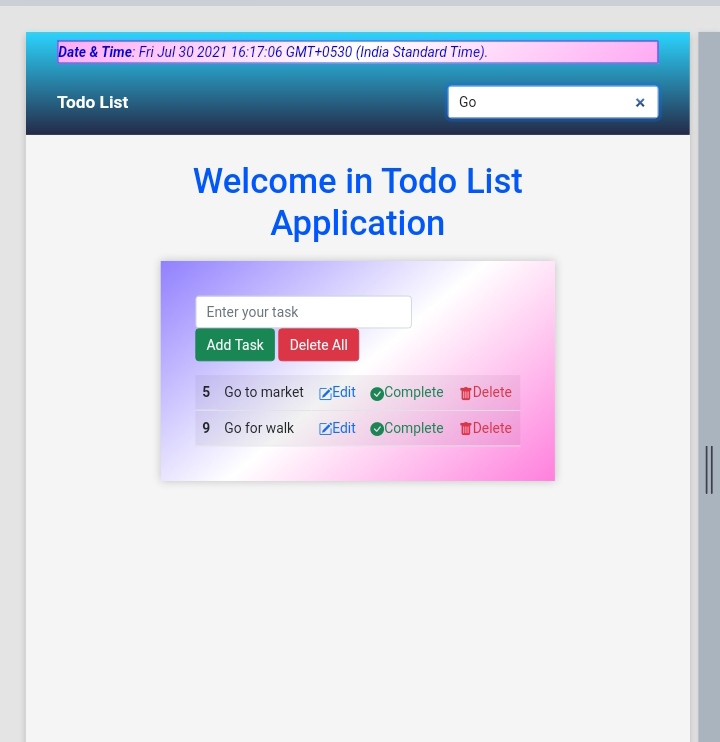
**5.1) Add Dummy Task**

****

**5.2) Edited Dummy Task**

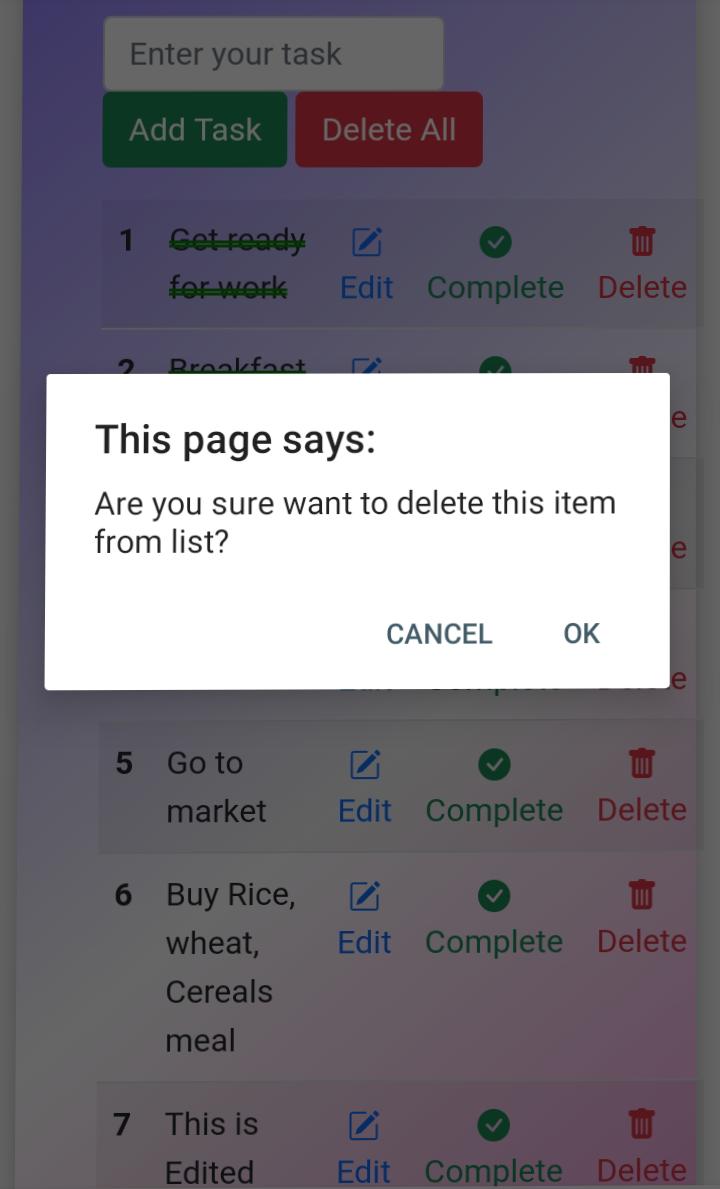
****

**6.Search Tasks**

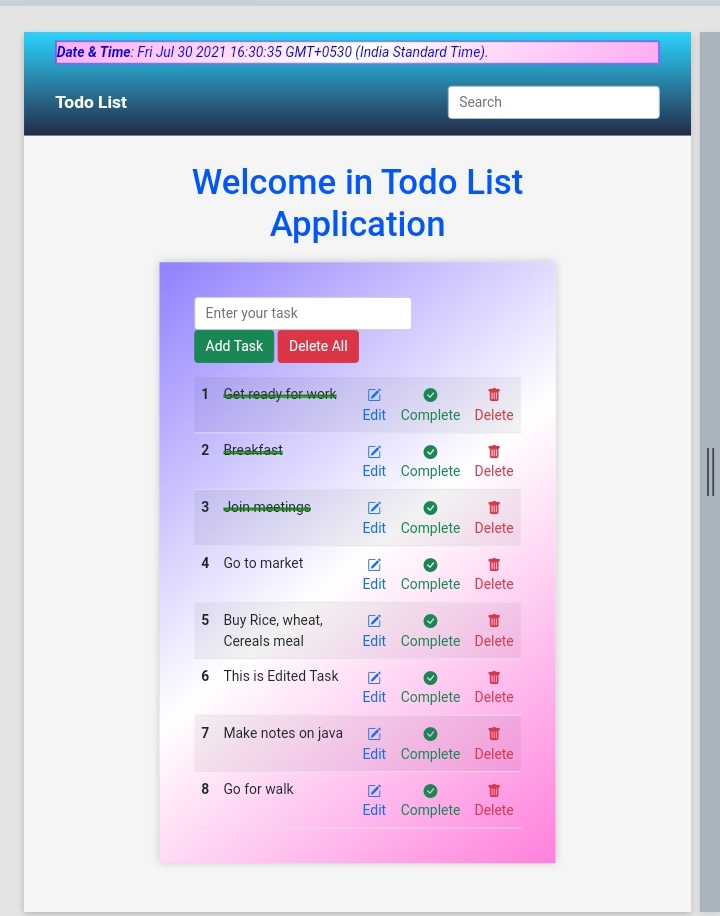


**6.Delete Tasks**

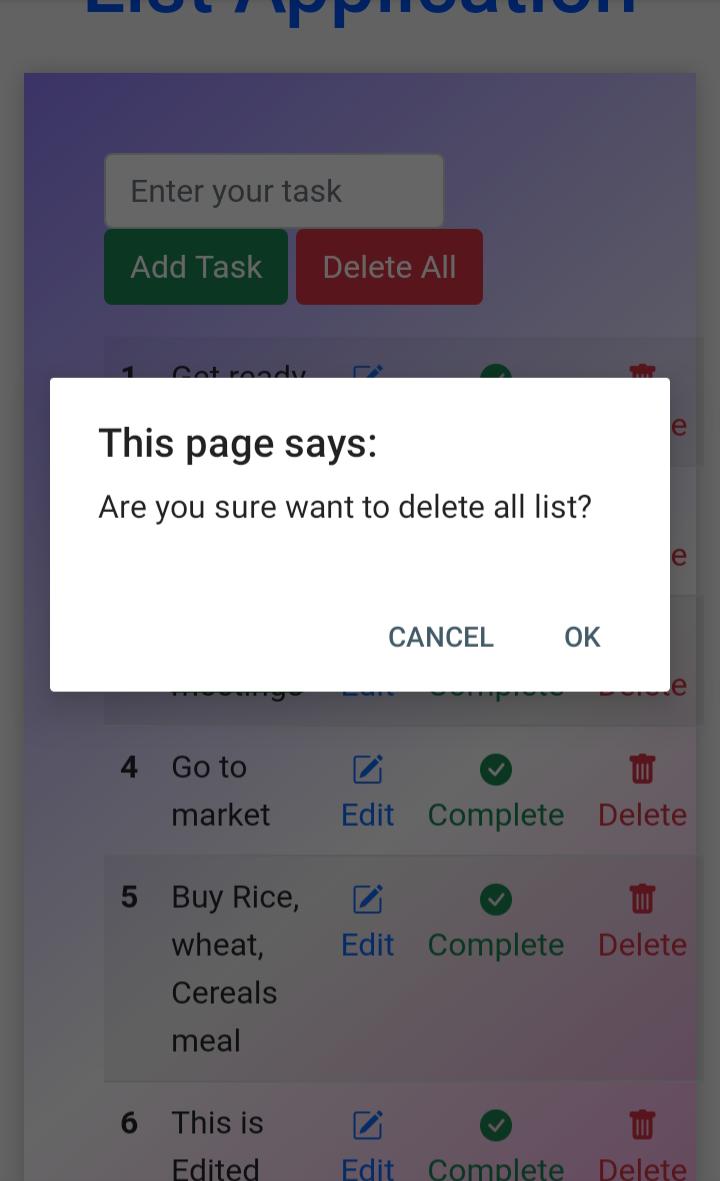
**6.1) Show Confirm or alert while deleting**

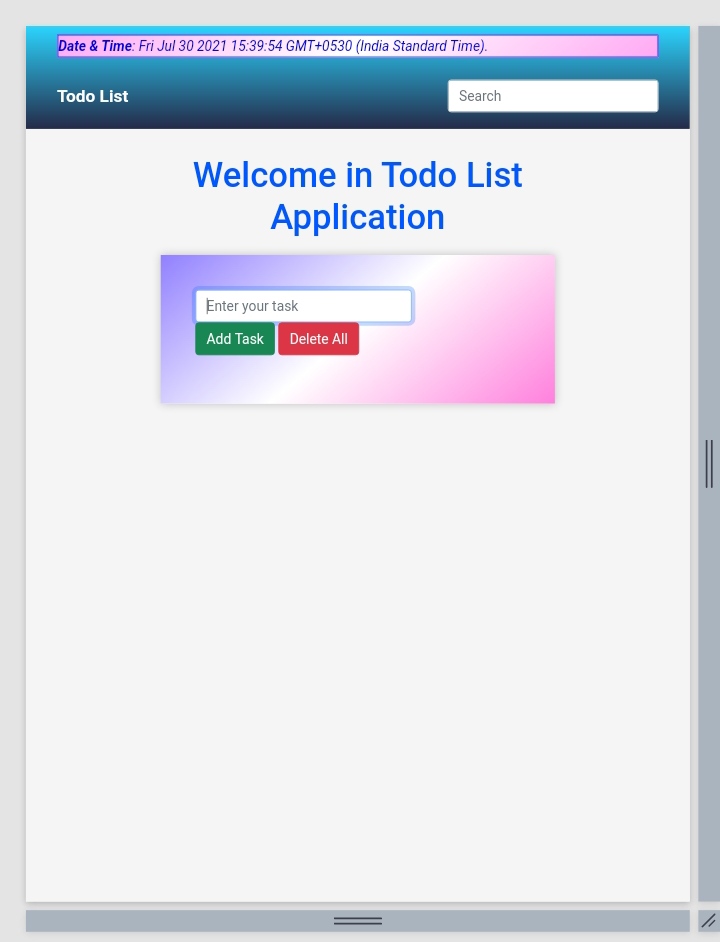
****

**6.2) After Delete tasks (Lunch is deleted from this list)**

****

**7.Delete All Tasks or delete all list**

**7.1) Show Confirmation page while deleting all list**

**8.Exit (After Deleting all list) **

**CHAPTER-5:**

**PROJECT TESTING**

**SOFTWARE TESTING STRATEGIES**

Testing is a set of activities that can be planned in advanced and conducted systematically. A strategy for software testing must accommodation low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements

There are three types of testing strategies

1. Unit test
2. Integration test

3. Performance test

**Unit Testing:**

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

**Integration Testing:**

Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

**Performance Testing:**

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

**CHAPTER-6:**

**CONCLUSION**

The **“TO-DO LIST”** has been successfully completed. The goal of the system is achieved and problems are solved. The package is developed in a manner that it is user friendly and required help if provided at different levels.

The project can be easily used in the process of decision making. Different types of reports can be generated which help the management to take correct decision and reduce the time delay which automatically increases the company’s work standards as well as the economical state of the company.

This system never decreases the manpower but helps the development of available manpower and optimizes the manpower by which company’s standards and capabilities can be scaled to higher dimensions.

**CHAPTER-7:**

**REFERENCE**

1. [www.Checkify.com](http://www.Checkify.com)
2. [www.wrike.com](http://www.wrike.com)
3. <https://medium.com>
4. <https://bootstrap.com>
5. <https://W3School.com>
6. [www.wikipedia.com](http://www.wikipedia.com)