**To-Do Application**

**A**

**Final REPORT**

***In partial fulfilment for the award of the degree***

***of***

**MASTERS OF COMPUTER APPLICATIONS**

***In***



**Directorate of Online Education Manipal University Jaipur**

**YEAR**

***November 2021-23***

1



**CERTIFICATE**

*This is to certify that the seminar entitled “ToDo Application” is a bona fide record of the major seminar report done by* ***Radhika Jethlia (Roll No 2114504907)****, under my supervision and guidance, in partial fulfilment of the requirements to complete the course MCA from Manipal University Jaipur for the year 2021-2023.*

***Signature of Student Signature of Faculty***

2

**ACKNOWLEDGEMENT**

*I would like to express my Gratitude towards my guide,* ***Shikha Maheshwari*** *without whose support, this seminar report would have been a complete failure. I would also like to thank other concerned faculties that supported me throughout the seminar report and helped me achieve this goal.*

**Radhika Jethlia**

3

**TABLE OF CONTENTS**

| **Section** | **Chapter** | **Topic** | **Page No.** |
| --- | --- | --- | --- |
| Section - A |  | Abstract | 5 |
| Section – B  (Seminar  Report) | I | Introduction | 6 |
| II | Objective of the study | 6,7 |
| III | Review of Literature | 7-25 |
| IV | Chapters developing the main theme of the seminar or any Practical Implication | 25,26 |
| V | Discussion | 27-34 |
| VI | Scope for further study | 35 |
| VII | Conclusion | 36 |
| VIII | References | 37 |

4

**SECTION - A**

**Abstract**

The ToDo app is a simple yet powerful tool that helps users manage and organize their tasks. Users can create and organize tasks, set reminders, and track progress over time. Business apps come in many forms, including web-based apps, mobile apps, and desktop software. It can be used for personal productivity, project management or team collaboration.

The ToDo app helps users prioritize tasks, avoid procrastination, and improve overall efficiency. Whether you're a student, professional, or entrepreneur, the ToDo app is an essential tool to help you stay on task and achieve your goals. The main purpose of the ToDo app is to help users be more productive, manage time effectively, and achieve their goals by being organized and focused.

A ToDo schedule is a list of tasks that need to be completed, usually organized by priority. It is one of the easiest task management solutions and offers a minimal and elegant way to manage the tasks you want to do. Our goal is to design a simple and beautiful PWA to help people track the status of their tasks.

Creating tasks is a simple and important thing that everyone should do. Completing a task and marking it off your to-do list gives you immense satisfaction. Plus, creating a to-do list ensures you never miss a thing. It's a scientific fact that writing down what needs to be done makes you more motivated to complete it. With this in mind, I decided to build a PWA ToDo application that helps people create their own to-do lists.

With the help of modern tools and technologies, we create a minimal and efficient work schedule that minimizes distractions and helps people manage work easily and effortlessly.

5

**SECTION – B**

**Introduction**

A new place to build a new life may seem like an exciting prospect, but there may be obstacles in the way. The current state of automation systems, whether software or hardware, is not recommended in this day and age. Manual systems. Validation of registration forms in other data storage processes is often done manually and on paper. Therefore, using an automated system avoids a lot of repetition.

By introducing this system, the efficiency of the system can be improved and the shortcomings of the existing manual system can be solved. This system is designed to help users organize and manage their daily tasks and goals. This program provides an automated system to avoid errors in manual labs and data entry. The system is user-friendly and does not require formal knowledge to operate. It is also safe, reliable and fast, allowing users to focus on other activities.

The ToDo app helps users with strategic planning and ensures that they have the right level of information and detail to achieve their goals. Users can maintain computerized records without making additional entries, thus reducing the distraction of irrelevant information.

The ToDo app has special features such as task categories, due dates, priority levels, and status updates. Users can create tasks, assign them to specific categories, set due dates and priorities, and update the status as they progress. The app also provides notifications and reminders so users don't miss important deadlines.

The app has a unique ID for each user and allows users to save personal details and settings. This information is used to provide personalized recommendations and suggestions to help you manage your tasks more efficiently.

**Objective of the Study**

The purpose of the ToDo app depends on the needs and goals of the user. Here are the goals that the ToDo app helps users achieve:

1. **Task Management -** The main purpose of the ToDo app is to help users manage tasks more efficiently. The ToDo app allows users to create, organize and manage their tasks in one place. This makes it easier to keep track of everything you need to do and avoid forgetting important tasks.

2. **Prioritization -** Another goal of ToDo apps is to help users prioritize their tasks. The To-Do app helps users prioritize tasks based on urgency, importance, and due

6

date. This allows users to focus their time and energy on the most important tasks first and avoid blogging less important tasks.

3. **Productivity** - The To-Do app helps users stay productive and focused throughout the day by providing a clear and concise overview of all their tasks. This increases efficiency and allows you to do more in less time.

4. **Accountability** - The To-Do app helps users hold themselves accountable for completing their tasks. The app provides visual reminders of tasks that need to be done and allows users to mark tasks as complete when completed.

5. **Collaboration -** Many work apps have collaboration features that allow teams to share tasks, assign responsibilities, and track progress. It is useful for group projects and teamwork.

6. **Time Management** - The To-Do app helps users manage their time more effectively by providing a clear view of their daily and weekly schedules. This allows users to avoid over-commitment and have enough time to complete all tasks.

**Review of Literature**

**Global Use:**

Universal use cases for the ToDo app can be seen in a variety of scenarios, from personal use to business operations. It helps people and organizations manage tasks, organize and improve productivity. For personal use, ToDo apps help people manage daily tasks such as grocery shopping, paying bills, and doing household chores. The app helps users prioritize tasks and set deadlines so they can be completed on time. Users can create lists of tasks in different categories, such as work-related tasks, personal tasks, and leisure activities.

The ToDo app is also useful for students who often manage a large number of assignments and projects. This app helps students prioritize their homework and keep track of important deadlines. It also helps students manage their study plans and track their progress in different subjects. In a professional setting, the ToDo app can be used by individuals and teams to manage tasks and projects. In this context, applications help users prioritize their work, collaborate with team members, and track progress on various tasks. It also helps teams manage workloads and ensure deadlines are met.

Businesses can use the ToDo app to manage tasks and projects across departments and teams. The program helps businesses streamline operations, increase efficiency, and improve productivity. It also helps companies track progress and identify areas for improvement. The ToDo app can be used not only for task management, but also for goal setting and habit tracking. Users can set goals for themselves, such as exercising regularly or reading more books, and track their progress over time. The app also helps users develop new habits and break old ones by providing reminders and notifications.

7

Overall, ToDo apps can be a valuable tool for individuals and organizations looking to manage their tasks and improve productivity. With a user-friendly interface and powerful features, this app helps users plan, stay focused, and stay on track to achieve their goals.

**About Technology:**

**1) NextJS:**

Next.js is a popular React-based open-source framework used to build web applications. The platform was created by Vercel, a cloud platform that allows developers to easily deploy web applications. Next.js makes it easy for developers to build server-rendered React applications without having to deal with complex build system configurations and server settings.

The main goal of Next.js is to provide developers with a seamless experience when building React applications. Next.js makes it easy for developers to build server-rendered React apps without worrying about server-side rendering details. This makes it easy to create fast, scalable and maintainable web applications.

One of the main features of Next.js is server-side rendering. This feature allows developers to render React components on the server instead of the client. This is a powerful feature that offers many benefits. First, it improves application performance by reducing time to first color (TTFP) and time to interaction (TTI). Second, it makes your app more accessible to search engines and screen readers. Finally, it improves the user experience with faster and more consistent page loading.

Another important feature of Next.js is automatic code splitting. This feature automatically splits your code into smaller chunks that can be loaded as needed. This improves app performance by reducing the size of the initial download and reducing the amount of code that needs to be loaded on subsequent page loads. This feature also helps developers optimize app performance by ensuring that only necessary code is loaded on each page.

Next.js also supports static site generation. This feature allows developers to generate a static HTML file for each page of their application. This is particularly useful for applications with static content, as the content can be pre-rendered and served from a content delivery network (CDN). This improves application performance by reducing the amount of server-side processing required to render each page.

Another important feature of Next.js is the ability to create dynamic routes. This feature allows developers to create pages with dynamic URLs. This is very useful for applications with dynamic content. For example, an e-commerce application can use dynamic routing to create product pages with unique URLs for each product.

Next.js also supports TypeScript, a popular programming language that adds type annotations to JavaScript. This makes it easier for developers to write maintainable code and catch errors at compile time instead of at runtime. This feature is especially useful for large applications that require more complex data types and data structures.

8

In addition to these features, Next.js has many other advantages that make it a popular choice for building web applications. For example, it provides built-in support for CSS modules, making it easy to create modular and reusable CSS styles. It also provides support for hot module reloading. This allows developers to view code changes in real time.

Overall, Next.js is a powerful framework that makes it easy for developers to build server-rendered React applications. Focusing on performance, accessibility and usability.

**2) PWA:**

Progressive Web Applications (PWA) is a revolutionary technology that enables developers to create web applications that behave like native applications. PWAs are designed to provide users with a better user experience, allowing them to access their favorite apps on any device, regardless of operating system or platform.

One of the main advantages of PWAs is that they work offline, so users can access your app even when they are not connected to the Internet. This is made possible by saving application data and resources on the user's device so that they can be accessed offline. This means that users can continue to use the app even in areas with poor or no internet connection.

Another advantage of PWAs is that they can be installed on users' devices like native apps without going to an app store. This makes the installation process much easier and faster, eliminating the need to download large program files. PWAs also offer many other features that set them apart from traditional web apps. For example, push notifications can be sent to the user's device, just like a native app. This allows developers to provide users with timely and relevant information such as reminders and updates of their favourite apps.

PWAs are also designed to be fast and responsive and focus on providing a seamless user experience. They are optimized for mobile devices and have a lightweight design that minimizes the amount of data that needs to be transferred over the network. This makes PWAs ideal for use on slow or unreliable internet connections where traditional web applications have difficulty loading.

In addition to these features, PWAs are also designed to be secure and reliable. We use HTTPS to encrypt all data transferred between user devices and our servers, ensuring that user data is always protected. It also provides a reliable and consistent user experience with automatic updates that keep your apps up-to-date and running smoothly.

Overall, PWAs are a big step forward in web application development. They offer a range of features and benefits that make them ideal for use on any device and provide a great user experience that rivals native apps. With the continued growth of the mobile web, PWAs will undoubtedly become an important part of the app development environment in the coming years.

3) SQL :

Creating a SQL use case for a TODO app in Next.js involves several components like database setup, SQL queries, coding standards, efficiency, error handling, and parameter validation. Let's break it down step by step.

1. \*\*Database Setup\*\*: First, you'll need to set up a database to store your TODO items. We'll use PostgreSQL as an example.

2. \*\*Coding Standards and Efficiency\*\*:

- Follow a consistent coding style guide (e.g., Airbnb JavaScript Style Guide).

- Use async/await for asynchronous operations.

- Keep code modular and organized.

- Use environment variables for sensitive data like database credentials.

3. \*\*Error Handling\*\*:

- Implement try-catch blocks to handle errors gracefully.

- Use meaningful error messages.

- Log errors for debugging purposes.

4. \*\*Parameter Validation\*\*:

- Validate input parameters to prevent SQL injection.

- Use parameterized queries or an ORM (Object-Relational Mapping) library to handle parameterization.

Here's an example of a SQL use case for creating and retrieving TODO items:

```javascript

// Import necessary libraries and set up a database connection

const { Pool } = require('pg');

const pool = new Pool({

connectionString: process.env.DATABASE\_URL,

});

// Create a TODO item

async function createTodoItem(userId, title, description) {

try {

// Validate input parameters

if (!userId || !title) {

throw new Error('User ID and title are required.');

}

// Insert TODO item into the database using a parameterized query

const query = 'INSERT INTO todo\_items (user\_id, title, description) VALUES ($1, $2, $3) RETURNING \*';

const values = [userId, title, description];

const result = await pool.query(query, values);

return result.rows[0];

} catch (error) {

console.error('Error creating TODO item:', error.message);

throw error;

}

}

// Retrieve TODO items for a specific user

async function getTodoItemsByUserId(userId) {

try {

// Validate input parameter

if (!userId) {

throw new Error('User ID is required.');

}

// Fetch TODO items from the database

const query = 'SELECT \* FROM todo\_items WHERE user\_id = $1';

const values = [userId];

const result = await pool.query(query, values);

return result.rows;

} catch (error) {

console.error('Error retrieving TODO items:', error.message);

throw error;

}

}

module.exports = {

createTodoItem,

getTodoItemsByUserId,

};

```

In this example, we've created functions for creating and retrieving TODO items. These functions include parameter validation, error handling, and use parameterized queries to prevent SQL injection.

Please note that this is a simplified example, and in a real-world application, you would likely have more features, such as updating and deleting TODO items, user authentication, and more comprehensive error handling.

**4) NodeJS:**

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside of a web browser. It was created by Ryan Dahl in 2009, and has since become a widely used technique for building scalable, fast, and efficient web applications.

One of the primary advantages of Node.js is its ability to handle asynchronous I/O operations, making it well suited for real-time, data-driven applications. This means that while one part of the application is waiting for a response from the database or API, other parts of the application can continue to work.

Another major advantage of Node.js is the wide range of packages and modules available through the Node Package Manager (NPM). These modules can be easily integrated into Node.js applications, allowing developers to leverage existing code and tools to accelerate development. In addition to its core functionality, Node.js also provides a number of tools and frameworks that make it easy to develop, test, and deploy applications.

Here are some of the most popular tools and frameworks in the Node.js ecosystem:

∙ Express.js: Express.js is a lightweight and flexible web application framework that provides a range of features for building scalable, modular, and secure web applications. It includes a robust middleware system, template engines, and support for HTTP/2 and WebSocket’s.

11

∙ Socket.io: Socket.io is a real-time engine that enables bidirectional communication between clients and servers. It's particularly useful for building chat applications, online games, and other real-time applications that require low-latency communication.

∙ Mongoose: Mongoose is a MongoDB object modelling tool designed to work in an asynchronous environment. It provides a straightforward, schema-based solution for modelling complex data structures and integrates easily with Express.js.

∙ PM2: PM2 is a production process manager for Node.js applications that provides features like automatic restarts, load balancing, and logging. It's particularly useful for applications that need to run continuously and require high availability.

∙ Jest: Jest is a JavaScript testing framework that makes it easy to write and run tests for Node.js applications. It includes features like snapshot testing, mocking, and parallel test execution.

∙ Webpack: Webpack is a module bundler that takes modules with dependencies and generates static assets representing those modules. It's particularly useful for building large, complex applications with multiple modules and dependencies.

∙ Node.js has also gained popularity as a tool for building microservices and serverless architectures. Its ability to handle high concurrency and I/O operations makes it well suited for these types of architectures, which typically involve many small, independent services communicating with each other over a network.

Node.js is used by many large organizations, including Netflix, LinkedIn, and PayPal, as well as by a growing number of startups and individual developers. Its popularity is due in large part to its flexibility, scalability, and ease of use, as well as its active community of developers who contribute to its ongoing development and improvement.

**5) Features:**

Our project is an ambitious endeavour designed to address the need for easy and convenient management of daily tasks. The project was developed using NextJS and PWA with Firebase as the backend and authentication system, making it a powerful and reliable tool to help users stay organized and productive.

One of the key features of the system is the management module. It provides a comprehensive dashboard where managers can easily view all the details. It includes the total number of users and allows administrators to easily manage user information and tasks. The User module allows administrators and users to enter the home page if the password and username match.

The User module is primarily used to manage system users. The user section allows the user to view general information such as our company and the registration page. Registered

12

users can add, update and manage their profile, change their password and view their tasks. The system also has a feature that allows registered users to recover their passwords. The system provides search functionality based on various factors such as pending and completed tasks, making it easy for users to find what they need quickly and efficiently. The system also has powerful edit, add and update capabilities that ensure proper resource management. This feature allows users to easily edit, add or update records and provides a seamless user experience. It also means users can easily manage information and descriptions, helping them stay organized and focused.

Login modules are used to manage login details and ensure the security of user information. On the other hand, the Tasks module is used to manage task details, allowing users to prioritize their work and avoid procrastination. This system also includes the ability to change users' passwords and add an additional layer of security.

In short, our project is an all-in-one task management solution designed to help users stay organized, productive, and efficient. With powerful features and an easy-to-use interface, this system is suitable for students, professionals and entrepreneurs alike. Its compatibility with various hosting platforms makes it an ideal choice for businesses looking to increase productivity and achieve their goals.

**6) Implementation/working:**

ToDo App is an innovative project that aims to provide an easy-to-use platform for managing daily tasks. With this project, new users can easily register and receive login credentials that they can use to access system functionality. The main goal of this project is to develop a system that helps users keep records and view information related to their tasks.

The system is designed to be simple and easy for users to navigate, making it easy for them to understand how to use it to manage their tasks. This project is based on the NextJS framework, which is known for its efficiency and ease of use. Using Firebase as a backup and authentication system ensures that the system is secure, reliable and protects your users' data.

One of the key features of this system is the ability to search tasks based on various factors such as pending tasks and completed tasks. This feature makes it easy for users to prioritize their tasks and manage their time effectively.

The system's resource management is also top-notch, simplifying editing, adding, and updating records, allowing users to effectively manage their information. The system also manages user information so that users can easily view their profile and manage their contact information.

The system has a management module that provides a dashboard where administrators can easily view all the details such as the total number of users. Administrators manage user information and tasks to maintain system efficiency.

The user module is also one of the key features of the system, which allows users to log in to the home page only if their username and password match. This module is used to manage system users and allows registered users to add, update and manage their profiles. You can also change your password and recover it if necessary.

13

The login module is used to manage the login details and the task module is used to manage the task details. The system allows users to easily change passwords and store user information such as profile details and contact information.

This system is designed to address the challenges that arise from manual task management. By identifying the shortcomings of the existing systems, the developers of this project designed a computer system compatible with the existing systems. This system is more user-friendly and based on graphical user interface, making it easier for users to navigate and use.

System efficiency is greatly improved and users can overcome the shortcomings of manual systems. The system also facilitates the generation of backup data and allows users to access their information at any time.

As a result, this project is an essential tool for anyone who wants to improve their productivity, manage their time effectively, stay organized and focused to achieve their goals. This system is designed to be simple and easy to use, users can easily use it. With features like task modules and user modules, this project provides a comprehensive solution for effective task management. The system's compatibility with existing systems and easy generation of backup data makes it a must-have for anyone looking to stay organized and productive.

**7) System Testing:**

During system testing, the system is used on a trial basis to ensure that the software does not crash. In other words, it works as designed and as users expect. Enter and process special test data and review results. A limited number of users can be allowed to use the system so that analysts can see if they are trying to use the system in unexpected ways. It is desirable to discover surprises before an organization implements and relies on a system. You should keep in mind that the philosophy behind testing is to find errors. Test cases are designed with this goal in mind. A test case is a set of data that the system treats as normal input. However, the data is created for the purpose of determining whether the system is processing the data correctly. For example, inventory handling test cases should include situations where the amount taken from inventory is greater than, equal to, or less than the actual inventory amount. Each test case is designed to detect errors in the way the system handles them.

There are two general strategies for testing software:

**1. Code testing and**

**2. Specification testing.**

In **Code testing**, the analyst develops those cases to execute every instruction and path in a program. Under specification testing, the analyst examines the program specifications and then writes test data to determine how the program operates under specific conditions. Regardless of which strategy the analyst follows, there are preferred practices to ensure that the testing is useful. The levels of tests and types of test data, combined with testing libraries, are important aspects of the actual test process.

14

Testing includes three parts:

1. Testing- Error detection and error removal of newly developed program, so that it produces specific output with correctness of live and artificial data.

2. Verification- It is also a kind of testing of system in simulated environment using simulated data (alpha testing). It is done to detect and wait error regarding end user and design specification we were specified in the earlier phase.

3. Validations- It refers to process of using software in a live environment in order to find errors. Feedback of validation phase generally produces change in software to deal with error and failure that are uncovered. Transaction that are extended and person using the system and real and continue for several months.

Systems are not designed as entire systems nor are they tested as single systems. The analyst must perform both unit and system testing.

Testing is of different types some of them are given below:

**1. Unit Testing**

**2. System Testing**

**Unit Testing:** In unit testing the analyst tests the programs making up a system. For this reason, unit testing is sometimes called program testing. Unit testing gives stress on the modules independently of one another, to find errors. This helps the tester in detecting errors in coding and logic that are contained within that module alone. The errors resulting from the interaction between modules are initially avoided. For example, a hotel information system consists of modules to handle reservations; guest check-in and check-out; restaurant, room service and miscellaneous charges; convention activities; and accounts receivable billing. For each, it provides the ability to enter, modify or retrieve data and respond to different types of inquiries or print reports. The test cases needed for unit testing should exercise each condition and option.

Unit testing can be performed from the bottom-up, starting with smallest and lowest level modules and proceeding one at a time. For each module in bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embed within the larger system.

∙ Test of individual module/program. It is called program testing.

∙ Valid and invalid data are entered to generate possible error. For Example: - Numeric error or Date error.

∙ It focuses in any module but not integrated part. It does not consider any linking error. First stage of testing.

∙ Output – error free program.

∙ All programs are working independently.

15

**A) Black Box Testing**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has following categories: • Incorrect or missing functions

• Interface errors

• Errors in data structure or external database access

• Performance errors

• Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

**B) White Box Testing**

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

• Guarantee that all independent paths have been executed.

• Execute all logical decisions on their true and false Sides.

• Execute all loops at their boundaries and within their operational bounds • Execute internal data structures to ensure their validity.

System testing is an important and integral part of the system development phase that follows software design and development. Not all programs and systems are perfectly designed. There are errors in software development due to lack of communication between users and designers. The number and nature of errors in newly designed systems depends on several common factors such as communication between users and designers. A programmer's ability to produce code that accurately reflects system specifications and design timeframes.

In theory, a newly designed system should have all parts or subsystems working properly, but in practice each subsystem works independently. Now gather all the subsystems into a pool and test the entire system to see if it meets your needs. This is the last change to identify and correct any errors before installing the system for user acceptance testing. The purpose of system testing is to push the system to its limits, taking into account all possible changes that the system may undergo.

Testing is an important function to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully the important and essential part of the system development phase, after designing and developing the software is system testing. We cannot say that every program or system design is perfect and because of lack of communication between the user and the designer, some error is there in the software development. The number and nature of errors in a newly designed

16

system depend on some usual factors like communication between the user and the designer; the programmer's ability to generate a code that reflects exactly the systems specifications and the time frame for the design.

In theory, a newly designed system should have all parts or subsystems working properly, but in practice each subsystem works independently. Now gather all the subsystems into a pool and test the entire system to see if it meets your needs. This is the last change to identify and correct any errors before installing the system for user acceptance testing. The purpose of system testing is to push the system to its limits, taking into account all possible changes that the system may undergo.

Testing is a critical function for the success of your system. System testing makes the reasonable assumption that if all parts of the system are correct, the objective will be achieved.

System testing is an important and integral part of the system development phase that follows software design and development. Not all programs and systems are perfectly designed. There are errors in software development due to lack of communication between users and designers. The number and nature of errors in newly designed systems depends on several common factors such as communication between users and designers. A programmer's ability to produce code that accurately reflects system specifications and design timeframes.

In theory, a newly designed system should have all parts or subsystems working properly, but in practice each subsystem works independently. Now gather all the subsystems into a pool and test the entire system to see if it meets your needs. This is the last change to identify and correct any errors before installing the system for user acceptance testing. The purpose of system testing is to push the system to its limits, taking into account all possible changes that the system may undergo.

Testing is a critical function for the success of your system. System testing makes the reasonable assumption that if all parts of the system are correct, the objective will be achieved.

**8) User Interface Design:**

Any project how so ever good it may be would become failure if the end user it is meant to cater cannot use it. For the end user to reap full benefits of the project the project should have a good interface. Interface is the way a project links with the end users; it accepts the inputs from the user and gives output to the user. For a project to be successful it should have a clean and understandable interface.

❖ Properties of a good interface

• The text written should be clear.

• Proper labelling should be done on labels and buttons.

• Proper warnings should be provided to the user at all levels.

• It should not be too bright in colours.

17

**9) System Architecture:**

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called presentation layer, business logic layer and data link layer.

This project was developed using 2-tier architecture.

**SYSTEM DESIGN**

Based on the user requirements and the detailed analysis of a new system, the new system must be designed. This is the phase of system designing. It is a most crucial phase in the development of a system.

Normally, the design proceeds in two stages:

• Preliminary or General design

• Structure or Detailed design

**Preliminary or General design:** In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible, we move to the detailed design stage.

**Structure or Detailed design:** In the detailed design stage, computer-oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structure design is a blue print of a computer system solution to a given problem having the same components and inter-relationship among the same components as the original problem. Input, output and processing specifications are drawn up in detail. In the design stage, the programming language and the platform in which the new system will run are also decided.

**THE PRIMARY OBJECTIVE OF THE DESIGN:**

Of course, is to deliver the requirements as specified in the feasibility report. In general, the following design objectives should be kept in mind:

∙ Practicality: The system must be stable and can be operated by the people with average intellectual.

∙ Efficiency: This involves accuracy, timeliness and comprehensiveness of the system output.

∙ Cost: it is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.

18

∙ Flexibility: The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems.

∙ Security: This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse.

System design involves first logical design and then physical construction of the system. The logical design describes the structure and characteristics of features, like the outputs, inputs, files, databases and procedures. The physical construction, which follows the logical design, produces actual program software, files and a working system.

The designer normally will work under following constraints:

∙ Hardware: The existing hardware will obviously affect the system design.

∙ Software: The available software (operating system, utilities, language etc.) in the market will constrain the design.

∙ Budget: The budget allocated for the project will affect the scope and depth of design.

∙ Time-scale: The new system may be required by a particular time (e.g., the start of a financial year). This may put a constraint on the designer to find the best design.

∙ Interface with other systems: The new system may require some data from another computerized system or may provide data to another system in which case the files must be compatible in format and the system must operate with a certain processing cycle.

**SYSTEM ANALYSIS**

Assuming that a new system is to be developed, the next phase is system analysis. Analysis involved a detailed study of the current system, leading to specifications of a new system. Analysis is a detailed study of various operations performed by a system and their relationships within and outside the system. During analysis, data are collected on the available files, decision points and transactions handled by the present system. Interviews, on-site observation and questionnaire are the tools used for system analysis.

Using the following steps, it becomes easy to draw the exact boundary of the new system under consideration:

▪ Keeping in view the problems and new requirements.

▪ Workout the pros and cons including new areas of the system

The main points to be discussed in system analysis are:

19

⮚ Specification of what the new system is to accomplish based on the user requirements. ⮚ Functional hierarchy shows the functions to be performed by the new system and them relationship with each other.

⮚ Function network which are similar to function hierarchy but they highlight those functions which are common to more than one procedure.

⮚ List of attributes of the entities - these are the data items which need to be held about each entity (record).

The selection process should be viewed as a project and a project team should be formed with the help of management. The selection process consists of several steps, which are discussed below:

⮚ **Requirements analysis-** The first step in selection understands the user's requirements within the framework of the organization’s objectives and the environment in which the system is being installed.

⮚ **Request for proposal-** After the requirement analysis and system specifications have been defined, a request for proposal is prepared and sent to selected vendors for bidding.

⮚ **Evaluation and validation-** The evaluation phase ranks various vendor proposals and determines the one best suited to the user's requirements. It looks into items such as price, availability and technical support. System validation ensures that the vendor can, in fact, match his/her claims, especially system performance.

⮚ **Vendor selection-** This step determines the vendor with the best combination of reputation, reliability, service record, training, delivery time, lease/finance terms. The selected vendors are invited to give a presentation of their system. The system chosen goes through contract negotiations before implementation.

⮚ **System specifications-** System specifications must be clearly defined. These specifications must reflect the actual applications to be handled by the system and include system objectives, flowcharts, input- output requirements, file structure and cost.

**10) Database:**

A Database is a collection of interrelated data stored with a minimum of redundancy to serve many applications. The database design is used to group data into a number of tables and minimizes the artificiality embedded in using separate files.

The tables are organized to:

❖ Reduced duplication of data.

❖ Simplify functions like adding, deleting, modifying data etc.,

❖ Retrieving data.

❖ Clarity and ease of use.

❖ More information at low cost.

20

**11) HTML:**

Hypertext Markup Language (HTML), the language of the World Wide Web (WWW), allows you to create web pages that contain text, graphics, and pointers (hyperlinks) to other web pages.

HTML is not a programming language, but an adaptation of the ISO 8879 standard, Standard Generalized Markup Language (SGML), but focused on hypertext and adapted for the web. The idea behind hypertext is from one point to another. You can browse the information based on your interests and preferences. A markup language is a set of things that can be displayed in an element.

A hyperlink is an underlined or highlighted piece of work that is loaded into another document or part of the same document. HTML can be used to display any type of document on host computers in different geographical locations. A versatile language that can be used on any platform or desktop

HTML provides tags (special codes) to make your documents look attractive. The HTML renderer is not case sensitive. Graphics, fonts, different sizes, colours and more can improve the presentation of your documents. Anything that is not a label is part of the document itself.

**Advantages**

⮚ A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.

⮚ HTML is platform independent.

**12) System Study:**

∙ Information Gathering: The system being developed is economic with respect to Organization's point of view. It is cost effective in the sense that has eliminated the paper posting work completely. The system is also time effective because the mailing system is very fast which sends Email quickly as per the user or organization requirement. The Emailing through JSM (Java Secure Mail) is secure, flexible and pretend to virus, spam or hacking attack by outside world. The system also collects the information of employee of organization who registered on it.

∙ Feasibility Analysis: Many feasibility studies are disillusioning for both users and analysts. First, the study often presupposes that when the feasibility document is being prepared, the analyst is in a position to evaluate solutions. Second, most studies tend to overlook the confusion inherent in system development the constraints and the assumed attitudes.

If the feasibility study is to serve as a decision document, it must answer three key questions:

21

o Is there a new and better way to do the job that will benefit the user? o What are the costs and savings of the alternative(s)?

o What is recommended?

Once the feasibility study is done and proposal is approved, we move to the requirement analysis. A feasibility study is the test to proposal according to its viability impact on the organization, ability to meet user needs, and effective use of the resources. The objective of the feasibility is not to solve the problem but to acquire a sense of its viability scope.

There are three key considerations involve in the feasibility analysis. Once it has been determined that a project is feasible, we can go ahead and prepare the project specification which finalizes project requirements.

Generally, feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. The contents and recommendations of such a study will be used as a sound basis for deciding whether to prove, postpone or cancel the project. Thus, since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental errors of judgment are made. The most successful system projects are not necessarily the biggest or most visible business but rather those that truly meet user expectations. More projects fail because of inflated expectations than for any other reason.

∙ Feasibility considerations: Three key considerations are involved in the feasibility analysis:

❖ Economic

❖ Technical

❖ Behavioural.

Let's briefly review each consideration and how it relates to the system efforts.

Depending on the results of the initial investigations, the survey is expanded to a more detailed feasibility study. A feasibility study is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources.

It focuses on three major questions:

1. What are the user's demonstrable needs and how does a candidate system meet them? 2. What resources are available for given candidate systems? Is the problem worth solving?

3. What are the likely impacts of the candidate system on the organization? How well does it fit within the organization's master MIS plan?

Each of these questions must be answered carefully. They revolve around investigation and evaluation of the problem, identification and description of candidate systems, specification of performance and the cost of each system, and final selection of the best system. The objective of a feasibility study is not to solve the problem but to acquire a sense of its scope.

22

During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Consequently, costs and benefits are estimated with greater accuracy at this stage.

The result of the feasibility study is a formal proposal. This is simply a report-a formal document detailing the nature and scope of the proposed solution. The proposal summarizes what is known and what is going to be done.

It consists of the following:

**Statement of the problem-** A carefully worded statement of the problem that led to analysis.

**Summary of findings and recommendations-** A list of the major findings and recommendations of the study. It is ideal for the user who requires quick access to the results of the analysis of the system under study. Conclusions are stated, followed by a list of the recommendations and a justification for them.

**Details of findings-** An outline of the methods and procedures under taken by the existing system, followed by coverage of the objectives and procedures of the candidate system. Included are also discussions of output reports, file structures, and costs and benefits of the candidate system.

**Recommendations and conclusions-** Specific recommendations regarding the candidate system, including personnel assignments, costs, project schedules and target dates.

After the proposal is reviewed by management, it becomes a formal agreement that paves the way for actual design and implementation. This is a crucial decision point in the life cycle. Many projects die here, whereas the more promising ones continue through implementation. Changes in the proposal are made in writing, depending on the complexity, size and cost of the project. It is simply common sense to verify changes before committing the project to design.

Eight steps are involved in the feasibility analysis. They are:

1. Form a project team and appoint a project leader.

2. Prepare system flowcharts.

3. Enumerate potential proposed systems.

4. Define and identify characteristics of proposed system.

5. Determine and evaluate performance and cost effectiveness of each proposed system. 6. Weight system performance and cost data.

7. Select the best proposed system.

8. Prepare and report final project directive to management

23

**13) UML Diagrams:**

❖ **Unified Modelling Language:** The Unified Modelling Language allows the software engineer to express an analysis model using the modelling 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 diagrams, which is as follows.

1. USER MODEL VIEW-

- This view represents the system from the user’s perspective.

- The analysis representation describes a usage scenario from the end user’s perspective.

2. STRUCTURAL MODEL VIEW-

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

- This model view models the static structures.

3. BEHAVIORAL MODEL VIEW-

- It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural

elements described in the user model and structural model view.

4. IMPLEMENTATION MODEL VIEW-

- In this the structural and behavioural as parts of the system are represented as they are to be built.

5. ENVIRONMENTAL MODEL VIEW-

- In these the structural and behavioural 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 modelling, this focuses on the user model and structural model views of the system.

⮚ UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

24

**14) Data Flow Diagram:**

Graphical description of a system's data and how the processes transform the data is known as Data Flow Diagram (or DFD). Unlike detail flowcharts, DFD’s do not supply detailed descriptions of modules but graphically describe a system's data and how the data interact with the system.

The following seven rules govern construction of data flow diagrams (DFD):

o Arrows should not cross each other.

o Squares, circles, and files must bear names.

o Decomposed data flows must be balanced (all data flows on the decomposed diagram must reflect flows in the original diagram).

o No two data flows, squares, or circles can have the same name.

o Draw all data flows around the outside of the diagram.

o Choose meaningful names for data flows, processes, and data stores. Use strong verbs followed by nouns.

o Control information such as record counts, passwords, and validation requirements are not pertinent to a data-flow diagram.

If too many events seem to be occurring at a given point, an analyst can decompose a data conversion (circle). The new data conversions form a parent-child relationship with the original data conversion: the child circle.

**Chapters developing the main theme of the seminar or any Practical Implication**

Time management issues affect almost everyone in some way. Whether a student juggling class and extracurricular activities, a professional trying to balance work and personal life, or a stay-at-home mom struggling with household chores, Effective time management is critical to your success and productivity.

One of the biggest challenges in time management is organizing and tracking tasks and responsibilities. It's easy to get overwhelmed with too many tasks to complete, leading to stress, anxiety, and feeling constantly behind schedule.

This is where the ToDo app comes in handy. The ToDo app helps users prioritize work, avoid procrastination, and improve overall efficiency by providing a simple and intuitive way to manage and organize tasks. Whether you use it on your phone, tablet, or desktop computer, the ToDo app is a powerful tool for keeping track of your tasks and achieving your goals.

However, not all ToDo apps are created equal. Some are too complex, hard to use, or lack important features. Others are poorly designed or crash-prone, which can lead to frustration and wasted time.

25

The ToDo App is designed to solve these problems and provide users with a simple, reliable, and effective tool for managing tasks and staying organized. With a focus on user friendly design, intuitive functionality, and powerful features, the ToDo App is the perfect solution for anyone looking to improve their productivity and manage their time more effectively.

**Key Features of the ToDo App:**

a. User-Friendly Design: The ToDo App is designed with the user in mind. The interface is clean, intuitive, and easy to navigate, making it simple for users to add, edit, and delete tasks as needed.

b. Task Management: The ToDo App allows users to create, organize, and manage tasks in a variety of ways. Tasks can be categorized by project, priority, due date, and more, making it easy to stay on top of what needs to be done.

c. Reminder Notifications: Users can set reminders for tasks, ensuring that important deadlines are not missed. The ToDo App will send notifications to remind users of upcoming tasks, making it easy to stay on schedule.

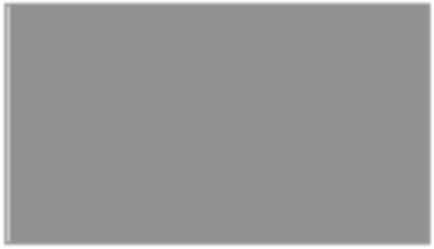
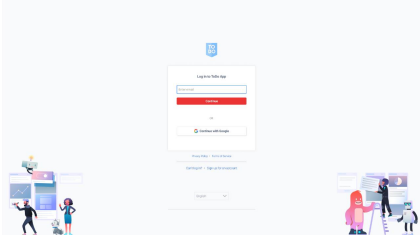
d. Sharing and Collaboration: The ToDo App allows users to share tasks and collaborate with others. This can be particularly helpful for team projects or family tasks, ensuring that everyone is on the same page and working towards a common goal.

e. Data Backup and Sync: The ToDo App automatically backs up data and syncs it across devices, ensuring that important information is never lost. This feature is particularly important for users who rely on the app for important tasks and deadlines.

Overall, the ToDo App is the perfect solution for anyone looking to improve their productivity, manage their time more effectively, and stay organized. With a focus on user friendly design, intuitive functionality, and powerful features, the ToDo App is the ultimate tool for managing tasks and achieving goals.

26

**Discussion**

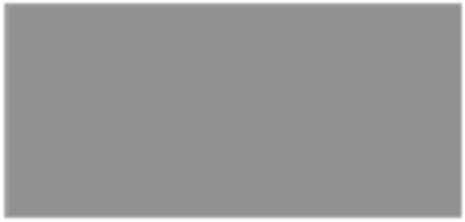
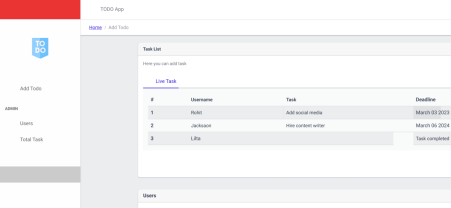
****

Built with NextJS and Firebase, the ToDo app login page is a key component of the app. It provides a secure way for users to access their personal tasks and ensures that only authorized people can make changes to their accounts. A login page typically contains a form with fields where the user can enter an email address and password that is verified against the Firebase authentication service.

If the credentials are correct, the user is redirected to the ToDo app home page, where they can view, add, or delete tasks. In addition, the login page may include options to reset passwords and create new user accounts. Using NextJS and Firebase brings many benefits to your login page, including better performance, scalability, and security. NextJS server-side rendering reduces page load times and makes the user experience more responsive. Firebase's strong authentication service also ensures that user data is protected from unauthorized access.

In general, the login page is an important part of the ToDo application, and with NextJS and Firebase we can make it secure, safe and user-friendly.

27



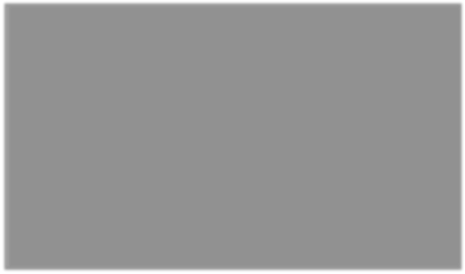
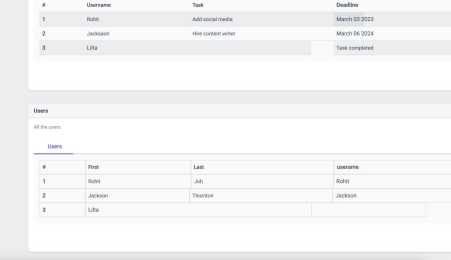
Adding a new ToDo task to a ToDo app built with NextJS and Firebase requires a few steps. First, users usually click a button or go to a specific page where they can create a new task. This page usually contains a form with fields for the job title, description, due date, and other relevant information.

When the user submits the form, your request requests the Firebase real-time database to store the new task data. This database is a cloud-hosted NoSQL database that enables real time data synchronization and auto-scaling. As new tasks are added to the database, the application can update the user interface to display the new tasks. This can be achieved using NextJS server-side rendering or client-side rendering in React.

Using NextJS and Firebase gives us advantages when adding new tasks to the ToDo app. NextJS server-side rendering can improve application performance by reducing the time it takes to load new tasks. Firebase Realtime Database provides a scalable and reliable backend for storing and synchronizing data and making new tasks available to all users in real time.

In general, adding a new ToDo task to a ToDo app built with NextJS and Firebase is a simple process that can be done with a few clicks and two API calls. These technologies provide a seamless and reliable user experience that helps users manage their tasks and stay productive.

28



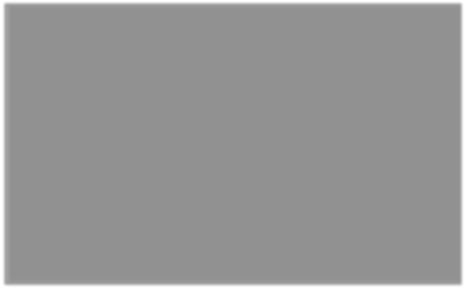
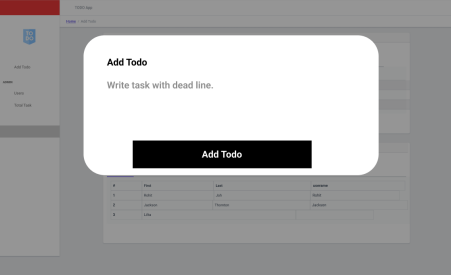
Created with NextJS and Firebase, the ToDo user list is a function that allows you to view a list of all registered users in the system. This list may include information such as user name, email address and other relevant information. User lists are useful for a variety of reasons, such as tracking team members and colleagues and managing user roles and permissions.

A user list typically consists of a server-side or client-side request to the Firebase real time database to retrieve user information. This data can be displayed using NextJS server-side rendering or client-side rendering with React. Depending on the user's access level and permission settings, in addition to displaying user information, the user list may also include options to edit or delete user accounts.

Using NextJS and Firebase gives us some advantages when implementing a list of users in the ToDo app. NextJS server-side rendering can improve application performance by reducing the time it takes to load user data. Firebase Realtime Database provides a scalable and reliable backend for storing and synchronizing user data, ensuring that your user list is always up-to-date and accurate.

Overall, the user list is a useful feature for managing user accounts and permissions in the ToDo app. The use of NextJS and Firebase ensures the reliability, scalability and ease of use of this feature.

29



Built with NextJS and Firebase, the Add Task popup in the ToDo app is a common UI element that allows users to quickly create new tasks without leaving the current page. When the user clicks a button or performs a certain action, a modal or popup window appears that displays a form with fields for the task name, description, due date, and other relevant information.

When the user submits the form, the application sends a request to the Firebase Realtime Database to store the new task's data. Once a new task is added to the database, the popup can be dismissed and the user interface can be updated to show the new task. This can be achieved using NextJS server-side rendering or client-side rendering using React.

Using the pop-up window to add a task in the ToDo app can provide several benefits. First, it allows users to quickly and efficiently create new tasks without disrupting their current workflow. Second, it can help reduce clutter on the main application screen by keeping the task creation process contained in a separate window. Finally, the use of Firebase Realtime Database ensures that a new job is available to all users in real time, regardless of the device or location they are using.

Overall, the task add popup is a useful and user-friendly feature in a ToDo app built with NextJS and Firebase. Its use can help streamline the task creation process, improve the user experience, and ensure that the application is reliable and scalable.

**DFD:**

The above diagram illustrates the main components of the system and the data flow between them. The user interacts with the app through the User Interface component, which allows them to perform actions such as adding new ToDo tasks or deleting existing ones. These actions are then processed by the ToDo Manager component, which interacts with the database to store or delete the corresponding data.

Overall, this diagram depicts a simple ToDo app with two main functionalities: adding new tasks and deleting existing ones. More complex applications could include additional features such as editing tasks, setting due dates, or categorizing tasks by project or priority.

34

**Scope for further study**

There are several areas for further study when it comes to building a ToDo application in Next.js. Here are some suggestions:

1. **Authentication and Authorization:** Consider implementing user authentication and authorization for your ToDo application. This can involve using Next.js built-in authentication system or integrating a third-party authentication provider such as Auth0 or Firebase.

2. **Database Integration:** Next.js is highly flexible when it comes to data management. You can choose to integrate various databases such as MongoDB, MySQL, Postgres, or Firebase Realtime Database for storing and managing your ToDo application data.

3. **State Management:** As your application grows, you may need to consider managing state efficiently. Next.js offers several state management options, including React Context API, Redux, and MobX.

4. **UI Design and Styling:** To enhance the user experience, you can invest time in designing a user-friendly interface and styling your application. You can use CSS frameworks such as Bootstrap, Tailwind CSS, or Material UI to achieve a professional look and feel.

5. **Testing and Deployment:** Ensure that your ToDo application is well-tested and works as expected. Consider implementing automated tests using Jest and Enzyme or Cypress. Finally, deploy your application on a hosting platform such as Vercel, Netlify, or AWS.

6. **Accessibility:** Consider making your application accessible to people with disabilities. This can involve following the Web Content Accessibility Guidelines (WCAG) or using tools like axe-core to test your application for accessibility issues.

7. **Performance:** Ensure that your application performs well by optimizing it for speed and reducing unnecessary loading time. You can use tools like Lighthouse or Google Page Speed Insights to analyze your application's performance and implement improvements accordingly.

Overall, building a ToDo application in Next.js provides an excellent opportunity to learn about various web development concepts and technologies, and there are plenty of areas for further study and improvement.

Cost estimation for a TODO app can vary significantly based on several factors, including the complexity of the application, the technology stack used, the team's experience, and development location. Here's a simplified cost estimation for a basic TODO app along with considerations for future scope and an example data dictionary. Please note that these estimates are rough and can vary widely:

### Cost Estimation for a Basic TODO App:

1. \*\*Development Costs\*\*:

- Frontend Development (Next.js, React, etc.): $5,000 - $15,000

- Backend Development (Node.js, Express, etc.): $5,000 - $15,000

- Database Setup and Configuration: $1,000 - $3,000

- Total Development Cost: $11,000 - $33,000

2. \*\*Infrastructure and Hosting\*\*:

- Web Hosting (e.g., AWS, Heroku): $500 - $2,000/year

- Database Hosting (e.g., PostgreSQL, AWS RDS): $1,000 - $3,000/year

- Domain Name Registration: $10 - $50/year

- SSL Certificate: $50 - $200/year

- Total Infrastructure Cost (first year): $1,560 - $5,250

3. \*\*Maintenance and Support\*\*:

- Ongoing Development and Bug Fixes: $5,000 - $15,000/year

- Server Maintenance and Updates: $1,000 - $3,000/year

- Total Annual Maintenance Cost: $6,000 - $18,000/year

4. \*\*Biography Appendices\*\*:

- If you want to include a biography or "About Us" section, the cost would depend on content creation, design, and development. Estimate: $1,000 - $5,000.

5. \*\*Future Scope\*\*:

- Integration with Mobile Apps (iOS/Android): $10,000 - $30,000 per platform.

- User Authentication and Authorization: $3,000 - $10,000.

- Additional Features (e.g., task categorization, notifications): $5,000 - $15,000.

- Scaling for High Traffic: Costs may vary; it depends on usage patterns.

### Example Data Dictionary:

Here's a simple data dictionary for the TODO app:

1. \*\*Users Table\*\*:

- `user\_id` (Primary Key): Unique identifier for each user.

- `username`: User's username for authentication.

- `password`: Encrypted user password.

- `email`: User's email address.

2. \*\*Todo\_Items Table\*\*:

- `todo\_id` (Primary Key): Unique identifier for each TODO item.

- `user\_id` (Foreign Key): References the user who owns the TODO item.

- `title`: Title of the TODO item.

- `description`: Description of the TODO item.

- `created\_at`: Timestamp for when the TODO item was created.

3. \*\*Sessions Table\*\* (For User Authentication):

- `session\_id` (Primary Key): Unique session identifier.

- `user\_id` (Foreign Key): References the user associated with the session.

- `expiry\_time`: Timestamp for session expiration.

This data dictionary represents a simplified structure. In a production application, you would likely have more tables and fields, especially if you expand the functionality.

Keep in mind that the costs mentioned here are estimates and can vary based on your specific requirements, development practices, and the development team you choose. It's essential to discuss your project's scope and needs with experienced developers to get a more accurate cost estimate.

**Conclusion**

Our project is a modest yet powerful investment aimed at meeting your task management needs. We understand that task management can be a difficult and time-consuming process. That's why we've created a comprehensive solution that simplifies task management and helps you stay organized. We've integrated some user-friendly coding features and integrated a number of critical features that offer opportunities for further development. Using different hosting platforms gives you a perspective from a business perspective.

Our packages are designed to be powerful tools that meet all your productivity needs. Our goal is to provide a framework that allows users to effectively manage their tasks and stay organized. We know that staying on task is critical to your success, and our solutions are designed to meet this challenge. Our packages are perfect for students, professionals, entrepreneurs, and anyone who wants to prioritize work, avoid procrastination, and improve overall efficiency.

To achieve our goal, we have developed a project using NextJS and PWA with Firebase as backend and authentication system. This makes the system flexible and expandable in the future. Our ToDo app is very useful for managing tasks and is designed for those who want to manage various tasks and prioritize their work. Our app helps users avoid procrastination and improve their overall efficiency.

We believe that the To-Do app can be an essential tool for everyone who wants to achieve their goals. Our programs are designed to help you be more productive, manage your time effectively, and achieve your goals by staying organized and focused. We know that managing tasks can be difficult, especially if done manually. In this way, our project is designed to avoid the problems associated with manual processing and make the work management more accessible and convenient.

Our project aims to solve the challenges of task management by designing a computer system that is compatible with the existing system. We identified the drawbacks of the current system and developed a system that is more user-friendly and GUI-oriented. This ensures that users can easily navigate the system and manage their tasks efficiently.

In short, our project is a comprehensive solution designed to simplify task management and help users stay organized. We developed the project using NextJS & PWA with Firebase as backend and authentication system. Our solution is flexible and allows for future improvements. Our ToDo app is very useful for managing tasks and has been designed to suit people who want to prioritize their work, avoid procrastination and improve their overall efficiency. Our project aims to solve the challenges associated with task management and make task management more accessible and convenient.

36

**References**

∙ **https://nextjs.org/**

∙ **https://web.dev/progressive-web-apps/**

∙ **https://stackoverflow.com/**

∙ **https://miro.com/index/**

**END**