BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE

OF THE

MANONMANIAM SUNDARANAR UNIVERSITY

K. ARUN

REG.NO: 20192021506107



DEPARTMENT OF COMPUTER SCIENCE

ADITANAR COLLEGE OF ARTS AND SCIENCE

VIRAPANDIANPATNAM

THIRUCHENDUR

MAY-2022

ADITANAR COLLEGE OF ARTS AND SCIENCE



**DEPARTMENT OF COMPUTER SCIENCE**

**DECLARATION**

**K. ARUN**

**REG.NO: 20192021506107**

The Final year student of B.SC Computer Science of **ADITANAR COLLEGE OF ARTS AND SCIENCE** assure that the projecte entitled **‘Time Sheet Monitoring’** is a bonafide

Record done by **K. ARUN** in practical fulfilment of the requirement for the **BACHELOR OF SCIENCE** and the work was held during the academic year **2021-2022.**

**Place: Thiruchendur**

**Date:**

**Signature**

**(K. ARUN)**

ADITANAR COLLEGE OF ARTS AND SCIENCE



**DEPARTMENT OF COMPUTER SCIENCE**

**CERTIFICATE**

This to certify the project entitled ‘**TIME SHEET MONITORING’** is a bonafide work done by

**K. ARUN**

**REG.NO: 20192021506107**

Is in partial fulfillments in requirements for the **B.SC(Computer Science)** and the work was carried out under my super vision during academic year **2021-2022.**

**SIGNATURE OF HOD SIGNATURE OF GUIDE (Dr.C.VELAYUTHAM) (Dr.C.VELAYUTHAM)**

**Place: Thiruchendur**

**Date:**

**External Examer**

**1.**

**2.**

# Acknowledgement

I sincerely thank my parent for being with me and who I am today has only been possible thanks to their contributions and sacrifices. “Thank you so much for caring and loving me unconditionally”.

I express my deep gratitude for our company ‘Tigeen Computing Solutions’ to make this project happen

I express my deep gratitude to Dr. D. S. Mahendran M.Sc., M.Phil., Ph.D., P.B.D.C.S.A., PRINCIPAL, Aditanar college of art and science Tiruchendur, who provided facilities to carry out this project.

Also, I thank my heart Head of Department Dr. C. Velayutham M.Sc., M.Phil., Ph.D., P.G.D.C.A., Associate professor Department of Computer science for his invaluable guidance and consultation during process of this Project.

I also thanks to our department staff members,

Mr.R.Balakrishnan BE., M.Sc.,

Mrs.M. Preethi M.Sc., M.Phil.,

Ms.S. Thanu M.Sc.,

Ms.M. Brintha M.Sc.,

I also thanked my friends for helping me to complete this project.

**CONTENTS**

**TITLE PAGE NO**

1. **SYNOPSIS 6**
2. **ANALYSIS 8**
3. **PROJECT ANALYSIS 9**
4. **SYSTEM ANALYSIS 10**
5. **DFD’s 11**
6. **What is DFD 12**
7. **DFD Rules 13**
8. **Diagrams 14**
9. **TECH STACK 15**
10. **React JS 16**
11. **PHP 17**
12. **FILEMAKER Pro 18**
13. **CODES 19**
14. **SCREEN SHOTS 75**
15. **Future Modules**
16. **CONCLUSION 82**
17. **REFERENCES 83**

**SYNOPSIS**

# Synopsis

An abstract or summary of our project as follows,

The project ‘Time Sheet Monitoring’ is an application for admins to monitor their faculties works on the projects like, working hours, on which date, whom worked that task and its screenshots of the work.

Modern companies or organizations are wants to monitor their faculty works on each task modules. From this application organization’s admins can easily monitor their faculties task works.

ANALYSIS

PROJECT ANALYSIS

1. PROPOSED SYSTEM

The ‘Time Sheet Monitor’ is used to monitor the company employees works. The admins of the company or organization can easily monitor their employees work. Organization admins don’t wants to their employee to report their works in directly emplo, but admin can easily monitor their works by remote using this application

1. OBJECTIVE OF THE SYSTEM

The main objective or goal of this ‘Time Sheet Monitor’ is to help the company’s admin to monitor their faculties works from each modules in a project.

SYSTEM ANALYSIS

**HARWARE REQUIREMENTS:-**

* Intel core i3 or better
* 4GB RAM or even more
* Atleast 5gb free space in C drive

**SOFTWARE REQUIREMENTS:-**

* Windows\_10\_1704\_build or newest
* An Apache server
* Node Env above 16.0.0 version
* React version above 17.0.1
* Php version atleast 7.6.4
* Internet connection is required
* A Web Browser(Chrome recommended)

DFD’s

DATA FLOW DIAGRAMS:-

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO.

That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

**Rules for Data Flow Diagram**

**Data can not flow between two entities.**

Data flow must be from entity to a process or a process to an entity. There can be multiple data flows between one entity and a process.

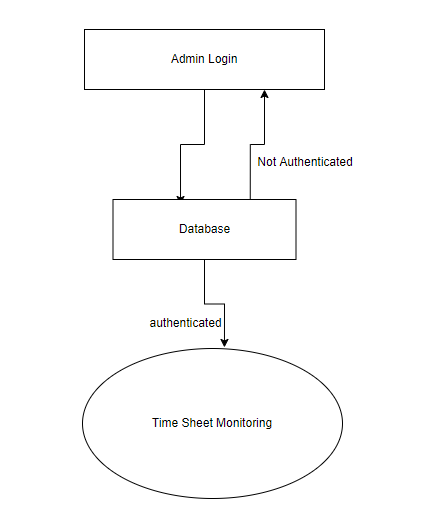
**Data can not flow between two data stores**

Data flow must be from data store to a process or a process to an data store. Data flow can occur from one data store to many process.

**Data can not flow directly from an entity to data store**

Data Flow from entity must be processed by a process before going to data store and vice versa.

**DATAFLOW DIAGRAMS:-**



**TECH STACK**

**RAECT JS**

React JS is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by **Meta** (formerly **Facebook**) and a community of individual developers and companies.

React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js.

However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

**PHP**

PHP is a general-purpose scripting language geared toward web development.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control.PHP code can also be directly executed from the command line.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on a variety of operating systems and platforms.

**FILEMAKER**

Claris FileMaker, previously known as FileMaker Pro, can be used to produce custom applications for enhancing your productivity and operations.

This workplace innovation platform is fast, robust and flexible. It also offers the benefit of working on both Windows and Mac operating systems and Android and iOS mobile devices.

Including MailChimp, QuickBooks and HubSpot, to name a few.Integration capabilities speed up development and typically reduce double and sometimestriple data entry across multiple tools*.*

Design is an essential element because the more professional your solution looks and feels, the more compelling it is to use. FileMaker is more than a database; it provides a graphical interface and security features that make it flexible and powerful.

CODES

FRONT-END CODES:

[Index.js]

**import** React **from** 'react';

**import** ReactDOM **from** 'react-dom';

**import** { createStore, combineReducers } **from** 'redux'

**import** {Provider} **from** 'react-redux'

**import** App **from** './App';

**import** userReducer **from** './store/reducers/userReducer';

**import** projectReducer **from** './store/reducers/projectReducer';

**import** './scss/main.scss'

**import** '../node\_modules/bootstrap/dist/js/bootstrap'

**import** './index.css'

**const** rootReducer = combineReducers({

**user**: userReducer,

**project**: projectReducer

})

**const** store = createStore(rootReducer)

ReactDOM.render(

  <React.StrictMode>

    <Provider store={store}>

      <App />

    </Provider>

  </React.StrictMode>,

  document.**getElementById**('root')

);

[App.js]

**import** React, { useState } **from** 'react';

**import** axios **from** 'axios';

**import** { connect } **from** 'react-redux'

**import** { AppContext } **from** './context';

**import** Shared **from** './components/shared/Shared';

**import** ListView **from** './components/Views/ListView/ListView';

**import** KanbanView **from** './components/Views/KanbanView/KanbanView';

**import** ViewChanger **from** './components/ViewChanger/ViewChanger';

**import** Login **from** './components/Login/Login';

**import** StatusView **from** './components/Views/StatusView/StatusView';

**import** \* **as** actions **from** "./store/actions"

**import** Spinner **from** './components/Spinner/Spinner';

**function** App(props) {

**const** [view, setView] = useState()

**const** [isAuth, setAuth] = useState(JSON.parse(localStorage.getItem("auth")))

**const** [project, setProject] = useState(false)

**const** [projects, setProjects] = useState([])

**const** [isLoad, setLoad] = useState(false)

**const** onViewChange = (viewType) **=>** {

    setView(viewType)

}

**const** loadProjects = (id) **=>** {

**let** url = process.env.REACT\_APP\_HOST + `/api.php?require=request&projectId=${id}`

    axios.get(url).then(res **=>** {

    setProjects(res.data)

    setLoad(false)

    }).catch(err **=>** {

    setLoad(true)

    console.log(err)

    })

}

**const** onProjectChange = (id) **=>** {

    setLoad(true)

    props.onProjectChange(id)

    loadProjects(id)

    setProject(true)

    setView('list')

}

**const** loginHandler = **async** (data) **=>** {

**let** host = process.**env**.REACT\_APP\_HOST

**let** url = `${host}/api.php`

**let** auth = false

**let** res = **await** axios.**get**(url, {

**params**: { ...**data**, **require**: "login" }

    })

    console.**log**(res)

    auth = res.**data**.auth

    localStorage.**setItem**("auth", auth)

    localStorage.**setItem**("userId", res.**data**.id || null)

**props**.onUserLogin(res.**data**.id)

    setAuth(auth)

**return** { **auth**, **msg**: res.**data**.result }

}

**const** logoutHandler = () **=>** {

    localStorage.**setItem**("auth", false)

    localStorage.**setItem**("userId", null)

    setAuth(false)

}

**const** value = {

**userId**: **props**.userId,

**onViewChange**,

**onProjectChange**,

**loginHandler**,

**logoutHandler**,

**projects**

}

**const** renderView = () **=>** {

**switch** (view) {

**case** 'list':

**return** <ListView taskData={projects} />;

**case** 'kanban':

**return** <KanbanView taskData={projects} />

**case** 'status':

**return** <StatusView onViewChange={onViewChange} />

**default**:

**break**;

    }

}

**const** renderApp = () **=>** {

**return** (<>

    <Shared view={view} />

    {project ? renderView() : null}

    <ViewChanger onClick={onViewChange} view={view} />

    </>)

}

**return** (

    <AppContext.Provider value={value}>

    <div className="container-fluid">

        {isLoad && <Spinner />}

        {isAuth ? renderApp() : <Login loginHandler={loginHandler} />}

    </div>

    </AppContext.Provider>

);

}

**const** mapStateToProps = (**state**) **=>** {

**return** {

**userId**: **state**.user.userId

}

}

**const** mapDispatchToProps = (**dispatch**) **=>** {

**return** {

**onUserLogin**: (**id**) **=>** dispatch(actions.setUserId(**id**)),

**onProjectChange**: (**id**) **=>** dispatch(actions.setProjectId(**id**))

}

}

**export** **default** connect(mapStateToProps,mapDispatchToProps)(App);

[Shared.js]

**import** React, { useContext } **from** 'react'

**import** { AppContext } **from** '../../context'

**import** AppHeader **from** './AppHeader/AppHeader'

**import** Header **from** './Header/Header'

**import** Row **from** './Row'

**import** Info **from** './User/Info/Info'

**import** Projects **from** './User/Projects/Projects'

**const** Shared = (**props**) **=>** {

**const** { onViewChange } = useContext(AppContext)

**let** ListViewHeader = ["Main Task", "Estimate Due Date", "Due Start", "Due End", "Duration", "Assignee", "Budget", "Status", "Work History"]

**let** KanbanViewHeader = ["Open", "In Progress", "To be Validate"]

**let** header = **props**.view === "kanban" ? KanbanViewHeader : ListViewHeader

**const** onClick = (**viewType**) **=>** {

        onViewChange(**viewType**)

    }

**return** (

        <>

            <AppHeader header="Client Access" />

            <Row>

                <Projects />

                <Info />

            </Row>

            {**props**.view !== "kanban" ? <Header

                onViewChange={onClick}

                header={{ **section**: header }} /> : null}

        </>

    )

}

**export** **default** Shared

[AppHeader.js]

**import** React **from** 'react'

**const** AppHeader = (**props**) **=>** {

**return** (

        <div className='row bg-success justify-content-center p-3 text-white h2'>{**props**.header}</div>

    )

}

**export** **default** AppHeader

[Projects.js]

**import** axios **from** 'axios'

**import** React, { useContext, useEffect, useState } **from** 'react'

**import** menuIcon **from** '../../../../assets/svg/burger\_list\_menu.svg'

**import** { AppContext } **from** '../../../../context'

**import** './Project.css'

**const** Projects = (**props**) **=>** {

**const** [projects, setProjects] = useState([])

**const** [dropDownText, setDropDownText] = useState("Project")

**const** { userId, onProjectChange } = useContext(AppContext)

**let** host = process.**env**.REACT\_APP\_HOST

**let** url = `${host}/api.php?id=${userId}&require=project`

    useEffect(() **=>** {

        axios.**get**(url).**then**(**res** **=>** {

            console.**log**(**res**.**data**)

            setProjects(**res**.**status** === 200 ? **res**.**data** : [])

        }).**catch**(**err** **=>** {

            console.**log**(**err**)

        })

    }, [url])

**const** onDropDownClick = (**text**) **=>** {

        setDropDownText(**text**)

    }

**return** (

        <div className='row col-5'>

            <img src={menuIcon} alt="not-available" className='col-1' />

            <a className="dropdown-toggle text-black h2 nav-link col-4" href="/" role="button" data-bs-toggle="dropdown">

                {dropDownText}

            </a>

            <ul className="dropdown-menu col-2" aria-labelledby="navbarDarkDropdownMenuLink">

                {projects.**map**(**project** **=>** <li key={**project**.projId}

                    onClick={() **=>** onDropDownClick(**project**.title)}>

                    <button className="dropdown-item h4" onClick={() **=>** onProjectChange(**project**.projId)}>

                        {**project**.title}

                    </button>

                </li>)}

            </ul>

        </div>

    )

}

**export** **default** Projects

[Info.js]

**import** React, { useContext } **from** 'react'

**import** userIcon **from** "../../../../assets/png/user\_icon.png"

**import** { AppContext } **from** '../../../../context'

**const** Info = (**props**) **=>** {

**const** { logoutHandler } = useContext(AppContext)

**return** (

        <div className='col-3 offset-4 d-flex justify-content-end'>

            <button className='btn btn-outline-success d-flex' onClick={logoutHandler}>

                <img src={userIcon}

                    className="mt-3 me-3"

                    style={{ **width**: "25px", **height**: "25px" }}

                    alt="not-available" />

                <h6 style={{ **width**: "100px" }} className='mt-3'>Log Out</h6>

            </button>

        </div>

    )

}

**export** **default** Info

[ViewChanger.js]

**import** React **from** 'react'

**import** gridIcon **from** '../../assets/svg/grid\_icon.svg'

**import** listIcon **from** '../../assets/svg/list\_icon.svg'

**const** ViewChanger = (**props**) **=>** {

**return** (

        <div style={{

**position**: "relative",

**height**: "60px",

**width**: "125px",

**top**: "80px",

**left**: "50%",

**marginBottom**: "150px"

        }} className="btn-group me-4" role="group">

            <input type="radio" checked={**props**.view === 'list'} onChange={() **=>** **props**.onClick('list')}

                className="btn-check" name="btnradio" id="btnradio1" autoComplete="off" />

            <label className="btn btn-outline-primary" htmlFor="btnradio1">

                <div><img src={listIcon} alt="not" /></div>

            </label>

            <input type="radio" onClick={() **=>** **props**.onClick('kanban')}

                className="btn-check" name="btnradio" id="btnradio2" autoComplete="off" />

            <label className="btn btn-outline-primary d-flex justify-content-center align-items-center"

                htmlFor="btnradio2">

                <div><img src={gridIcon} alt="not" /></div>

            </label>

        </div>

    )

}

**export** **default** ViewChanger

[ListView.js]

**import** React **from** 'react'

**import** Task **from** '../../Task/Task'

**const** ListView = (**props**) **=>** {

**return** (

        <div key={Math.**random**()}>

            {**props**.taskData.map((**data**, **idx**) **=>** <Task data={**data**} main key={**idx** + Math.**random**()}/>)}

        </div>

    )

}

**export** **default** ListView

[KanbanView.js]

**import** React **from** 'react'

**import** { connect } **from** 'react-redux'

**import** AddTask **from** '../../Modal/AddTask'

**import** KanbanCard **from** './KanbanCard/KanbanCard'

**const** KanbanView = (**props**) **=>** {

**return** (

        <div className='row justify-content-around'>

            <div className='col-lg-2 mt-4'>

                <h2 className='d-flex justify-content-center bg-danger'>Open</h2>

                {**props**.taskData.map((**card**, **idx**) **=>** **card**.status.msg === 'Open' ? <KanbanCard key={**idx**} theme="danger" data={**card**} /> : null)}

                <AddTask />

            </div>

            <div className='col-lg-2 mt-4'>

                <h2 className='d-flex justify-content-center bg-success'>In Progress</h2>

                {**props**.taskData.map((**card**, **idx**) **=>** **card**.status.msg === 'In-progress' ? <KanbanCard key={**idx**} theme="primary" data={**card**} /> : null)}

            </div>

            <div className='col-lg-2 mt-4'>

                <h2 className='d-flex justify-content-center bg-info'>To Be Validate</h2>

                {**props**.taskData.map((**card**, **idx**) **=>** **card**.status.msg === 'To-be-validate' ? <KanbanCard key={**idx**} theme="info" data={**card**} /> : null)}

            </div>

            <div className='col-lg-2 mt-4'>

                <h2 className='d-flex justify-content-center bg-success'>Validated</h2>

                {**props**.taskData.map((**card**, **idx**) **=>** **card**.status.msg === 'Validated' ? <KanbanCard key={**idx**} theme="primary" data={**card**} /> : null)}

            </div>

            <div className='col-lg-2 mt-4'>

                <h2 className='d-flex justify-content-center bg-info'>Client Review</h2>

                {**props**.taskData.map((**card**, **idx**) **=>** **card**.status.msg === 'Client-Review' ? <KanbanCard key={**idx**} theme="info" data={**card**} /> : null)}

            </div>

            <div className='col-lg-2 mt-4'>

                <h2 className='d-flex justify-content-center bg-success'>Closed</h2>

                {**props**.taskData.map((**card**, **idx**) **=>** **card**.status.msg === 'Closed' ? <KanbanCard key={**idx**} theme="success" data={**card**} /> : null)}

            </div>

        </div>

    )

}

**const** mapStateToProps = (**state**) **=>** {

**return**{

**projectId**: **state**.project.projectId

    }

}

**export** **default** connect(mapStateToProps)(KanbanView)

[StatusView.js]

**import** React, { useContext } **from** 'react'

**import** { AppContext } **from** '../../../context'

**import** Card **from** '../../Card'

**import** Header **from** '../../shared/Header/Header'

**import** Task **from** '../../Task/Task'

**const** StatusView = (**props**) **=>** {

**const** { projects } = useContext(AppContext)

**let** header = []

**const** groupTheArray = (**array**) **=>** {

**array**.forEach((**data**) **=>** {

**if** (!header.**includes**(**data**.status.msg)) {

                header.**push**(**data**.status.msg)

            }

        })

    }

    groupTheArray(projects)

**return** (

        <>

            <div className='container-fluid'>

                <Card>

                    <h6 className='d-flex'>Status

                        <button onClick={() **=>** **props**.onViewChange('list')}

                            className='ms-5 btn-close' />

                    </h6>

                </Card>

            </div>

            {header.**map**(**text** **=>** {

**return** (<>

                    <Header isArray text={**text**} header={{ **section**: "" }} />

                    {projects.map(**project** **=>**

**project**.status.msg === **text** ? <Task isMain data={**project**} /> : null)}

                </>)

            })}

        </>

    )

}

**export** **default** StatusView

[Task.js]

**import** React, { useState } **from** 'react'

**import** axios **from** 'axios'

**import** Badge **from** '../Badge'

**import** Subtask **from** './Subtask/Subtask'

**import** Modal **from** '../Modal/Modal'

**import** Card **from** '../Card'

**import** Info **from** './Subtask/Info'

**const** Task = (**props**) **=>** {

**let** data = **props**.data

**const** [subTask, setSubTask] = useState([])

**const** [isMain] = useState(**props**.main)

**const** onClick = (**id**) **=>** {

**if**(isMain){

**let** url = process.**env**.REACT\_APP\_HOST + `/api.php?require=task&rqstid=${**id**}`

            axios.**get**(url).**then**(**res** **=>** {

                setSubTask(**res**.**data**)

            }).**catch**(**err** **=>** {

                console.**log**(**err**)

            })

        }

    }

**return** (

        <div className='row m-1 mt-4 shadow' key={**props**.unique}>

            <nav className="navbar navbar-expand navbar-light bg-light">

                <div className="row collapse navbar-collapse p-2" id="navbarNavDropdown">

                    <ul className="navbar-nav justify-content-between h6">

                        <a className="navbar-brand nav-link dropdown-toggle me-0"

                            style={{ **width**: "50%", **overflow**: "auto" }}

                            data-bs-toggle="collapse"

                            href={"#collapseExample" + data.id} role="button"

                            onClick={() **=>** onClick(data.id)}>

                            {data.task}

                        </a>

                        <li>{data.dueDate}</li>

                        <li>{data.dateRecieved}</li>

                        <li>{data.dueEnd}</li>

                        <li>{data.duration}</li>

                        <li>{data.assignee}</li>

                        <li>{data.budget}</li>

                        <Badge text={data.status.msg} color={data.status.color} />

                        <Modal title={data.task} id="taskModal" />

                    </ul>

                </div>

            </nav>

            <div className="collapse ps-0"

                id={"collapseExample" + data.id}

                style={{ **position**: 'relative', **left**: "-10px", **top**: "10px" }}>

                <ul>

                    {**props**.main ? <Subtask data={subTask} /> : <Card><Info /></Card>}

                </ul>

            </div>

        </div >

    )

}

**export** **default** Task

[KanbanCard.js]

**import** React, { useState } **from** 'react'

**import** axios **from** 'axios'

**import** imageIcon **from** '../../../../assets/png/image\_icon.png'

**import** calanderIcon **from** '../../../../assets/png/calander\_icon.png'

**import** TaskInfo **from** '../../../Task/Info/TaskInfo'

**import** './Kanban.css'

**import** TaskInfoModal **from** '../../../Modal/TaskInfoModal'

**const** KanbanCard = (**props**) **=>** {

**const** [task, setTask] = useState(**props**.data)

    console.**log**(**props**.data.id, task)

**const** onCardClick = (**id**) **=>** {

        console.**log**("card clicked", **id**)

**let** url = process.**env**.REACT\_APP\_HOST + `/api.php?require=task&rqstid=${**id**}`

        axios.**get**(url).**then**(**res** **=>** {

            console.**log**("card data" ,**res**.**data**)

            setTask(**res**.**data**)

        }).**catch**(**err** **=>** {

            console.**log**(**err**)

        })

    }

**return** (

        <>

            <div className={'p-4 mt-3 mb-3 callout callout-' + **props**.theme}

            style={{ **cursor**: "pointer" }}

            data-bs-toggle="modal"

            data-bs-target="#kanbanCard"

            onClick={() **=>** onCardClick(**props**.data.id)}>

                <span className='h5'>{**props**.data.task}</span>

                <div className='d-flex justify-content-around links'>

                    <img src={imageIcon} alt="not" width="25" className='screenshot-img' />

                    <div className='calander d-inline-block'>

                        <img src={calanderIcon} alt="not" width="25" className='calander-img' />

                        <span className='date'>{**props**.data.dateRecieved}</span>

                    </div>

                </div>

            </div>

            <TaskInfoModal>

                {<TaskInfo task={task}/> }

            </TaskInfoModal>

        </>

    )

}

**export** **default** KanbanCard

[Login.js]

**import** React, { useRef, useState } **from** 'react'

**import** Error **from** '../Error'

**import** Spinner **from** "../Spinner/Spinner"

**import** './Login.css'

**const** Login = (**props**) **=>** {

**let** userNameRef = useRef()

**let** passwordRef = useRef()

**const** [error, setError] = useState({ **msg**: "No Error", **color**: "danger", **isError**: false })

**const** [loading, setLoading] = useState(false)

**const** onLogin = () **=>** {

**let** username = userNameRef.**current**.value

**let** password = passwordRef.**current**.value

        setLoading(true)

        setError({

**msg**: "",

**color**: "warning",

**isError**: false

        })

**let** data = { **username**, **password** }

**if** (username && password) {

**props**.loginHandler(data).then(**res** **=>** {

**if** (!**res**.auth) {

                    setError({

**msg**: **res**.msg,

**color**: "warning",

**isError**: true

                    })

                    setLoading(false)

                }

            })

        } **else** {

            setLoading(false)

            setError({

**msg**: "All fields are required!",

**color**: "danger",

**isError**: true

            })

        }

    }

**return** (

        <div className='container'>

            <div className='row'>

                <div className='col-md-4 offset-md-4 form d-grid'>

                    {loading === true ? <Spinner /> : null}

                    {error.**isError** ? <Error {...error} /> : null}

                    <h2 className='p-3' style={{ **justifySelf**: "center" }}>Login</h2>

                    <div className="input-group mb-3">

                        <span className="input-group-text" id="basic-addon1">@</span>

                        <input type="text" required className="form-control" placeholder="Username"

                            ref={userNameRef} />

                    </div>

                    <div className="input-group mb-3">

                        <span className="input-group-text bi bi-key" id="basic-addon1"></span>

                        <input type="password" required className="form-control" placeholder="Password" ref={passwordRef} />

                    </div>

                    <div className="offset-md-5">

                        <button className="btn btn-primary bi bi-send"

                            onClick={onLogin}

                            type="submit"> Login</button>

                    </div>

                </div>

            </div>

        </div>

    )

}

**export** **default** Login

[TaskInfo.js]

**import** React, { useEffect, useState } **from** 'react'

**import** axios **from** 'axios'

**import** Badge **from** '../../Badge'

**import** Card **from** '../../Card'

**import** userLogo **from** '../../../assets/png/user\_icon.png'

**import** calanderLogo **from** '../../../assets/png/calander\_icon.png'

**import** timerLogo **from** '../../../assets/png/timer\_icon.png'

**import** TaskInfoModal **from** "../../Modal/Kanban/TaskInfo/TaskInfo.js"

**import** './TaskInfo.css'

**const** TaskInfo = (**props**) **=>** {

**let** [tasks, setTasks] = useState([])

    useEffect(() **=>** {

**let** url = process.**env**.REACT\_APP\_HOST + `/api.php?require=task&rqstid=${**props**.task.id}`

        console.**log**("getting", url)

        axios.**get**(url).**then**(**res** **=>** {

            console.**log**(**res**.**data**)

            setTasks(**res**.**data**)

        }).**catch**(**err** **=>** {

            console.**log**(**err**)

        })

    }, [])

    console.**log**(**props**.task)

**return** (

        <React.Fragment key={Math.**random**()}>

            <Card>

                <p className='h5'>{**props**.task.task}</p>

                <table className="table table-borderless mt-5">

                    <thead>

                        <tr>

                            <th scope="col"><Badge color="danger" text="Open" /></th>

                            <th scope="col">

                                <img src={userLogo} width="20" alt="not-available" /> Assignee

                            </th>

                            <th scope="col">

                                <img src={calanderLogo} width="20" alt="not-available" /> Due Date

                            </th>

                            <th scope="col">

                                <img src={timerLogo} width="20" alt="not-available" /> Duration

                            </th>

                        </tr>

                    </thead>

                    <tbody>

                        <tr>

                            <th scope="row"></th>

                            <td>{**props**.task.assign}</td>

                            <td>{**props**.task.dueDate}</td>

                            <td>{**props**.task.duration}</td>

                        </tr>

                    </tbody>

                </table>

            </Card>

            <p className='h5 mt-5'>Sub Task</p>

            <table className="table table-borderless mt-5">

                <tbody>

                    {tasks.**map**((**task**, **idx**) **=>**{

**return**(

                            <tr className=''>

                                <React.Fragment key={**idx**}>

                                    <td>{**task**.task}</td>

                                    <td>

                                        <img src={userLogo} width="20" alt="not-available" /> {**task**.assignee}

                                    </td>

                                    <td>

                                        <img src={timerLogo} width="20" alt="not-available" /> {**task**.dueEnd}

                                    </td>

                                    <td>

                                        <Badge text={**task**.status.msg} color={**task**.status.color}/>

                                    </td>

                                    <td>

                                        <TaskInfoModal />

                                    </td>

                                </React.Fragment>

                            </tr>

                        )

                    })}

                </tbody>

            </table>

        </React.Fragment>

    )

}

**export** **default** TaskInfo

[TaskInfoModal.js]

**import** React,{useState} **from** 'react'

**import** { Button, Modal } **from** 'react-bootstrap'

**const** TaskInfoModal = (**props**) **=>** {

**const** [show, setShow] = useState(false);

**const** handleClose = () **=>** setShow(false);

**const** handleShow = () **=>** setShow(true);

**return** (

        <>

        <Button variant="primary" onClick={handleShow}>

            View

        </Button>

        <Modal show={show} onHide={handleClose} backdrop='static' centered size='lg'>

            <Modal.Header className='bg-success'>

                <h3>Request</h3>

            </Modal.Header>

            <Modal.Body>{**props**.children}</Modal.Body>

            <Modal.Footer>

                <Button variant="secondary" onClick={handleClose}>

                    Close

                </Button>

            </Modal.Footer>

        </Modal>

        </>

    )

}

**export** **default** TaskInfoModal

[ScreenShot.js]

**import** React,{useState} **from** 'react'

**import** { Button, Modal } **from** 'react-bootstrap'

**const** ScreenShot = (**props**) **=>** {

**const** [show, setShow] = useState(false);

**const** handleClose = () **=>** setShow(false);

**const** handleShow = () **=>** setShow(true);

**return** (

        <>

        <Button variant="primary" onClick={handleShow}>

            View

        </Button>

        <Modal show={show} onHide={handleClose} backdrop='static' centered size='lg'>

            <Modal.Header className='bg-success'>

                <h3>Develop the mockup</h3>

            </Modal.Header>

            <Modal.Footer>

                <Button variant="secondary" onClick={handleClose}>

                    Close

                </Button>

            </Modal.Footer>

        </Modal>

        </>

    )

}

**export** **default** ScreenShot

[Modal.js]

**export** **default** **function** Modal(**props**) {

**return** (<>

        {!**props**.noBtn ? <button type="button" className="btn btn-primary" data-bs-toggle="modal" data-bs-target={`#${**props**.id}`}>

            View

        </button> : null

        }

        <div className="modal fade" id={**props**.id}

            data-bs-backdrop="static"

            data-bs-keyboard="false"

            tabIndex="-1">

            <div className="modal-dialog modal-lg modal-dialog-centered modal-dialog-scrollable">

                <div className="modal-content w-100">

                    <div className="modal-header bg-success">

                        <h5 className="modal-title">{**props**.title}</h5>

                        <button type="button" className="btn-close" data-bs-dismiss="modal"></button>

                    </div>

                    <div className="modal-body p-4">

                        {**props**.children}

                    </div>

                </div>

            </div>

        </div>

    </>);

}

[userReducer.js]

**import** \* **as** actions **from** '../actions'

**const** initialState = {

**userId**: JSON.**parse**(localStorage.**getItem**("userId")) || null

}

**const** userReducer = (**state** = initialState, **action**) **=>** {

**switch** (**action**.type) {

**case** actions.SET\_USER\_ID:

**return** {

                ...**state**,

**userId**: **action**.userId

            }

**default**:

**return** **state**

    }

}

**export** **default** userReducer

[projectReducer.js]

**import** \* **as** actions **from** "../actions"

**const** initialState = {

**projectId**: null,

**requstId**: null,

**taskId**: null

}

**const** projectReducer = (**state** = initialState, **action**) **=>** {

**switch**(**action**.type){

**case** actions.SET\_PROJECT\_ID:

**return** {

                ...**state**,

**projectId**: **action**.projectId

            }

**case** actions.SET\_REQUEST\_ID:

**return** {

                ...**state**,

**requstId**: **action**.requstId

            }

**case** actions.SET\_TASK\_ID:

**return** {

                ...**state**,

**taskId**: **action**.taskId

            }

**default**:

**return** **state**

    }

}

**export** **default** projectReducer

[actions.js]

**export** **const** USER\_ID = "USERID"

**export** **const** PROJECT\_ID = "ProjectId"

**export** **const** REQUEST\_ID = "requestId"

**export** **const** TASK\_ID = "TaskId"

**export** **const** SET\_USER\_ID = "setUserId"

**export** **const** SET\_PROJECT\_ID = "setProjectId"

**export** **const** SET\_REQUEST\_ID = "setRequestId"

**export** **const** SET\_TASK\_ID = "setTaskId"

**export** **const** userId = () **=>** ({ **type**: USER\_ID })

**export** **const** projectId = () **=>** ({ **type**: PROJECT\_ID })

**export** **const** requestId = () **=>** ({ **type**: REQUEST\_ID })

**export** **const** taskId = () **=>** ({ **type**: TASK\_ID })

**export** **const** setUserId = (**id**) **=>** ({ **type**: SET\_USER\_ID, **userId**: **id** })

**export** **const** setProjectId = (**id**) **=>** ({ **type**: SET\_PROJECT\_ID, **projectId**: **id** })

**export** **const** setRequestId = (**id**) **=>** ({ **type**: SET\_REQUEST\_ID, **requestId**: **id** })

**export** **const** setTaskId = (**id**) **=>** ({ **type**: SET\_TASK\_ID, **taskId**: **id** })

[Header.js]

**import** React **from** 'react'

**const** Header = (**props**) **=>** {

**let** header = **props**.header.section

**let** style = {

**cursor**: "pointer",

**width**: "85px"

    }

**let** thStyle = {

**width**: "85px"

    }

**let** tableStyle = {

**display**: "inline-grid",

**marginBottom**: "0px"

    }

**const** onClick = () **=>** {

**props**.onViewChange('status')

    }

**const** isStatus = (**title**) **=>** **title** === "Status"

**return** (

        <div className='row mt-4 bg-success p-2 m-1 align-items-center'>

            <table className='table' style={tableStyle}>

                <thead className='d-flex list-unstyled h5 text-white justify-content-between align-items-center'>

                    {!**props**.isArray ?

                    header.map((**\_**, **idx**) **=>**

                    <tr key={**idx**} style={**\_** === 'Status' ? style : **idx** === 0 ? {**width**: "50%"} : thStyle}>

                        <td style={{**borderBottomWidth**: "0px"}} onClick={isStatus(**\_**) ? onClick : null}

                            >

                            {**\_**}

                        </td>

                    </tr>

                    ) :

                    <tr><td style={thStyle}>{**props**.text}</td></tr>}

                </thead>

            </table>

        </div>

    )

}

**export** **default** Header

[SubTask.js]

**import** React **from** 'react'

**import** Card **from** '../../Card'

**import** Task **from** '../Task'

**const** Subtask = (**props**) **=>** {

**let** header = ["Task", "Estimate Due Date", "Due Start", "Due End", "Duration", "Assignee", "Budget", "Status", "Work History"]

**let** taskData = **props**.data

**let** thStyle = {

**width**: "85px"

    }

**return** (

        <Card>

            <table className='table'>

                <thead>

                    {header.**map**((**title**, **idx**) **=>**  <td style={**idx** === 0 ? {**width**: "50%"} : thStyle} key={**idx**}>{**title**}</td>)}

                </thead>

            </table>

            {taskData.map((**data**, **idx**) **=>** <Task data={**data**} main={false} key={**idx** + Math.**random**()} />)}

        </Card>

    )

}

**export** **default** Subtask

[Info.js]

**import** React **from** 'react'

**import** Modal **from** '../../Modal/Modal'

**import** Header **from** '../../shared/Header/Header'

**const** Info = (**props**) **=>** {

**let** header = ["Date", "Start Time", "End Time", "Duration", "Description", "Work History"]

**return** (

        <>

            <table className='table'>

                <thead>

                    {header.**map**((**title**, **idx**) **=>** <th key={**idx**}>{**title**}</th>)}

                </thead>

            </table>

            <Header header={{ **section**: ["MS", "Total 10:10:25"] }} />

            <table className="table">

                <tbody style={{ **textAlign**: "center" }}>

                    <tr>

                        <td>01-01-2022</td>

                        <td>10:30:55</td>

                        <td>10:30:55</td>

                        <td>10:30:55</td>

                        <td>Develop the form page</td>

                        <td>

                            <Modal title="Develop the form page" id="info" />

                        </td>

                    </tr>

                    <tr>

                        <td>01-01-2022</td>

                        <td>10:30:55</td>

                        <td>10:30:55</td>

                        <td>10:30:55</td>

                        <td>Develop the form page</td>

                        <td>

                            <Modal title="Develop the form page" id="info" />

                        </td>

                    </tr>

                </tbody>

            </table>

        </>

    )

}

**export** **default** Info

PHP Codes

<?php

include "config.php";

header("Access-Control-Allow-Origin:\*");

//header("content-type:application/text");

$json = array();

$findCommand = $fm->newFindCommand('PHP\_\_CON');

$method = $\_SERVER["REQUEST\_METHOD"];

switch($method){

// case "POST":

// echo "GET method";

// if(isset($\_GET[""]))

// $username = $\_GET["username"];

// $query = "SELECT \* FROM table WHERE username='$username'";

// echo ExecuteSQL ( "SELECT Department FROM Employees WHERE EmpID = 1", "", "" );

// break;

case "GET":

$require = $\_GET["require"];

switch ($require) {

case 'login':

$username = $\_GET["username"];

$password = $\_GET["password"];

$findCommand->addFindCriterion('Username','=='.$username);

$result = $findCommand->execute();

if(FileMaker::isError($result)){

$temp = [

'auth' => false,

'result' => "Invalid Username",

'status' => 0

];

echo json\_encode($temp);

}

else{

$findCommand->addFindCriterion('Password','=='.$password);

$result = $findCommand->execute();

if(FileMaker::isError($result)){

$temp = [

'auth' => false,

'result' => "Invalid password",

'status' => 0

];

echo json\_encode($temp);

}

else{

$record = $result -> getFirstRecord();

$conid = $record->getField("\_\_kp\_\_ConId\_\_lsan");

$temp = [

'auth' => true,

'result' => "Success",

'status' => 1,

'id' => "$conid"

];

echo json\_encode($temp);

}

}

break;

case 'project':

$data = array();

$conid = $\_GET["id"];

$findCommand = $fm->newFindCommand('PHP\_\_PROJ');

$findCommand->addFindCriterion('\_kf\_\_ConId\_\_lsxn','=='.$conid);

$result = $findCommand->execute();

$records = $result -> getRecords();

$count = count($records);

foreach($records as $record)

{

$title = $record -> getField("TitreAffaire");

$projid = $record -> getField("\_\_kp\_\_ProjId\_\_lsxn");

$userid = $record -> getField("\_kf\_\_UsrId\_current\_\_gsxn");

$status = $record -> getField("IsActive");

$temp = [

'title' => "$title",

'projId' => "$projid",

'userId' => "$userid",

'status' => "$status"

];

array\_push($data,$temp);

}

echo json\_encode($data);

break;

case 'request':

$data = array();

$projid = $\_GET["projectId"];

$findCommand = $fm->newFindCommand('PHP\_\_RQST');

$findCommand->addFindCriterion('\_kf\_\_ProjId\_\_lsxn','=='.$projid);

$result = $findCommand->execute();

$records = $result -> getRecords();

// print\_r($records);

// $count = count($records);

foreach($records as $record)

{

$rqstId = $record -> getField('\_\_kp\_\_RqstId\_\_lsan');

$rqstPrj = $record -> getField('IsActive');

$recieved = $record -> getField('Date\_recieved');

$target = $record -> getField('Date\_target');

$name = $record -> getField('Name');

$des = $record -> getField('Description');

$statusId = $record -> getField('IsActive');

$assign = $record -> getField('\_kf\_\_UserId\_\_recievedby\_\_lsxn');

$statusid = $record -> getField('\_kf\_Tsk\_StatusId\_\_lsxn');

$duration = $record -> getField('Worked\_Duration');

$assigns = array();

switch ($statusId) {

case '1':

$status = 'Open';

$color = 'danger';

break;

case '2':

$status = 'In progress';

$color = 'success';

break;

case '3':

$status = 'To be validate';

$color = 'info';

break;

case '4':

$status = 'Validated';

$color = 'danger';

break;

case '5':

$status = 'Client Review';

$color = 'danger';

break;

case '6':

$status = 'Closed';

$color = 'success';

break;

}

$assignees = explode("\n", $assign);

foreach($assignees as $assignee){

switch($assignee){

case '230':

$assign = 'AB';

break;

case '246':

$assign = 'ADM';

break;

case '264':

$assign = 'BB';

break;

case '270':

$assign = 'CM';

break;

case '':

$assign = 'DB';

break;

case '':

$assign = 'DS';

break;

case '':

$assign = 'FB';

break;

case '':

$assign = 'HM';

break;

case '':

$assign = 'JJ';

break;

case '':

$assign = 'JR';

break;

case '':

$assign = 'KK';

break;

case '':

$assign = 'KR';

break;

case '':

$assign = 'KS';

break;

case '':

$assign = 'LP';

break;

case '':

$assign = 'LS';

break;

case '':

$assign = 'MN';

break;

case '':

$assign = 'MS';

break;

case '':

$assign = 'SA';

break;

case '':

$assign = 'SP';

break;

case '':

$assign = 'VA';

break;

}

array\_push($assigns, $assign);

}

$temp = [

'id' => $rqstId,

'task' => $name,

'status' => [

'msg' => $status,

'color' => $color,

],

'dateRecieved' => $recieved,

'dueEnd' => $target,

'dueDate' => $target,

'duration' => $duration,

'assign' => $assigns,

'rqstProj' => $rqstPrj,

'statusid' => $statusId,

];

array\_push($data, $temp);

}

echo json\_encode($data);

// echo sizeof($data);

break;

case 'task';

$data =array();

$rqstid = $\_GET["rqstid"];

$findCommand = $fm->newFindCommand('PHP\_\_RQST');

$findCommand->addFindCriterion('\_\_kp\_\_RqstId\_\_lsan','=='.$rqstid);

$result = $findCommand->execute();

$records = $result->getRecords();

foreach($records as $record){

$status = $record -> getField('IsActive');

$relatedSet = $record->getRelatedSet('rqst\_\_TSK');

foreach ($relatedSet as $relatedRow) {

$taskid = $relatedRow->getField('rqst\_\_TSK::\_\_kp\_\_TskId\_\_lsan');

$daterecvd = $relatedRow->getField('rqst\_\_TSK::Date\_Recevied');

$datedue = $relatedRow->getField('rqst\_\_TSK::Date\_Due');

$desc = $relatedRow->getField('rqst\_\_TSK::description');

$assign = $relatedRow -> getField('Assigned to');

$ex\_duration = $relatedRow->getField('rqst\_\_TSK::Duration\_expected');

$worked\_duration = $relatedRow->getField('rqst\_\_TSK::Duration\_worked');

$statusId = $relatedRow->getField('rqst\_\_TSK::\_kf\_Tsk\_StatusId\_\_lsxn');

switch ($statusId) {

case '1':

$status = 'Open';

$color = 'danger';

break;

case '2':

$status = 'In progress';

$color = 'success';

break;

case '3':

$status = 'To be validate';

$color = 'info';

break;

case '4':

$status = 'Validated';

$color = 'danger';

break;

case '5':

$status = 'Client Review';

$color = 'danger';

break;

case '6':

$status = 'Closed';

$color = 'success';

break;

}

$temp = [

'id' => "$taskid",

'dueDate' => "$ex\_duration",

'dueStart' => "$daterecvd",

'dueEnd' => "$datedue",

'status' => [

'msg' => $status,

'color' => $color,

],

'assign' => $assign,

'task' => "$desc",

'duration' => "$worked\_duration",

'statusId' => "$statusId"

];

array\_push($data,$temp);

}

echo json\_encode($data);

}

break;

case 'details':

$data = array();

$taskid = $\_GET["taskid"];

$findCommand = $fm->newFindCommand('PHP\_\_SCREENSHOT');

$findCommand->addFindCriterion('\_kf\_\_TskId\_\_lsxn','=='.$taskid);

$result = $findCommand->execute();

$records = $result->getRecords();

foreach($records as $record){

$name = $record -> getField('screenshot\_\_USR::Nom\_Prenom\_Societe');

$findCommand->addFindCriterion('screenshot\_\_USR::Nom\_Prenom\_Societe','=='.$name);

$result = $findCommand->execute();

$record = $result->getRecords();

foreach($records as $rec){

$date = $rec -> getField('Date');

$duration = $rec -> getField('Screenshot\_time');

$title = $rec -> getField('screenshot\_\_tsk\_\_PROJ::TitreAffaire');

$temp = [

'name' => $name,

'task:' => [[

'date' => $date,

'duration' => $duration,

'description' => $title

]]

];

array\_push($data,$temp);

}

echo json\_encode($data);

}

break;

}

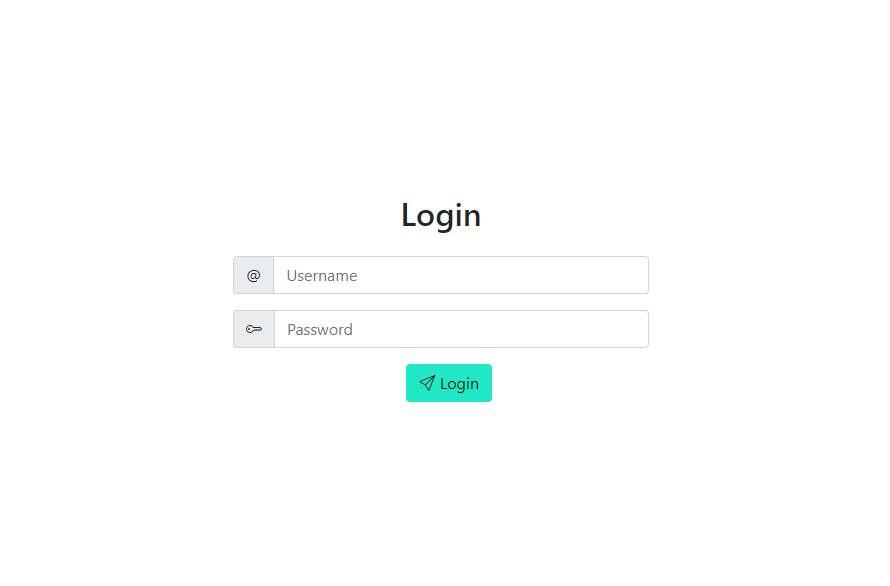
}

?>

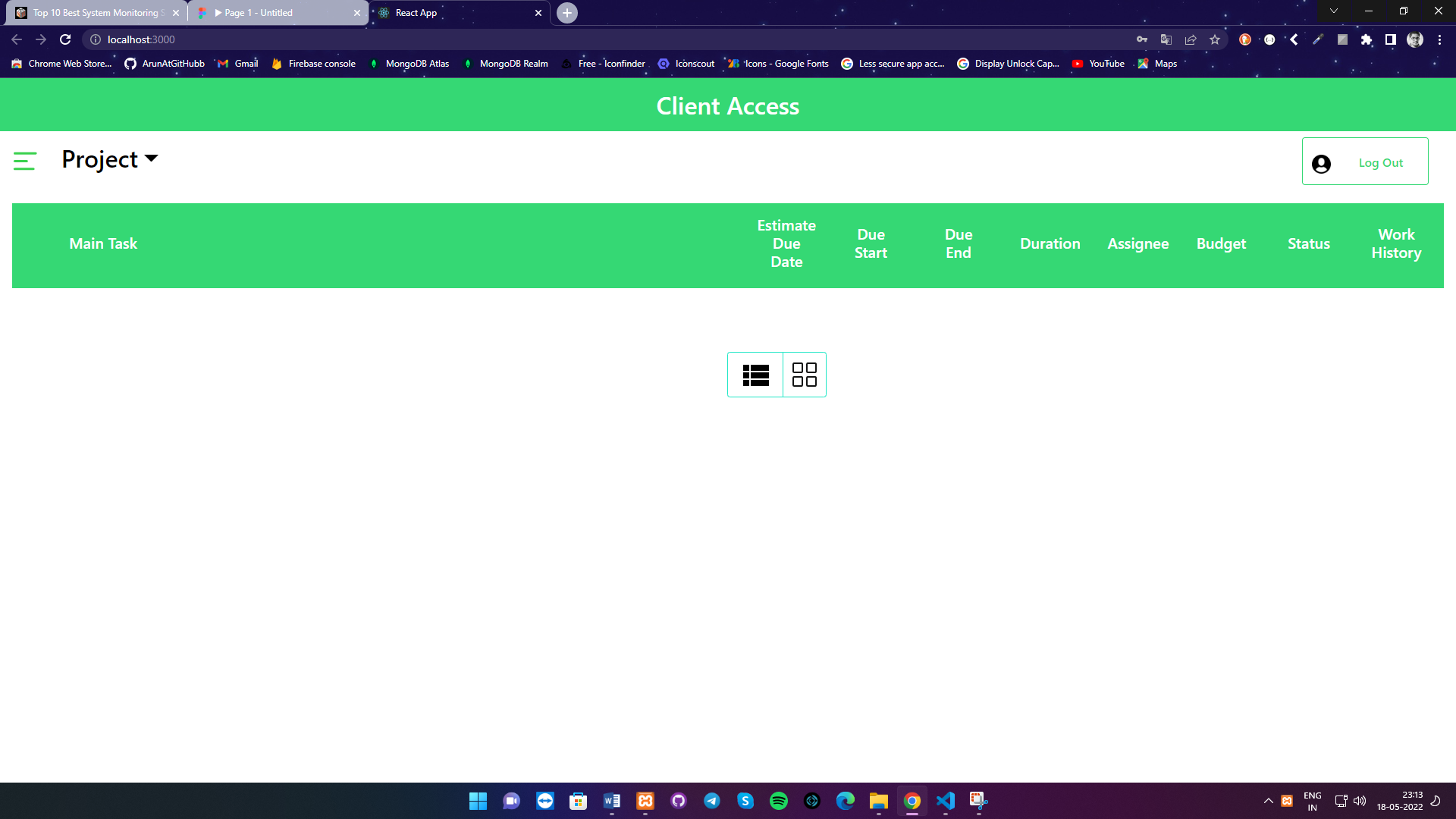
SCREEN

SHOTS

**Login Screen**

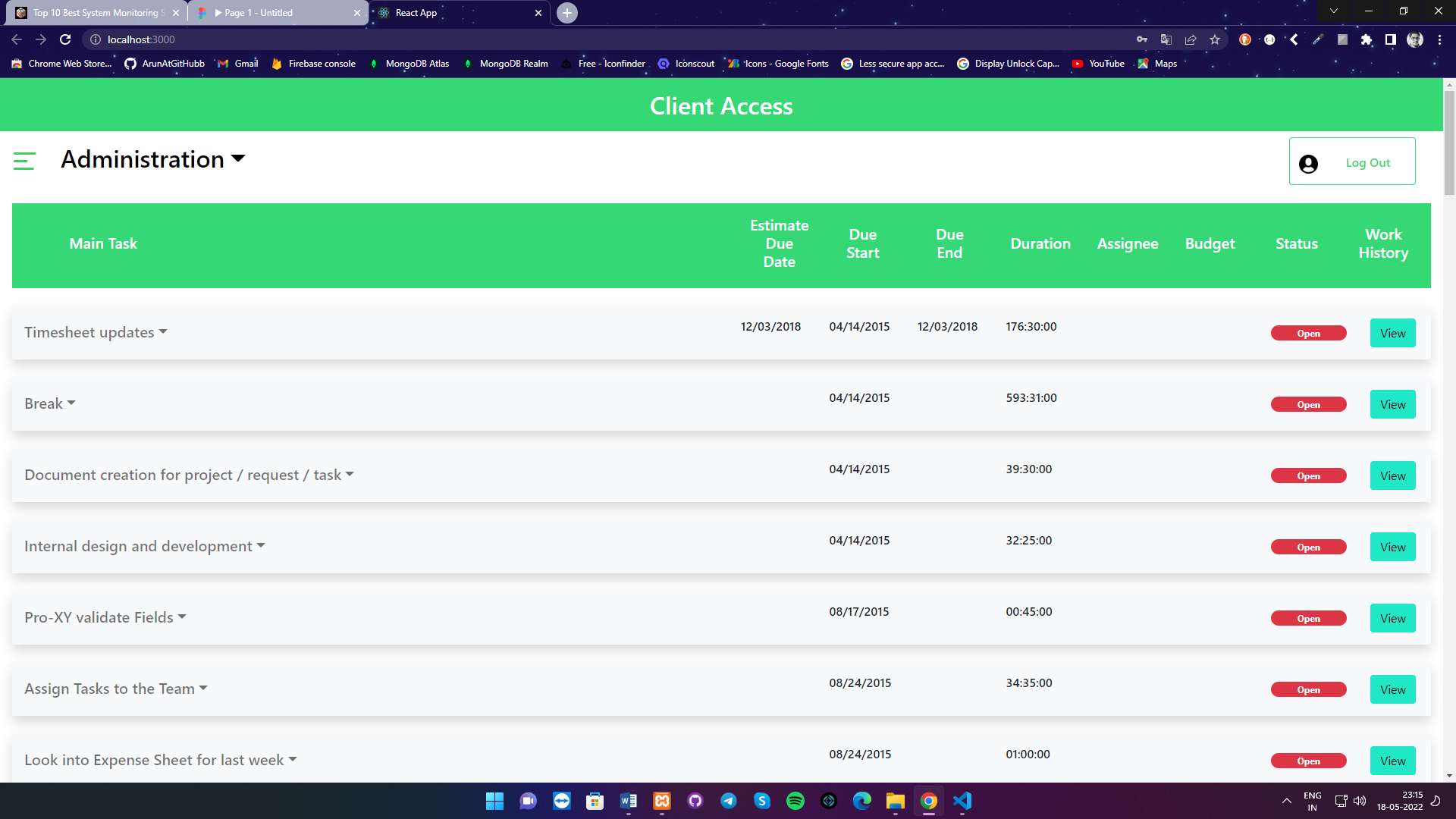


**Home Screen**



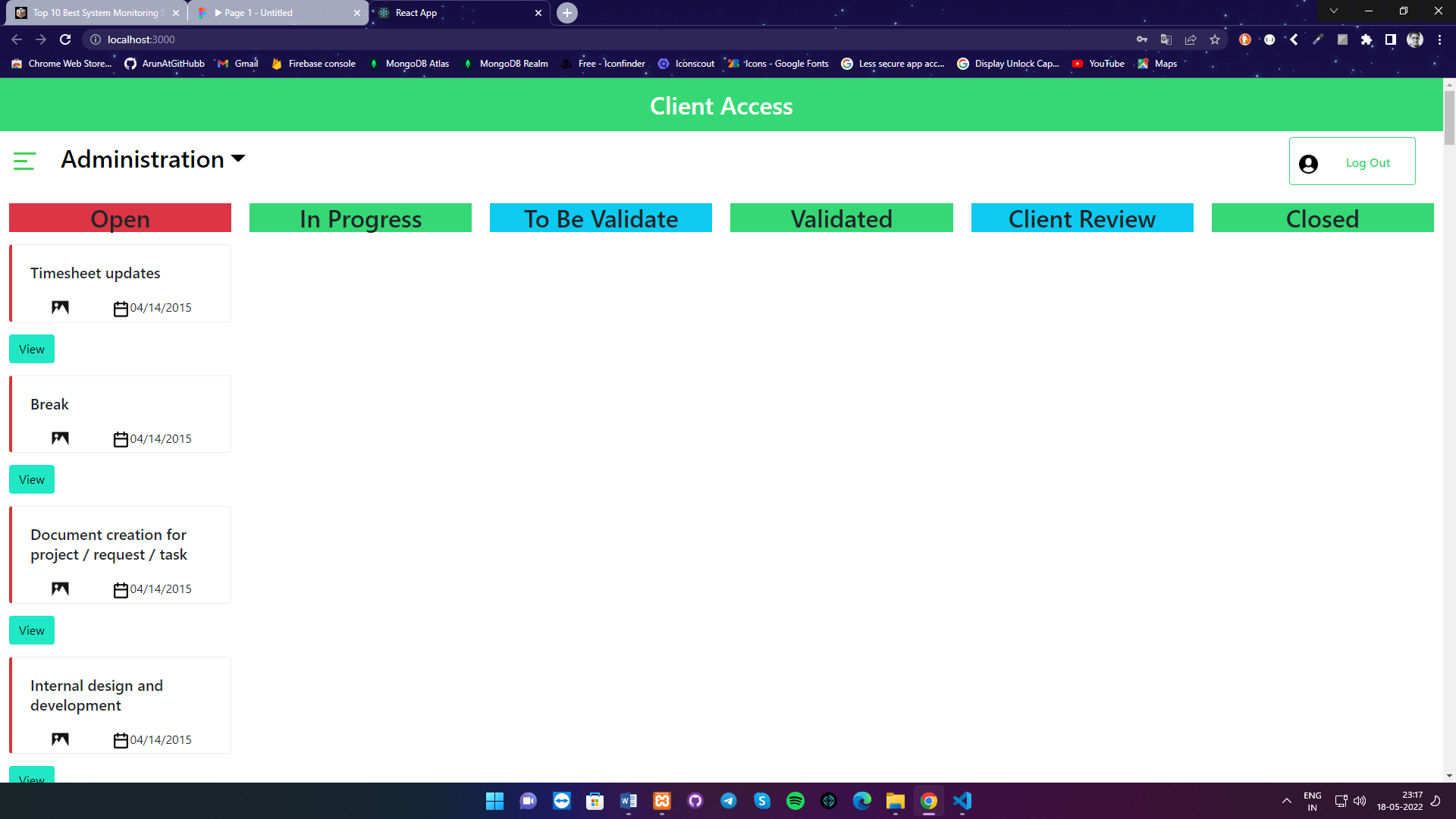
**After selecting Project**

**(List View)**



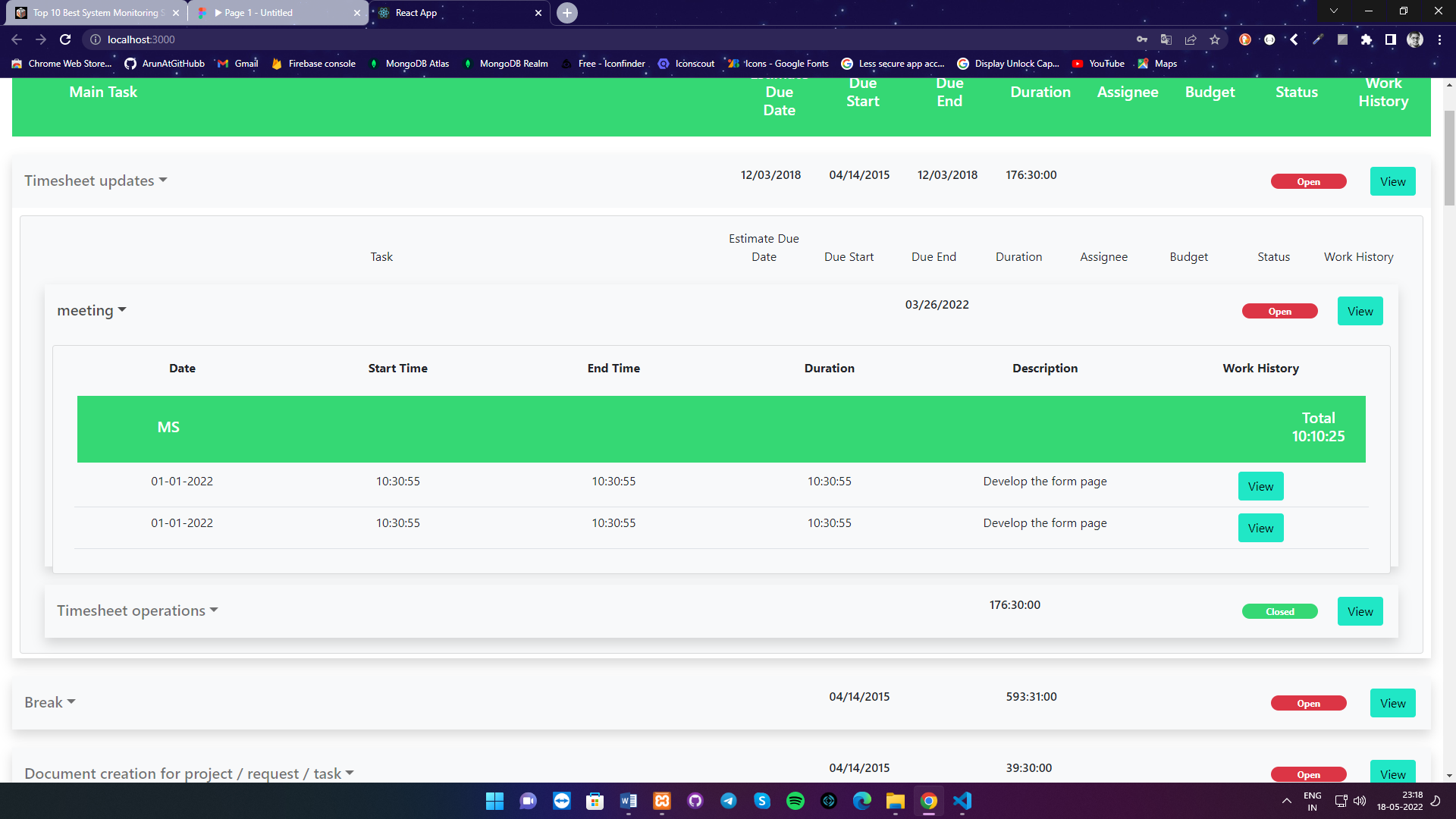
**After Selecting Project**

**(Kanban View)**



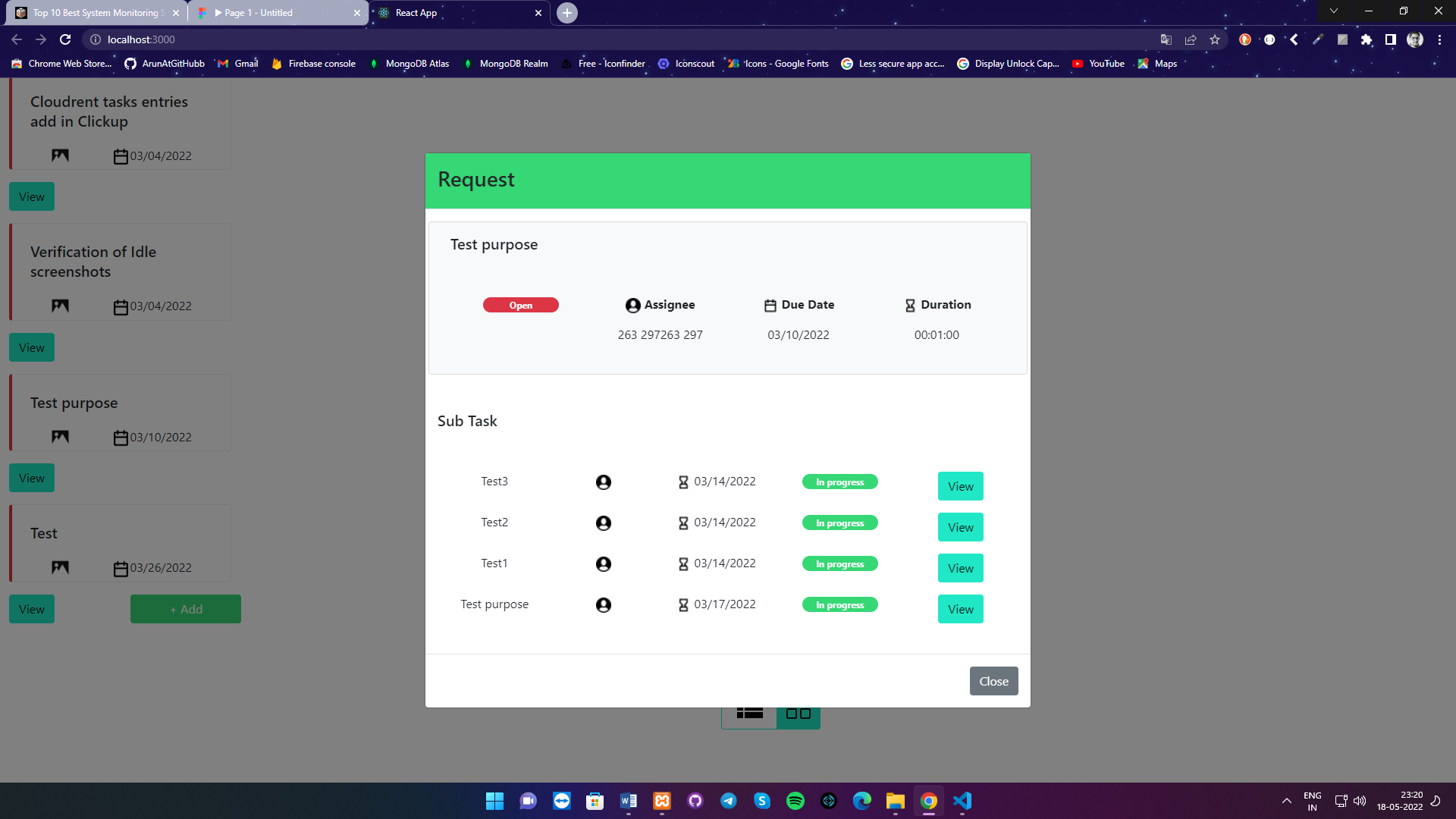
**Detailed view in request**

**(List View)**



**Detailed View in Request**

**(Kanban View)**



**CONCLUSION**

**CONCLUSION:-**

**This application was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.**

* **Automation of the entire system improves the efficiency**
* **It provides a friendly graphical user interface which proves to be better when compared to the existing system**
* **Updating information become easier**
* **The system has adequate scope for modification in future if it is necessary.**

**REFERENCES**

**For project mockup**

[**https://www.figma.com/**](https://www.figma.com/)

**For doubts about during bugs**

[**https://stackoverflow.com/**](https://stackoverflow.com/)

**For code sharing**

[**https://github.com/**](https://github.com/)