Full Stack Development Course Project

"News App"

Master of Computer Application

Submitted by:

Alzaahid Nadaf 2GI22MC011

Under the guidance of

Prof. H.S.Mogre

Department of Master of Computer Applications

Karnataka Law Society's

GOGTE INSTITUTE OF TECHNOLOGY

Udyambag, Belgavi- 590008

1. INTRODUCTION

NEED OF THE PROJECT -

Embark on a journey of innovation with NewsApp, a dynamic web application crafted with the latest technologies to elevate your digital experience. Powered by Node.js and Express on the server side, and React on the client side, NewsApp seamlessly integrates a variety of cutting-edge tools to bring you a feature-rich and responsive platform.

Key Technologies:

- Node.js and Express: The backbone of our server, Node.js and Express provide a robust foundation, ensuring high-performance and scalability forNewsApp.
- Jsonwebtoken: Security is paramount. With Jsonwebtoken, we implement a secure and reliable authentication system, safeguarding your data and ensuring a trustworthy user experience.
- Body-parser: Simplifying data handling, Body-parser streamlines the processing of incoming request bodies, making interactions withNewsApp swift and efficient.
- CORS: Embracing seamless cross-origin resource sharing, CORS enablesNewsApp to interact with various platforms and services, enhancing the versatility of your digital experience.
- MySQL: Our choice for data storage, MySQL ensures a robust and scalable database solution. Your data is in safe hands, ready to be retrieved and processed with efficiency.
- React: On the client side,NewsApp leverages the power of React to deliver a dynamic and interactive user interface. Expect a smooth, engaging, and responsive experience.
- React Bootstrap: Merging the flexibility of React with the sleek designs of Bootstrap,NewsApp delivers an aesthetically pleasing and user-friendly interface that adapts seamlessly to various devices.

3.SOFTWARE REQUIREMENTS SPECIFICATIONS

System Specifications:

Processor Required: Intel® Core™ ie-5005U @2.00GHz.
RAM: Should be 2 Gigabytes or more.
System Type: Should be 32-bit/64-bit operating system, x32 or x64-based Processor.
OS Required: Windows 7/8/10/11.

4.IMPLEMENTATION

Source Code

Backend(Node.js)

(Server.js)

```
var mysql = require('mysql');

var jwt = require('jsonwebtoken');

var express = require('express')

var bodyParser = require('body-parser');

var cors = require('cors')


var app = express()

app.use(cors())


app.use(bodyParser.json());
```

```
var con = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "",
    database: 'WDF_CA',
});

con.connect(function (err) {
    if (err) {
        console.log(err);
    }
    else {
        console.log("Connected!");
    };
});



/* getting all records from category table */
app.get('/category', (req, res) => {
    let id = req.query.id;
    let sql = 'select * from category';
    con.query(sql, (err, data) => {
        if (err) {
            res.send('Error !');
        } else {
            res.send(data);
        }
    });

});



/* Adding record to category table */
app.post('/category', (req, res) => {
    let token = req.body.token;
    let name = req.body.name;
    let sql = 'insert into category values(null,"' + name + '") ';
    if (token) {
        try {

            con.query(sql, (err, data) => {
                if (err) {
                    res.send({
                        'msg': 'flase',
                        'err': err
                    });
                } else {
                    res.send({
                        'msg': 'true',
                    });
                }
            });
        } catch (err) {
            res.send('invalid token');
        }
    } else {
        res.send("token not found")
    }

});

/* updating record from category table */
app.put('/category', (req, res) => {
    let token = req.body.token;
    let id = req.body.id;
    let name = req.body.name;
    let sql = 'update category set name = "' + name + '"  where id = ' + id + ' ';
    console.log(sql);

    if (token) {
        try {

            con.query(sql, (err, data) => {
                if (err) {
                    res.send({
                        'msg': 'flase',
                        'err': err
                    });
                } else {
                    res.send({
                        'msg': 'true',
                    });
                }
            });
        } catch (err) {
            res.send('invalid token');
        }
    } else {
        res.send("token not found")
    }

});

/* deleteing record from catogary table*/
app.delete('/category', (req, res) => {
    let id = req.body.id;
    let token = req.body.token;
    let sql = 'delete from category where id = ' + id + ' ';
    if (token) {
        try {

            con.query(sql, (err, data) => {
                if (err) {
                    res.send({
                        'msg': 'flase',
                        'err': err
                    });
                } else {
                    res.send({
                        'msg': 'true',
                    });
                }
            });
        } catch (err) {
            res.send('invalid token');
        }
    } else {
        res.send("token not found")
    }

});



/* getting all records from news-stories table */
app.get('/news-stories', (req, res) => {
    let id = req.query.id;
    let sql = 'select * from news_stories';
    con.query(sql, (err, data) => {
        if (err) {
            res.send('Error !');
        } else {
            res.send(data);
        }
    });

});

/* Adding record to Story table */
app.post('/news-stories', (req, res) => {
    let token = req.body.token;
    let title = req.body.title;
    let description = req.body.description;
    let category_id = req.body.category_id;
    let date = req.body.date;
    let sql = 'insert into news_stories VALUES(null,"' + title + '","' + description + 
'","' + date + '",' + category_id + ')';
    console.log(sql)
    if (token) {
        try {

            con.query(sql, (err, data) => {
                if (err) {
                    res.send({
                        'msg': 'flase',
                        'err': err
                    });
                } else {
                    res.send({
                        'msg': 'true',
                    });
                }
            });
        } catch (err) {
            res.send('invalid token');
        }
    } else {
        res.send("token not found")
    }

});

/* updating record from news-stories table */
app.put('/news-stories', (req, res) => {
    let id = req.body.id;
    let title = req.body.title;
    let token = req.body.token;
    let description = req.body.description; ``
    let sql = 'update news_stories set title = "' + title + '",description = "' + 
description + '"  where id = ' + id + ' ';
    if (token) {
        try {

            con.query(sql, (err, data) => {
                if (err) {
                    res.send({
                        'msg': 'flase',
                        'err': err
                    });
                } else {
                    res.send({
                        'msg': 'true',
                    });
                }
            });
        } catch (err) {
            res.send('invalid token');
        }
    } else {
        res.send("token not found")
    }

});



/* deleteing news stories */
app.delete('/news-stories', (req, res) => {
    let token = req.body.token;
    let id = req.body.id;
    let sql = 'delete from news_stories where id = ' + id + ' ';
    if (token) {
        try {

            con.query(sql, (err, data) => {
                if (err) {
                    res.send({
                        'msg': 'flase',
                        'err': err
                    });
                } else {
                    res.send({
                        'msg': 'true',
                    });
                }
            });
        } catch (err) {
            res.send('invalid token');
        }
    } else {
        res.send("token not found")
    }

});

/* login */
app.post('/login', (req, res) => {
    let email = req.body.email;
    }

    let password = req.body.password;



    let sql = "select * from user where email = '" + email + "' and password = '" + password + "'";
    console.log(sql);
    con.query(sql, (err, data) => {
        if (err) {
            res.send("error")
        } else {
            if (data.length > 0) {
                let token = jwt.sign({ msg: 'admin' }, 'abcd1234');
                res.send({
                    'msg': 'true',
                    'token': token
                });
            }
            else {
                res.send({
                    'msg': 'false'
                });
            }

        }
    });
```

```
})




/* Getting All News */

app.get('/all-news', (req, res) => {

   let sql = 'SELECT * from news_stories,category where
news_stories.category_id = category.id';

   con.query(sql, (err, data) => {

      if (err) {

         res.send('Error !');

      } else {

         res.send(data);

      }

   });


});


/* Getting news with filter  */

app.post('/news-filter', (req, res) => {

   let filters = req.body.filters

   let date = req.body.date


   let sql_all = 'SELECT * from news_stories,category where
news_stories.category_id = category.id';
```

```
   let sql_date_and_filter = 'SELECT
news_stories.id,news_stories.title,news_stories.description,\

   news_stories.date,category.name from news_stories,category where
news_stories.category_id = category.id and category_id in ('+ filters.toString() +
') and date =  "'+ date + '"';


   let sql_filter = 'SELECT
news_stories.id,news_stories.title,news_stories.description,\

   news_stories.date,category.name from news_stories,category where
news_stories.category_id = category.id and category_id in ('+ filters.toString() +
')';


   let sql_date = 'SELECT
news_stories.id,news_stories.title,news_stories.description,\

   news_stories.date,category.name from news_stories,category where
news_stories.category_id = category.id and date = "'+ date + '"';




   if (filters.length > 0 && date) {

      con.query(sql_date_and_filter, (err, data) => {

         if (err) {

            res.send('Error !');

         } else {

            res.send(data)

            console.log(data)

         }

      });
```

```
}
else if (filters.length > 0) {
    con.query(sql_filter, (err, data) => {
        if (err) {
            res.send('Error !');
        } else {
            res.send(data)
            console.log(data)
        }
    });

}
else if (date) {
    con.query(sql_date, (err, data) => {
        if (err) {
            res.send('Error !');
        } else {
            res.send(data)
            console.log(data)
        }
    });

}
else{
    con.query(sql_all, (err, data) => {
        if (err) {
                res.send('Error !');
            } else {
                res.send(data)
                console.log(data)
            }
        });
    }

});



app.listen(8080, () => {
    console.log('server is up')
})
```

Frontend(React.js)

(App.js)

```
import React, { useState, useEffect } from "react";
import Nav from "./navbar";
import "./App.css";
import 'bootstrap/dist/css/bootstrap.css';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';



export default function App() {
  const [loggedIn, setLoggedIn] = useState(false)

  const [email, setEmail] = useState('')
  const [password, setPassword] = useState('')
  const [loginfail, setSetLoginFail] = useState('')
  const [emailValidation, setEmailValidation] = useState('')
  const [passwordValidation, setPasswordValidation] = useState('')
  const [token, setToken] = useState('')


  const login = () => {
    var validRegex = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;

    if (email.length == 0) {
      setEmailValidation(
        <p class="pull-right" style={{ marginLeft: 150, color: 'red', position: 'absolute' }}>
          Please enter a email.
        </p>
      )
      setTimeout(() => { setEmailValidation(''); }, 2000);

    }
```

```
else if (!email.match(validRegex)) {
  setEmailValidation(
    <p style={{ marginLeft: 100, color: 'red', position: 'absolute' }}>
      Please Enter a valid email
    </p>
  )
  setTimeout(() => { setEmailValidation(''); }, 2000);

}
else if (password.length == 0) {
  setPasswordValidation(
    <p style={{ marginLeft: 80, color: 'red', position: 'absolute' }}>
      Please enter your password.
    </p>
  )
  setTimeout(() => { setPasswordValidation(''); }, 2000);
}
else {
  fetch("http://localhost:8080/login", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ email: email, password: password })
  })
    .then(res => res.json())
    .then((res) => {
      if (res.msg == "true") {
        setToken(res.token)
        setLoggedIn(true)
```

```
     }
     else {
      setSetLoginFail(
       <p style={{ color: 'red' ,position: 'absolute',marginLeft : 20}}>
        Your Email Or Password Is Incorrect!
       </p>)
       setTimeout(() => { setSetLoginFail(''); }, 2000);
     }

    })
  }
 }

 if (loggedIn == true) {
  return (
   <div className="container-fluid">
    <div className="main-app" >
     <Nav token={token} />
    </div>
   </div>
  );

 }
 else {
  return (
   <div className="login" >
    <div className="login-box" style={{margin : 'auto', padding : '10px', width : '320px', backgroundColor : 'white', height : '370px', borderRadius : '5%'}}>
     <img src="img/download.png" style={{ marginLeft: 100 }} width="100px" height="30px" /><br />
```

```
        {loginfail}

        <label style={{ marginTop: 30 }}>Email</label>

        <input type="email" className="form-control" value={email} onChange={event =>
setEmail(event.target.value)}  />

        <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"
class="bi bi-envelope" viewBox="0 0 16 16" style={{marginTop : -65, marginLeft:280}}>

  <path d="M0 4a2 2 0 0 1 2-2h12a2 2 0 0 1 2 2v8a2 2 0 0 1-2 2H2a2 2 0 0 1-2-2V4Zm2-1a1 1
0 0 0-1 1v.217l7 4.2 7-4.2V4a1 1 0 0 0-1-1H2Zm13 2.383-4.708 2.825L15
11.105V5.383Zm-.034 6.876-5.64-3.471L8 9.583l-1.326-.795-5.64 3.47A1 1 0 0 0 2 13h12a1
1 0 0 0 .966-.741ZM1 11.105l4.708-2.897L1 5.383v5.722Z"/>

</svg>

<br />

        {emailValidation}


        <label>Password</label>

        <input type="password" className="form-control" value={password}
onChange={event => setPassword(event.target.value)}

        />

        <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"
class="bi bi-shield-lock" viewBox="0 0 16 16" style={{marginTop : -65, marginLeft:280}}>

  <path d="M5.338 1.59a61.44 61.44 0 0 0-2.837.856.481.481 0 0 0-.328.39c-.554 4.157.726
7.19 2.253 9.188a10.725 10.725 0 0 0 2.287
2.233c.346.244.652.42.893.533.12.057.218.095.293.118a.55.55 0 0 0 .101.025.615.615 0 0
0 .1-.025c.076-.023.174-.061.294-.118.24-.113.547-.29.893-.533a10.726 10.726 0 0 0
2.287-2.233c1.527-1.997 2.807-5.031 2.253-9.188a.48.48 0 0
0-.328-.39c-.651-.213-1.75-.56-2.837-.855C9.552 1.29 8.531 1.067 8 1.067c-.53
0-1.552.223-2.662.524zM5.072.56C6.157.265 7.31 0 8 0s1.843.265 2.928.56c1.11.3
2.229.655 2.887.87a1.54 1.54 0 0 1 1.044 1.262c.596 4.477-.787 7.795-2.465 9.99a11.775
11.775 0 0 1-2.517 2.453 7.159 7.159 0 0
1-1.048.625c-.28.132-.581.24-.829.24s-.548-.108-.829-.24a7.158 7.158 0 0 1-1.048-.625
11.777 11.777 0 0 1-2.517-2.453C1.928 10.487.545 7.169 1.141 2.692A1.54 1.54 0 0 1 2.185
1.43 62.456 62.456 0 0 1 5.072.56z"/>

  <path d="M9.5 6.5a1.5 1.5 0 0 1-1 1.415l.385 1.99a.5.5 0 0 1-.491.595h-.788a.5.5 0 0
1-.49-.595l.384-1.99a1.5 1.5 0 1 1 2-1.415z"/>

</svg>

        <br />
        {passwordValidation}

        <button style={{ marginLeft: 90, marginTop: 30, width:'40%' }} onClick={login}
className="btn btn-success">Login</button>

        <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"
class="bi bi-joystick" viewBox="0 0 16 16" style={{marginTop : -65, marginLeft:180,
color:'whitesmoke'}}>

  <path d="M10 2a2 2 0 0 1-1.5 1.937v5.087c.863.083 1.5.377 1.5.726 0 .414-.895.75-2
.75s-2-.336-2-.75c0-.35.637-.643 1.5-.726V3.937A2 2 0 1 1 10 2z"/>

  <path d="M0 9.665v1.717a1 1 0 0 0 .553.894l6.553 3.277a2 2 0 0 0 1.788 0l6.553-3.277a1
1 0 0 0 .553-.894V9.665c0-.1-.06-.19-.152-.23L9.5 6.715v.993l5.227 2.178a.125.125 0 0 1
.001.23l-5.94 2.546a2 2 0 0 1-1.576 0l-5.94-2.546a.125.125 0 0 1 .001-.23L6.5
7.708l-.013-.988L.152 9.435a.25.25 0 0 0-.152.23z"/>

</svg>


      </div>

     </div>


   );

  }



}
```

(Category.js)

```
import React, { useState, useEffect } from "react";

import { BsPencilSquare } from "react-icons/bs";
import { BsTrashFill } from "react-icons/bs";
import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom";

export default function Chatorey({ getData ,category,token }) {



 /* Adding record state variables */
 const [name,setName] = useState('')
 const [addValidation,setAddValidation] = useState('')
 const [addMess,setAddMess] = useState('')

 /* edit state variables */
 const [editId,setEditId] = useState('')
 const [editName,setEditName] = useState('')
 const [editValidation,setEditValidation] = useState('')
 const [editMess,setEditMess] = useState('')




 const addCat = () =>{
  if(name.length == 0){
   setAddValidation(
   <div class="alert alert-danger" role="alert">
    Name Cannot Be Eemty!
   </div>)
   setTimeout(() => {  setAddValidation(''); }, 2000);
  }
  else{
   setAddValidation('')


   fetch("/category", {
   method: "POST",
   headers: { "Content-Type": "application/json" },
   body: JSON.stringify({ name: name ,token: token})
  })
   .then(res => res.json())
   .then((res) => {
    console.log(res)
    if (res.msg == "true") {
     getData()
     setAddMess(
      <div class="alert alert-success" role="alert">
       New Category Created!
      </div>
     )
     setName('')
     setTimeout(() => {  setAddMess(''); }, 2000);
    }
   })
  }
 }



 const editCat = (c) => {
  setEditName(c.name)
  setEditId(c.id)
```

```
};

const updateCat = () =>{
 if(editName.length == 0){
  setEditValidation(
  <div class="alert alert-danger" role="alert">
   Name Cannot Be Eemty!
  </div>)
  setTimeout(() => {  setEditValidation(''); }, 2000);
 }
 else{
  setEditValidation('')

  fetch("/category", {
  method: "PUT",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({ name: editName, id: editId,token: token })
 })
  .then(res => res.json())
  .then((res) => {
   if (res.msg == "true") {
    getData()
    setEditMess(
     <div class="alert alert-success" role="alert">
      Updated!
     </div>
    )
    setTimeout(() => {  setEditMess(''); }, 2000);
   }
  })
 }
}

const deleteCat = () =>{
  fetch("/category", {
  method: "Delete",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({ id: editId,token: token })
 })
  .then(res => res.json())
  .then((res) => {
   if (res.msg == "true") {
    getData()
   }
  })
}


return (
 <div>
  <button className="catAddBtn btn btn-success" data-toggle="modal" data-target="#addModelBox" >
   Add News Categories
  </button>



  {/* add catogary model pop up box  */}
  <div className="modal fade" id="addModelBox" tabIndex={-1} role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
```

```
    <div className="modal-dialog modal-dialog-centered" role="document">
      <div className="modal-content">
       <div className="modal-header">
        <h5 className="modal-title" id="exampleModalLongTitle">Add News
Categories</h5>
        <button type="button" className="close" data-dismiss="modal"
aria-label="Close">
         <span aria-hidden="true">×</span>
        </button>
       </div>
       <div className="modal-body">
        {addValidation}
        {addMess}
        <input
        type="text"
        onChange={event => setName(event.target.value)}
        placeholder="Enter News Category"
        name="name"
        value={name}
        className="form-control"
       />
       </div>
       <div className="modal-footer">
        <button type="button" className="btn btn-secondary"
data-dismiss="modal">Close</button>
        <button type="button" className="btn btn-primary"
onClick={addCat}>Add</button>
       </div>
      </div>
    </div>
   </div>
```

```
      {/* table to display all the catogaries */}
      <table className="table table-striped ">
        <thead>
          <tr>
            <th scope="col">Name</th>
            <th scope="col">Action</th>
          </tr>
        </thead>
        <tbody>
          {
            category.map(c => {
              return(
                <tr>
            <td>{c.name}</td>
              <td>
                <button className="btn btn-danger" onClick={() => editCat(c)} style={{ marginRight
: 20,marginLeft : 20}} data-toggle="modal"
data-target="#deleteModelBox"><BsTrashFill/></button>
                <button className="btn btn-primary" onClick={() => editCat(c)}
data-toggle="modal" data-target="#editModelBox"> <BsPencilSquare/></button>
              </td>
            </tr>
              )
            })
          }
        </tbody>
      </table>


      {/* modal pop up for updating record */}
```

```
<div className="modal fade" id="editModelBox" tabIndex={-1} role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
  <div className="modal-dialog modal-dialog-centered" role="document">
    <div className="modal-content">
      <div className="modal-header">
        <h5 className="modal-title" id="exampleModalLongTitle">Edit</h5>
        <button type="button" className="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">×</span>
        </button>
      </div>
      <div className="modal-body">
        {editValidation}
        {editMess}
        <input type="text" className="form-control" value={editName} onChange={event => setEditName(event.target.value)}/>
      </div>
      <div className="modal-footer">
        <button type="button" className="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" className="btn btn-primary" onClick={updateCat}>Save changes</button>
      </div>
    </div>
  </div>
</div>



<div className="modal" id="deleteModelBox" tabIndex={-1} role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
  <div className="modal-dialog modal-dialog-centered" role="document">
```

```
     <div className="modal-content">
      <div className="modal-header">
       <h5 className="modal-title" id="exampleModalLongTitle">Are You Sure ? You want to delete "{editName}"</h5>
       <button type="button" className="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">×</span>
       </button>
      </div>
      <div className="modal-footer">
       <button type="button" className="btn btn-secondary" onClick={deleteCat} data-dismiss="modal">Yes</button>
       <button type="button" className="btn btn-secondary" data-dismiss="modal">No</button>


      </div>
     </div>
    </div>
   </div>



  </div>
 );
}
```

(Home.js)

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

```
export default function Home({ news, filters, removeFilter, setTdate, tdate, handleDateChange }) {




  return (
    <div>
      <div className="d-flex justify-content-center">
        <div class="btn-group">

          <div class="dropdown-menu" style={{ backgroundColor: 'white', marginRight: '80px' }}>
            <input type="date" className="form-control" value={tdate} onChange={event => handleDateChange(event.target.value)} />
            <div class="dropdown-divider"></div>
            <button
              style=
              {{
                textDecoration: 'none',
                backgroundColor: 'whitesmoke',
                border: 'none',
                width : '100%'

              }}
              onClick={() => setTdate('')}
              >
              Reset Date
            </button>
          </div>
        </div>
      </div>


      {
        filters.map(c => {
          return (
            <span style={{ marginRight: '5px' }} class="badge badge-secondary">{c.name} <button style={{ border: 'none', textDecoration: "none", backgroundColor: '#c3c3c3', borderRadius: '50px' }} onClick={() => removeFilter(c.id)}>&times;</button></span>
          )
        })
      }


      {
        news.map(c => {
          return (
            <div style={{
              padding: '15px', border: '1px solid grey' , margin: '10px'
            }}
            >




                <img
                src="img/sports.jpg"
               width="440px"
                className="d-inline-block align-top"
                alt="React Bootstrap logo"
              />
```

```
 <br />
 <hr style={{

  borderRadius: '5px',
  width: '100%'
}} />
 <b>
 <p style={{ color: '#444444' }}>Category : {c.name}
 </p>
 </b>
 <hr style={{

   borderRadius: '5px',
   width: '100%'
 }} />
 <h4>Title :{c.title}</h4>
 <hr style={{

  borderRadius: '5px',
  width: '100%'
}} />
 <p>{c.description}</p>
 <hr style={{

  borderRadius: '5px',
  width: '100%'
}} />
  <p>  <img  style={{width: 30, height: 30, borderRadius: 400/ 2}}
```

```
    src="img/admin.png"
    alt="new"
    />Posted By Admin | {
          new Date(
            c.date
          ).toLocaleDateString("en-US", {
            weekday: "short",
            year: "numeric",
            month: "short",
            day: "2-digit",
          })
        } </p>


       </div>



      )
     })
    }


   </div>
  )
}
```

(Nav.js)

```
import React, { Component, useState, useEffect } from "react";
import { FaBars } from "react-icons/fa";
import { BsFillPersonFill } from "react-icons/bs";
import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom";
import Chatorey from './Category.js'
import AddStories from './Stories.js'
import Home from './Home'



export default function Nav({ token }) {
 const [width, setWidth] = useState(0);

 const openMenu = () => {
  setWidth(250);
 };
 const close = () => {
  setWidth(0);
 };

 const test = () => {
  console.log("test");
 };

 /* contains all cat records from database */
 const [category, setCategory] = useState([])

 const getData = () => {
  fetch("/category")
```

```
    .then(res => res.json())
    .then(res => {
      console.log(res);
      setCategory(res)
    })
};

const [news, setNews] = useState([])
const getNews = () => {
  fetch("/all-news")
    .then(res => res.json())
    .then(res => {
      console.log(res);
      setNews(res)
    })
};



var today = new Date();
var dd = String(today.getDate()).padStart(2, '0');
var mm = String(today.getMonth() + 1).padStart(2, '0'); //January is 0!
var yyyy = today.getFullYear();

const [tdate, setTdate] = useState()

const handleDateChange = (date) => {
  setTdate(date)
}
```

```
const [filters, setFilters] = useState([])
const [filterId, setFilterId] = useState([])

const filter = (id, name) => {
  let new_list = [...filters, { id: id, name: name }]
  setFilters(new_list)

  let new_filter_id = [...filterId, id]
  setFilterId(new_filter_id)


}

const removeFilter = (id) => {
  let newList = filters.filter((ele) => {
    if (ele.id != id) {
      return (ele)
    }
  })
  setFilters(newList)

  let newListIds = filterId.filter((ele) => {
    if (ele != id) {
      return (ele)
    }
  })
  setFilterId(newListIds)
}
```

```
  const getFilterNews = () => {
    setNews([])
    fetch("/news-filter", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ filters: filterId ,date:tdate })
    })
      .then(res => res.json())
      .then(res => {
        console.log(res);
        setNews(res)

      })
  }

  const handleAll = () => {
    setTdate('')
    setFilters([])
}




  useEffect(() => {
    getFilterNews()
  }, [filterId,tdate]);

  useEffect(() => {
    getData();
    getNews();
  }, []);

  return (
    <div>
      <nav className="navbar navbar-dark bg-dark">
        <a className="navbar-brand" href="#">
          <img src="img/download.png" width="80px" />{" "}
        </a>
        <button className="barsIcon" onClick={openMenu}>
          <FaBars />
        </button>
      </nav>
      <Router>
        <div id="mySidenav" class="sidenav" style={{ width: width }}>
          <div className="dropdown" >
            <button
              className="btn btn-secondary dropdown-toggle"
              type="button"
              id="dropdownMenuButton"
              data-toggle="dropdown"
              aria-haspopup="true"
              aria-expanded="false"
            >
              <BsFillPersonFill />  Hi Admin
            </button>
            <div className="dropdown-menu" aria-labelledby="dropdownMenuButton">
              <Link to="/add-chatorey" onClick={close}>Manage Category</Link>
              <Link to="/add-stories" onClick={close}>Manage Stories</Link>
            </div>
    </div>

    <button className="closebtn" onClick={close}>
      &times;
    </button>

    <Link to="/" onClick={(test, close)}>
      Home
    </Link>
    <button className="filter-btn"
      onClick={() => handleAll()}
    >
      All
    </button>
    {
      category.map(c => {
        return (
          <button className="filter-btn" onClick={() => filter(c.id, c.name)}>
            {c.name}
          </button>
        )
      })
    }


  </div>

  <Switch>
    <Route path="/add-chatorey">
```

```
          <Chatorey category={category} getData={getData} token={token} />
        </Route>
        <Route path="/add-stories">
          <AddStories token={token} />
        </Route>

        <Route path="/">
          <Home news={news} filters={filters} removeFilter={removeFilter} tdate={tdate}
setTdate={setTdate} handleDateChange={handleDateChange} />

        </Route>

      </Switch>
    </Router>
  </div>
 );
}
```

(Navbar.js)

```
import React from 'react';
import 'bootstrap/dist/css/bootstrap.css';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import {Container ,Card, Col, Navbar,Nav} from 'react-bootstrap';
import { Component, useState, useEffect } from "react";
import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom";
import Chatorey from './Category.js'

import AddStories from './Stories.js'

import Home from './Home'



export default function Navigation({token}) {

  {

    const [width, setWidth] = useState(0);


    const openMenu = () => {

      setWidth(250);

    };

    const close = () => {

      setWidth(0);

    };


    const test = () => {

      console.log("test");

    };


    /* contains all cat records from database */

    const [category, setCategory] = useState([])


    const getData = () => {

      fetch("/category")

        .then(res => res.json())

        .then(res => {

          console.log(res);

          setCategory(res)
        })
    };

    const [news, setNews] = useState([])
    const getNews = () => {
      fetch("/all-news")
        .then(res => res.json())
        .then(res => {
          console.log(res);
          setNews(res)
        })
    };



    var today = new Date();
    var dd = String(today.getDate()).padStart(2, '0');
    var mm = String(today.getMonth() + 1).padStart(2, '0'); //January is 0!
    var yyyy = today.getFullYear();

    const [tdate, setTdate] = useState()

    const handleDateChange = (date) => {
      setTdate(date)
    }

    const [filters, setFilters] = useState([])
    const [filterId, setFilterId] = useState([])

    const filter = (id, name) => {
      let new_list = [...filters, { id: id, name: name }]
      setFilters(new_list)

      let new_filter_id = [...filterId, id]
      setFilterId(new_filter_id)



    }

    const removeFilter = (id) => {
      let newList = filters.filter((ele) => {
        if (ele.id != id) {
          return (ele)
        }
      })
      setFilters(newList)

      let newListIds = filterId.filter((ele) => {
        if (ele != id) {
          return (ele)
        }
      })
      setFilterId(newListIds)
    }

    const getFilterNews = () => {
      setNews([])
      fetch("/news-filter", {
        method: "POST",
       headers: { "Content-Type": "application/json" },
       body: JSON.stringify({ filters: filterId ,date:tdate })
      })
       .then(res => res.json())
       .then(res => {
        console.log(res);
        setNews(res)

       })
     }

     const handleAll = () => {
      setTdate('')
      setFilters([])
    }



     useEffect(() => {
      getFilterNews()
     }, [filterId,tdate]);

     useEffect(() => {
      getData();
      getNews();
     }, []);
     return (
      <div>
        <Router>
     <Navbar bg="primary" variant="dark">
      <Container>
      <Navbar.Brand href="#home"><img
       src="img/download.png"

       className="d-inline-block align-top"
       alt="React Bootstrap logo"
      /></Navbar.Brand>
      <Nav className="me-auto">
      <Link to="/" onClick={(test, close)} style={{color:'whitesmoke', textDecoration:'none',
   marginTop:7}}>
         Home
        </Link>
     <Nav.Link>

     <Link to="/add-chatorey" onClick={close} style={{color:'whitesmoke' ,
   textDecoration:'none'}}>Add News Categories</Link>

     </Nav.Link>

          <Link to="/add-stories" onClick={close} style={{color:'whitesmoke' ,
   textDecoration:'none', marginTop:8}}>Add News</Link>
      </Nav>




      </Container>
     </Navbar>

     <Switch>
```

```
        <Route path="/add-chatorey">
          <Chatorey category={category} getData={getData} token={token} />
        </Route>
        <Route path="/add-stories">
          <AddStories token={token} />
        </Route>

        <Route path="/">
          <Home news={news} filters={filters} removeFilter={removeFilter} tdate={tdate} setTdate={setTdate} handleDateChange={handleDateChange} />

        </Route>

      </Switch>
    </Router>

    </div>
    )
  }
}
```

(Stories.js)

```
import React, { useState, useEffect } from "react";
import { BsPencilSquare } from "react-icons/bs";
import { BsTrashFill } from "react-icons/bs";
```

```
export default function Stories({token}) {

  /* contains all cat records from database */
  const [stories, setStories] = useState([])
  const [categories, setCategories] = useState([])

  const getData = () => {
    fetch("/news-stories")
      .then(res => res.json())
      .then(res => {
        console.log(res);
        setStories(res)
      })
  };

  const getCat = () => {
    fetch("/category")
      .then(res => res.json())
      .then(res => {
        console.log(res);
        setCategories(res)
      })
  };

  useEffect(() => {
    getData();
    getCat();
  }, []);
```

```
useEffect(() => {
  getData();
}, []);


/* Adding record state variables */
const [title, setTitle] = useState('')
const [description, setDescription] = useState('')
const [categoryId, setCategoryId] = useState('')
const [date, setDate] = useState('')
const [addValidation, setAddValidation] = useState('')
const [addMess, setAddMess] = useState('')



const addStory = () => {
  if (title.length == 0) {
    setAddValidation(
      <div class="alert alert-danger" role="alert">
        Title Cannot Be Emty!
    </div>)
    setTimeout(() => { setAddValidation(''); }, 2000);
  }
  else if (description.length == 0) {
    setAddValidation(
      <div class="alert alert-danger" role="alert">
        Description Cannot Be Emty!
      </div>)
```

```
   setTimeout(() => { setAddValidation(''); }, 2000);
  }
  else if (date.length == 0) {
   setAddValidation(
    <div class="alert alert-danger" role="alert">
     Date Cannot Be Emty!
    </div>)
   setTimeout(() => { setAddValidation(''); }, 2000);
  }
  else if (categoryId.length == 0 || categoryId == 0) {
   setAddValidation(
    <div class="alert alert-danger" role="alert">
     Please select a Category !
    </div>)
   setTimeout(() => { setAddValidation(''); }, 2000);
  }
  else {
   setAddValidation('')

    fetch("/news-stories", {
     method: "POST",
     headers: { "Content-Type": "application/json" },
     body: JSON.stringify({ title: title, description: description, date: date, category_id:
categoryId, token: token })
    })
     .then(res => res.json())
     .then((res) => {
      console.log(res)
      if (res.msg == "true") {
       getData()
      setAddMess(
       <div class="alert alert-success" role="alert">
        Inserted!
      </div>
      )
      setTimeout(() => { setAddMess(''); }, 2000);
      setTitle('')
      setDescription('')
      setDate('')
      setCategoryId('')
     }
    })
 }
}

/* edit state variables */
const [editId, setEditId] = useState('')
const [editTitle, setEditTitle] = useState('')
const [editDescription, setEditDescription] = useState('')
const [editCatId, setEditCatId] = useState('')
const [editDate, setEditDate] = useState('')
const [editValidation, setEditValidation] = useState('')
const [editMess, setEditMess] = useState('')

const editStory = (s) => {
 console.log(s.id)
 setEditId(s.id)
 setEditTitle(s.title)
 setEditDescription(s.description)
```

```
  setEditCatId(s.category_id)
  setEditDate(s.date)
 };

 const updateStory = () => {
  if (editTitle.length == 0) {
   setEditValidation(
    <div class="alert alert-danger" role="alert">
     Title Cannot Be Eemty!
   </div>)
   setTimeout(() => { setEditValidation(''); }, 2000);
  }
  else if (editDescription.length == 0) {
   setEditValidation(
    <div class="alert alert-danger" role="alert">
     Description Cannot Be Eemty!
    </div>)
   setTimeout(() => { setEditValidation(''); }, 2000);

  }
  else {
   setEditValidation('')

   fetch("/news-stories", {
    method: "PUT",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ title: editTitle, description: editDescription, id: editId,token: token
})
   })
    .then(res => res.json())
```

```
      .then((res) => {
        console.log(res)
        if (res.msg == "true") {
          getData()
          setEditMess(
            <div class="alert alert-success" role="alert">
              Updated!
          </div>
          )
          setTimeout(() => { setEditMess(''); }, 2000);
        }
      })
  }
}

const deleteStory = () => {
  fetch("/news-stories", {
    method: "Delete",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ id: editId ,token: token})
  })
    .then(res => res.json())
    .then((res) => {
      if (res.msg == "true") {
        getData()
      }
    })
}
```

```
 return (
  <div>
   <button className="catAddBtn btn btn-success" data-toggle="modal"
data-target="#addStoryBox" style={{width:'30%'}}>
    Add Story
   </button>
   <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"
class="bi bi-folder-plus" viewBox="0 0 16 16" style={{marginTop : -82,color:'white',
marginLeft:430}}>
 <path d="m.5 3 .04.87a1.99 1.99 0 0 0-.342 1.311l.637 7A2 2 0 0 0 2.826 14H9v-1H2.826a1
1 0 0 1-.995-.91l-.637-7A1 1 0 0 1 2.19 4h11.62a1 1 0 0 1 .996 1.09L14.54
8h1.005l.256-2.819A2 2 0 0 0 13.81 3H9.828a2 2 0 0 1-1.414-.586l-.828-.828A2 2 0 0 0 6.172
1H2.5a2 2 0 0 0-2 2zm5.672-1a1 1 0 0 1 .707.293L7.586 3H2.19c-.24 0-.47.042-.683.12L1.5
2.98a1 1 0 0 1 1-.98h3.672z"/>
 <path d="M13.5 10a.5.5 0 0 1 .5.5V12h1.5a.5.5 0 1 1 0 1H14v1.5a.5.5 0 1 1-1
0V13h-1.5a.5.5 0 0 1 0-1H13v-1.5a.5.5 0 0 1 .5-.5z"/>
</svg>

   {/*  Adding story modal box  */}
   <div className="modal fade" id="addStoryBox" tabIndex={-1} role="dialog"
aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div className="modal-dialog modal-dialog-centered" role="document">
     <div className="modal-content">
      <div className="modal-header">
       <h5 className="modal-title" id="exampleModalLongTitle">Add a Story</h5>
       <button type="button" className="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">×</span>
       </button>
      </div>
      <div className="modal-body">
       <div className="container-fluid">
    <div className="row">
      <div className="col-md-12">
        {addValidation}
        {addMess}
        <label >Title</label>
        <textarea id="w3review" value={title} onChange={event => setTitle(event.target.value)} className="form-control" ></textarea><br />
        <label >Description</label>
        <textarea id="w3review" value={description} onChange={event => setDescription(event.target.value)} className="form-control"></textarea><br />
        <label>Date</label>
        <input type="date" className="form-control" value={date} onChange={event => setDate(event.target.value)} /><br />
        <select onChange={event => setCategoryId(event.target.value)} className="form-control"><br />
          <option value="0">Select Category</option>
          {
            categories.map(c => {
              return (
                <option value={c.id}>{c.name}</option>
              )
            })
          }

        </select>

      </div>

    </div>
  </div>
</div>
```

```
          <div className="modal-footer">
            <button type="button" className="btn btn-secondary"
data-dismiss="modal">Close</button>
            <button type="button" className="btn btn-primary"
onClick={addStory}>Add</button>
          </div>
        </div>
      </div>
    </div>




    {/* table to display all the catogaries */}
    <table className="table table-hover">
      <thead>
        <tr>
          <th scope="col">Name</th>
          <th scope="col" crSpan='2'>Action</th>
        </tr>
      </thead>
      <tbody>
        {
          stories.map(s => {
            return (
              <tr>
                <td>{s.title.substr(0, 20)}...</td>
                <td>
          <button className="btn btn-danger" data-toggle="modal" onClick={() => editStory(s)} data-target="#deleteModelBox"><BsTrashFill /></button>

          <button className="btn btn-primary" onClick={() => editStory(s)} style={{ marginLeft: 10 }} data-toggle="modal" data-target="#editStoryModal"> <BsPencilSquare /></button>

         </td>

        </tr>

       )

      })

     }

    </tbody>

   </table>


   {/*  edit story modal box  */}

   <div className="modal fade" id="editStoryModal" tabIndex={-1} role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">

    <div className="modal-dialog modal-dialog-centered" role="document">

     <div className="modal-content">

      <div className="modal-header">

       <h5 className="modal-title" id="exampleModalLongTitle">Edit Story</h5>

       <button type="button" className="close" data-dismiss="modal" aria-label="Close">

        <span aria-hidden="true">×</span>

       </button>

      </div>

      <div className="modal-body">

       <div className="container-fluid">

        <div className="row">

         <div className="col-md-12">

          {editMess}

          {editValidation}

          <label >Title</label>
```

```
                    <textarea id="w3review" value={editTitle} onChange={event => setEditTitle(event.target.value)} className="form-control" ></textarea><br />


                    <label >Description</label>
                    <textarea id="w3review" value={editDescription} onChange={event => setEditDescription(event.target.value)} className="form-control" rows="6" cols="50"></textarea>
                  </div>
                </div>
              </div>
            </div>
            <div className="modal-footer">
              <button type="button" className="btn btn-secondary" data-dismiss="modal">Close</button>
              <button type="button" className="btn btn-primary" onClick={updateStory}>Save changes</button>
            </div>
          </div>
        </div>
      </div>


      <div className="modal" id="deleteModelBox" tabIndex={-1} role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
        <div className="modal-dialog modal-dialog-centered" role="document">
          <div className="modal-content">
            <div className="modal-header">
              <h5 className="modal-title" id="exampleModalLongTitle">Are You Sure ? You want to delete "{editTitle}"</h5>
              <button type="button" className="close" data-dismiss="modal" aria-label="Close">
                <span aria-hidden="true">×</span>
              </button>
            </div>
            <div className="modal-footer">
              <button type="button" className="btn btn-secondary" onClick={deleteStory}
  data-dismiss="modal">Yes</button>
              <button type="button" className="btn btn-secondary"
  data-dismiss="modal">No</button>

            </div>
          </div>
        </div>
      </div>


    </div>
   );
  }
```

(Index.js)

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
 <React.StrictMode>
   <App />
```

```
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

(App.css)

```
.container-fluid{
  padding: 0 !important;
}
.barsIcon{
  border: 0;
  background-color: #343a40;
  color: white;
  font-size: 30px;
}

.sidenav {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  right: 0;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.2s;
  padding-top: 60px;
}

.sidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 30px;
  color: #818181;
  display: block;
  transition: 0.3s;
}

.filter-btn{
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 30px;
  color: #818181;
  display: block;
  transition: 0.3s;
  background-color: #111111;
  border:none
}

.sidenav a:hover {
  color: #f1f1f1;
```

```
}

.closebtn {
  border: none;
  background-color: #111111;
  color: white;
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}

@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}

#dropdownMenuButton{
  position: absolute;
  bottom: 10px;
  border: none;
  left: 30px;
  background-color: #111111;
  font-size: 20px;
  color : whitesmoke;

}
.dropdown-menu{
  border: none;
  background-color: #343a40;

}

.dropdown-menu a{
  font-size: 15px;
  text-decoration: none;
  color: whitesmoke;
}

.catAddBtn{
  margin-top: 20px;
  margin-left: 63%;
  margin-bottom: 10px;
}
.login{
  background: #a29f9f;
  padding-top: 160px;
  padding-bottom: 240px;

}
.login-box{
  margin: auto;
  padding: 10px;
  width: 320px;
  background-color: whitesmoke;
  height: 370px;
  border-radius: 5%;
```

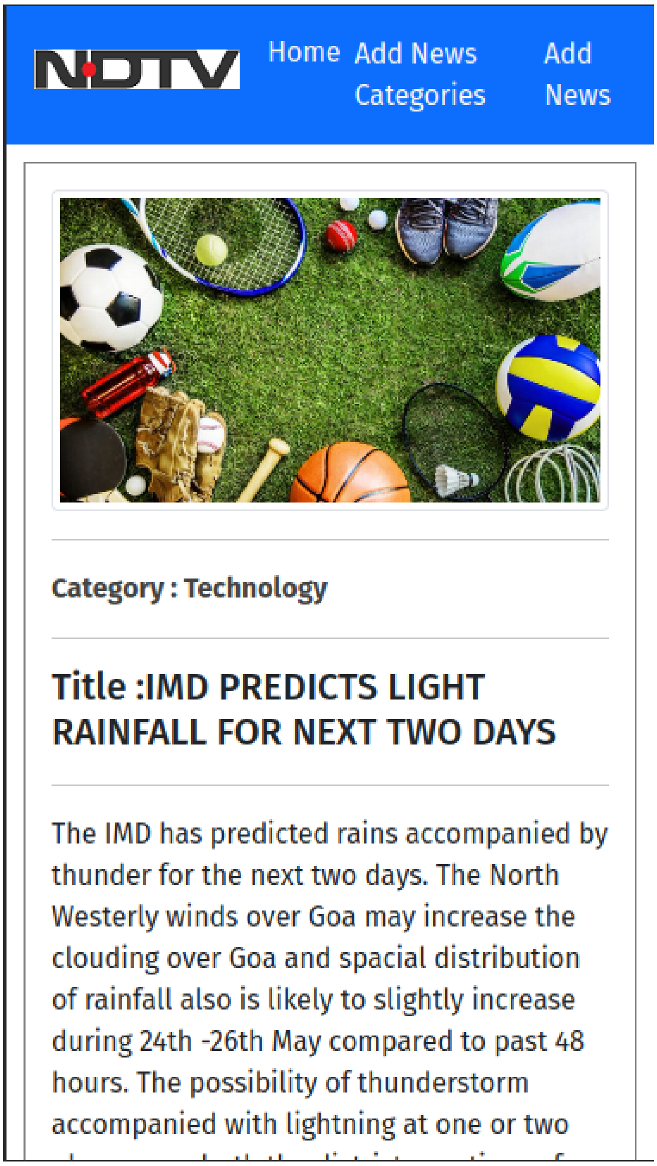
```
}
```

(Index.css)

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

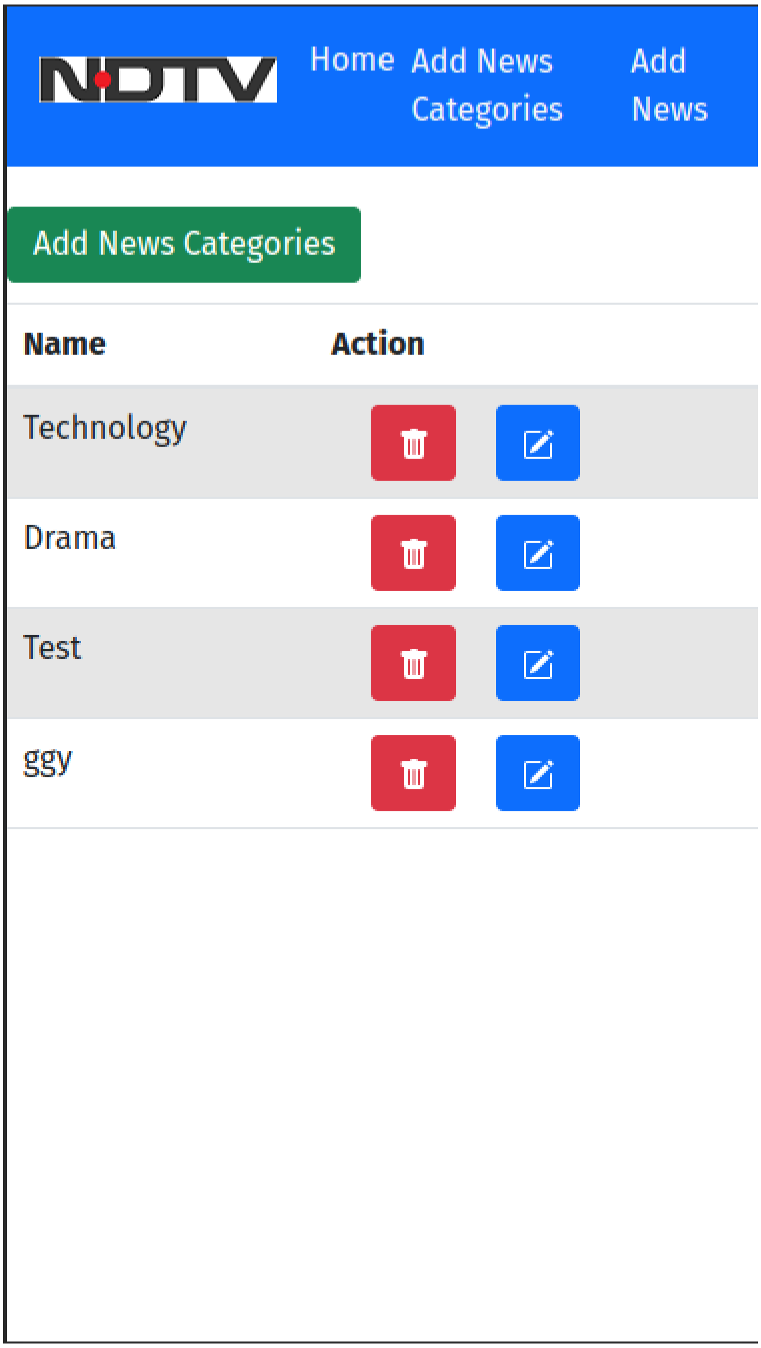
5.SCREENSHOTS



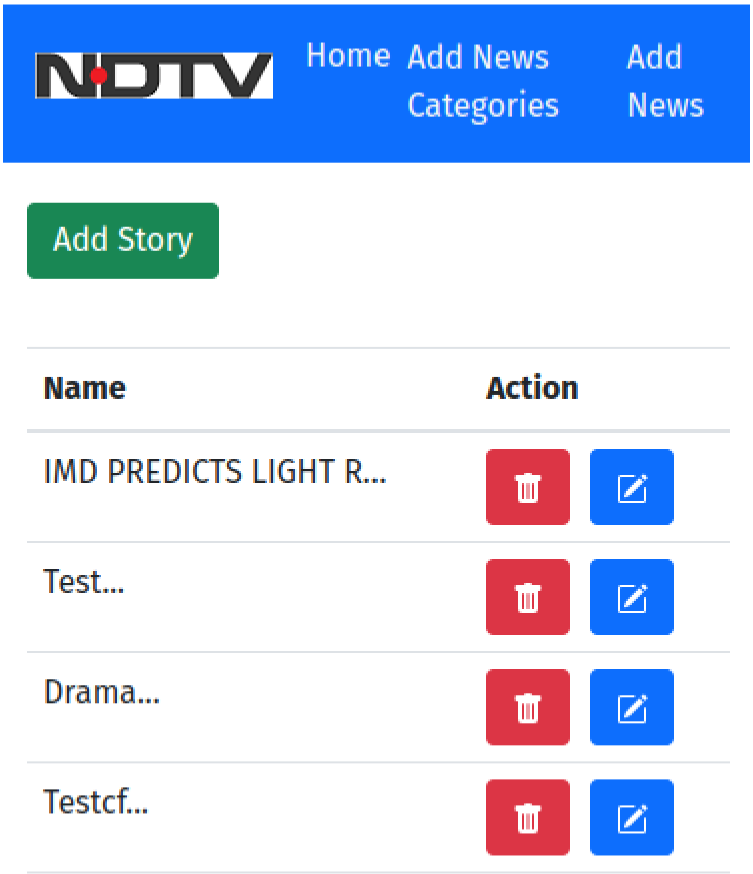
Category : Technology

Title :IMD PREDICTS LIGHT RAINFALL FOR NEXT TWO DAYS

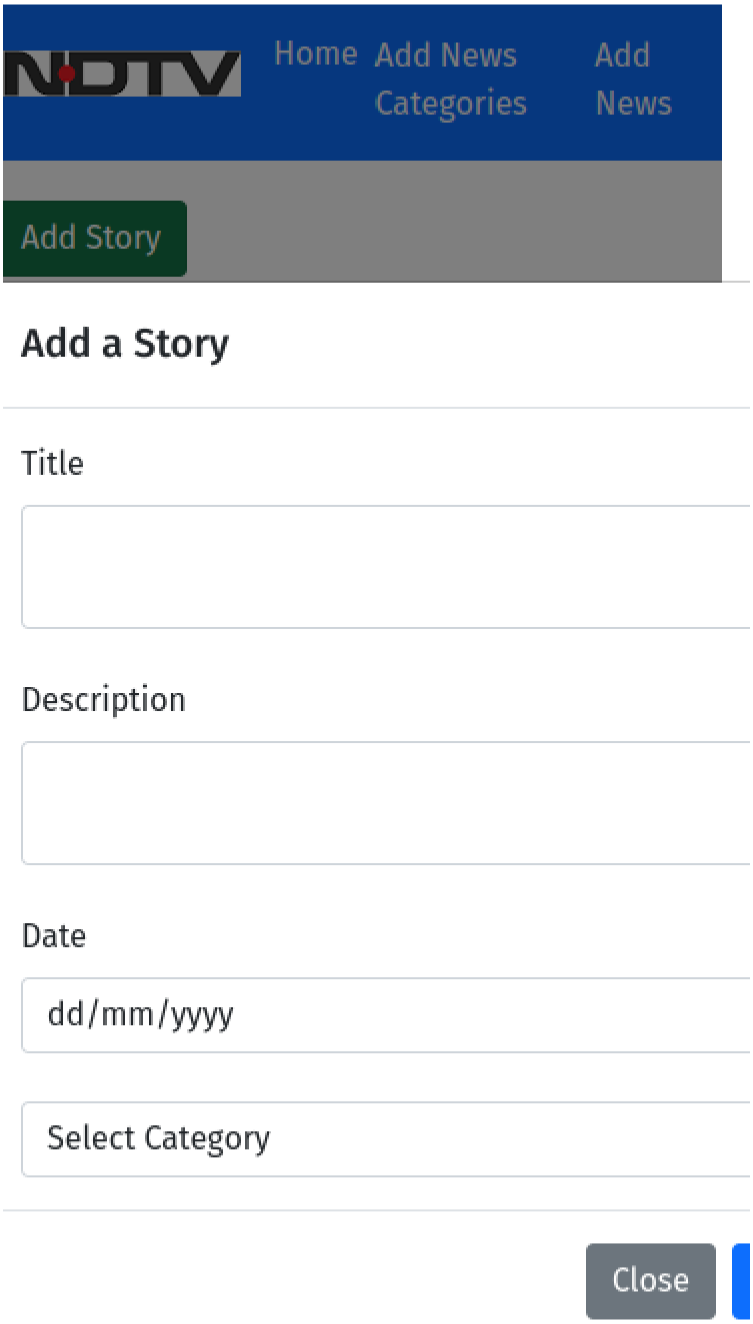
The IMD has predicted rains accompanied by thunder for the next two days. The North Westerly winds over Goa may increase the clouding over Goa and spacial distribution of rainfall also is likely to slightly increase during 24th -26th May compared to past 48 hours. The possibility of thunderstorm accompanied with lightning at one or two



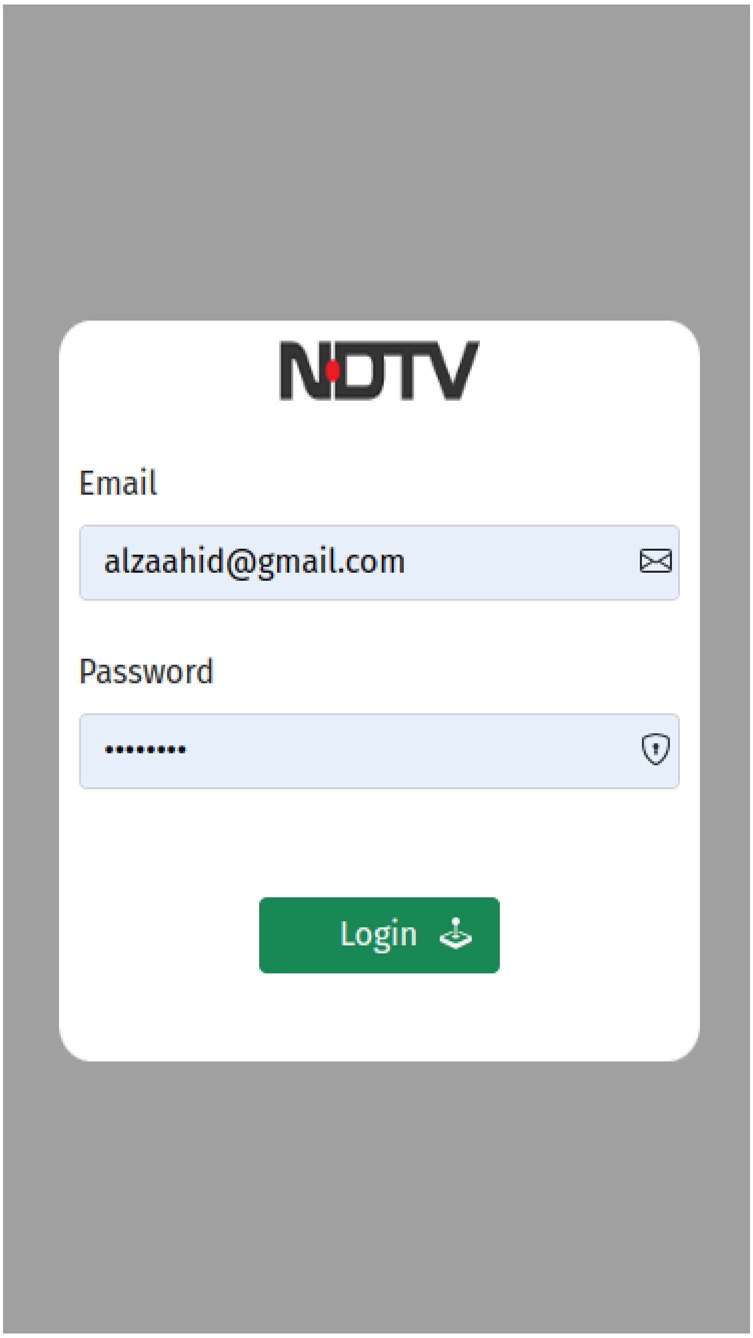
Name	Action	
Technology	🗑	✎
Drama	🗑	✎
Test	🗑	✎
ggy	🗑	✎



Name	Action
IMD PREDICTS LIGHT R...	🗑 ✎
Test...	🗑 ✎
Drama...	🗑 ✎
Testcf...	🗑 ✎



NDTV
Home
Add News Categories
Add News
Add Story
Add a Story
Title
Description
Date
dd/mm/yyyy
Select Category
Close



Email

alzaahid@gmail.com

Password

•••••••••