

Full Stack Development Course Project
“News App”

Master of Computer Application

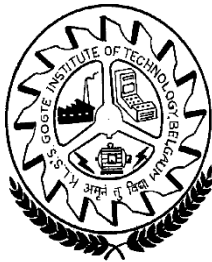
Submitted by:

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 for NewsApp.
- 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 with NewsApp swift and efficient.
- CORS: Embracing seamless cross-origin resource sharing, CORS enables NewsApp 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™ i5-5005U @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)
    }
  })
}

```



```

{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 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>


);
}

}

```

5.SCREENSHOTS

[Home](#)[Add News Categories](#)[Add News](#)



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

[Home](#) [Add News Categories](#)[Add News](#)[Add News Categories](#)**Name****Action**

Technology



Drama



Test



ggy













Home Add News
Categories

Add
News

Add Story

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

Add a Story

Title

Description

Date

Close



Email

alzaahid@gmail.com



Password

.....



Login

