# LPU TWEET ADVANCED WEB DEVELOPMENT ASSIGNMENT

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

By

**Vivek Tiwari (11702815)**

**Section: KM013**

Under the guidance of

**Ms. Neha Sharma**


**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

**OCTOBER 2020**

# ACKNOWLEDGMENT

We would like to thanks our teacher Ms. Neha Sharma for giving us this opportunity to build a real-world project using web-development tools and technologies.

“LPU TWEET” is the project that i have decided to build using powerful web-technologies and deploy it on a platform to make it available to everyone as a freeware.

This project is a complete mixture of Web-development technologies like HTML-5, CSS-3, JavaScript, jQuery, NodeJs, ExpressJs, MongoDB and Bootstrap Design for UI combined together in a best possible way to provide users an awesome experience with user-friendly interface, navigation and features throughout the application.

While developing and building this web-based application we came across various reliable features and built-in methods provided by both front-end and back-end web-technologies like routing, and many other node modules features that really helped us to achieve such a milestone.

Place : Lovely Professional University Teacher: Ms. Neha Sharma

Date : 2nd November 2020

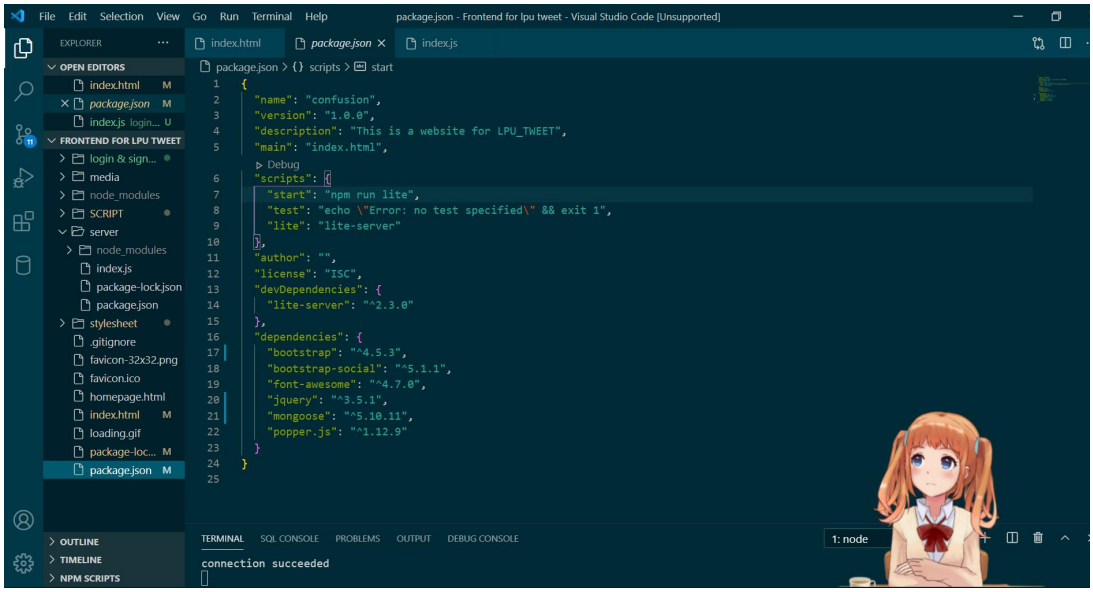
# Table of Content

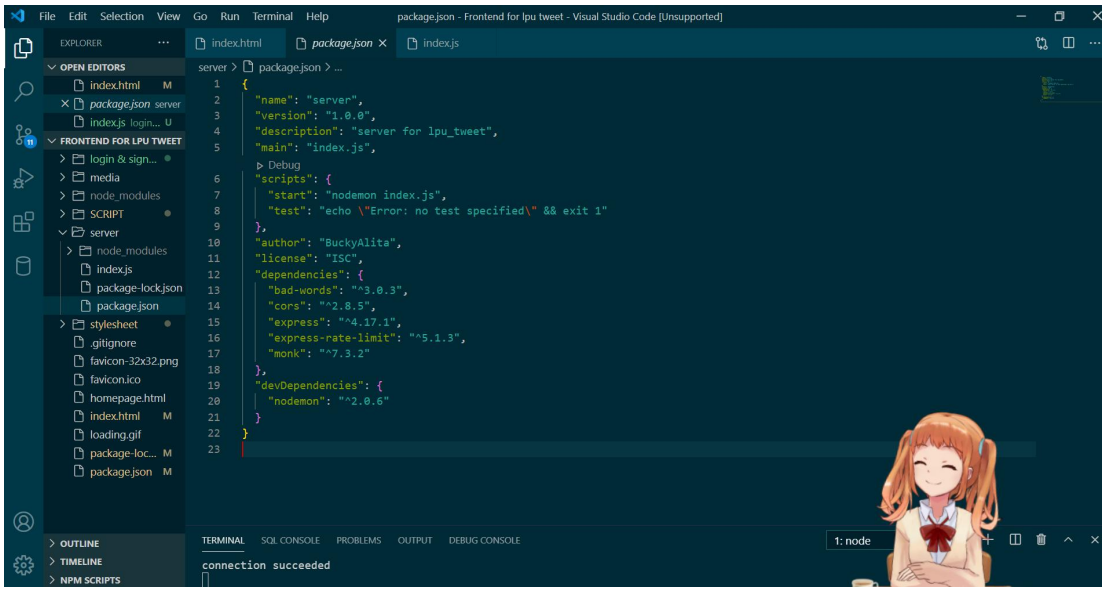
**CONTENTS** **PAGE NO.**

# 1. INTRODUCTION

LPU TWEET is a express application where user wants to tweet anything after login or signup. Whereas you have to signup if you are new member. Or you have to login if you're an member. After login you have two field one for name by which you want to show your tweet and second One is for content where you actually post what you want to.

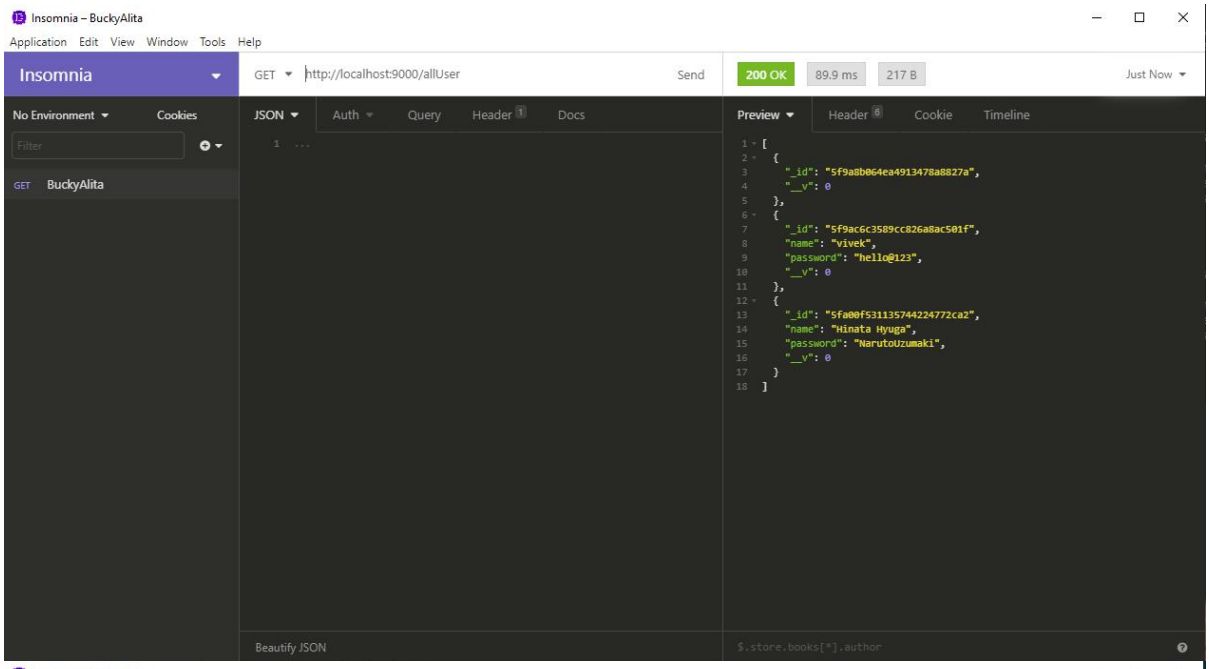
For this application we used some node modules like express, Mongoose which is an object document modeling (ODM) layer which sits on the top of Node's MongoDB driver.

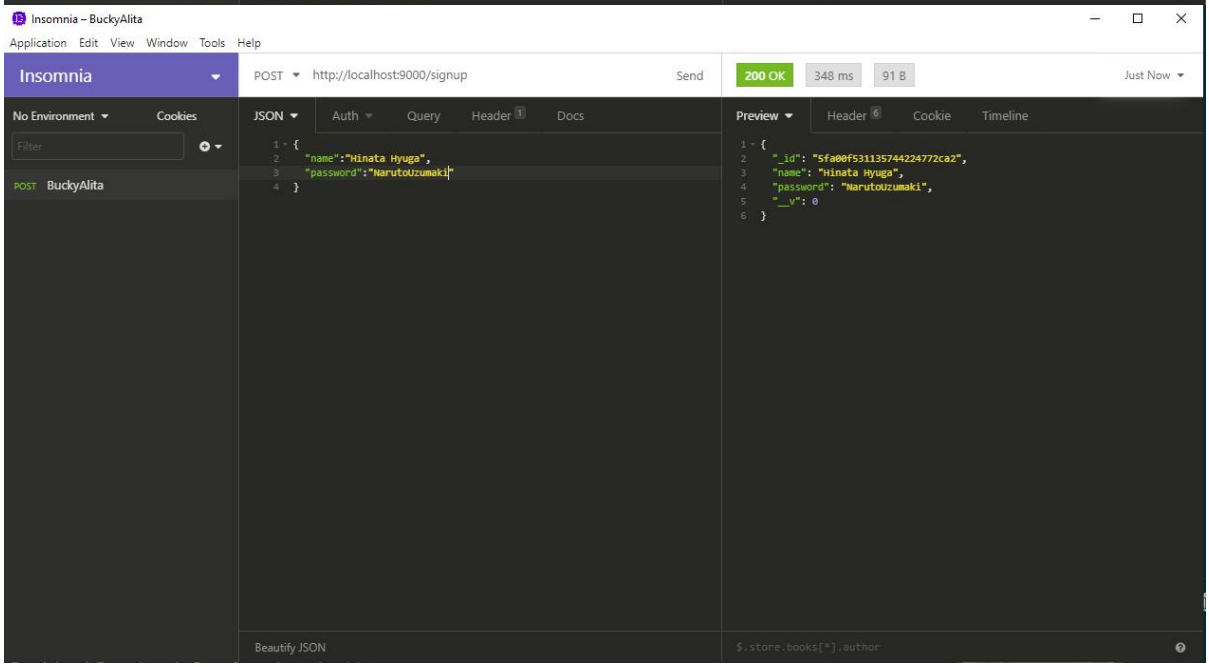


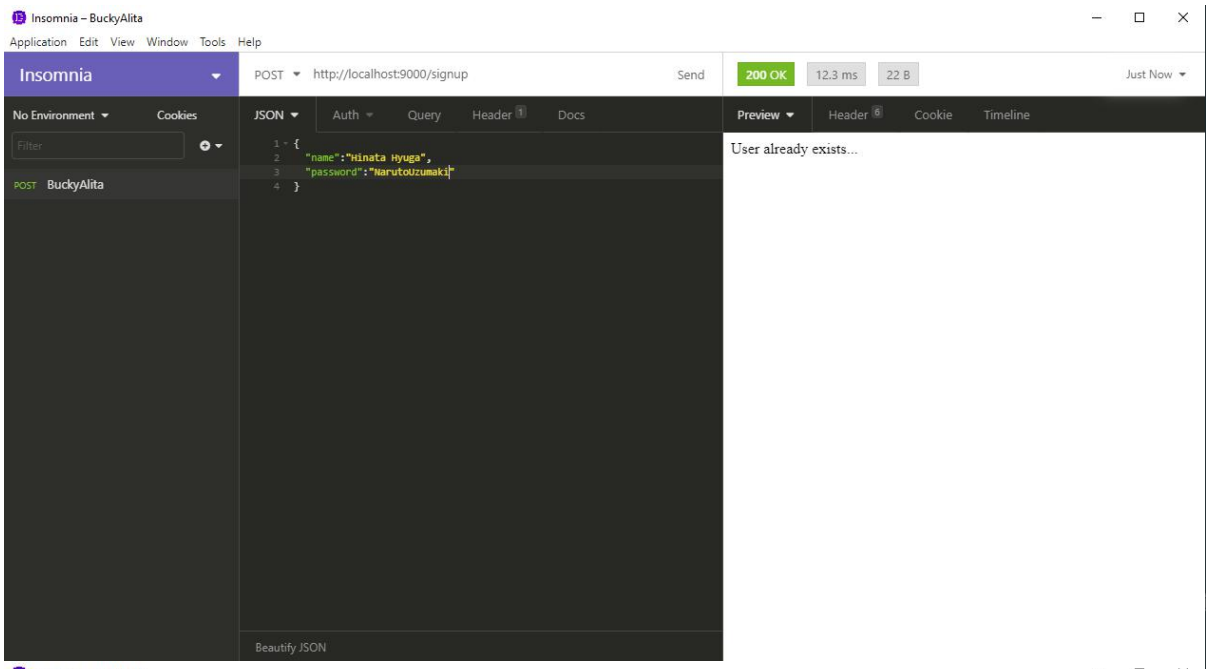



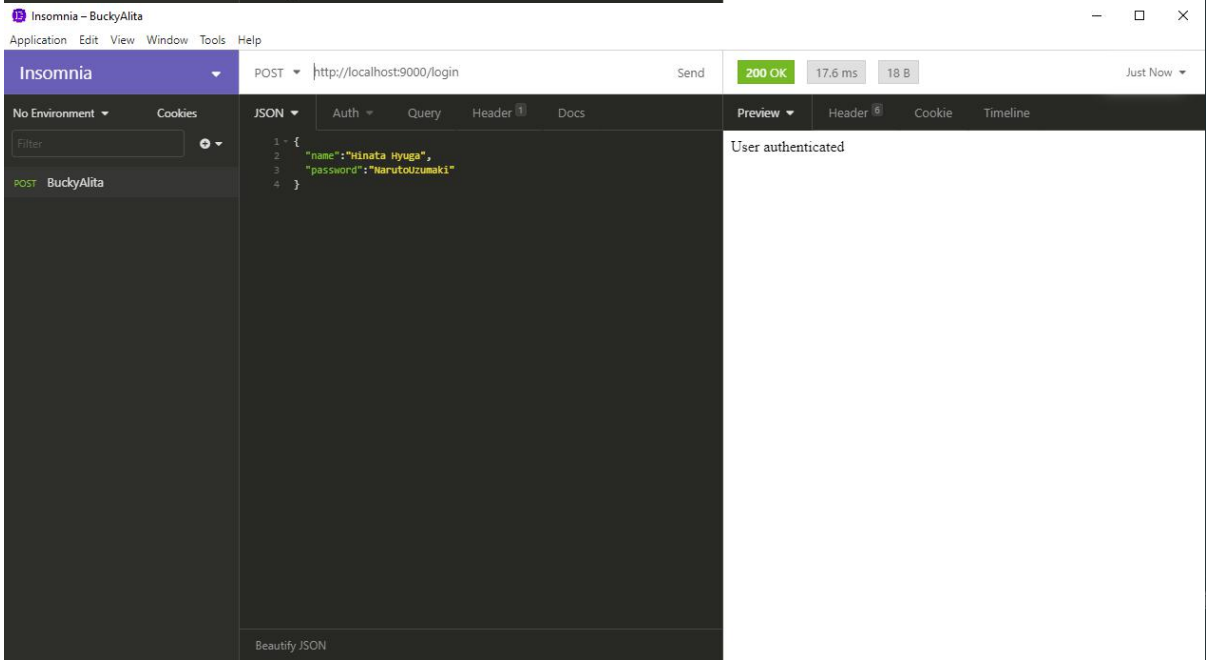
# 2. SCREENSHOTS

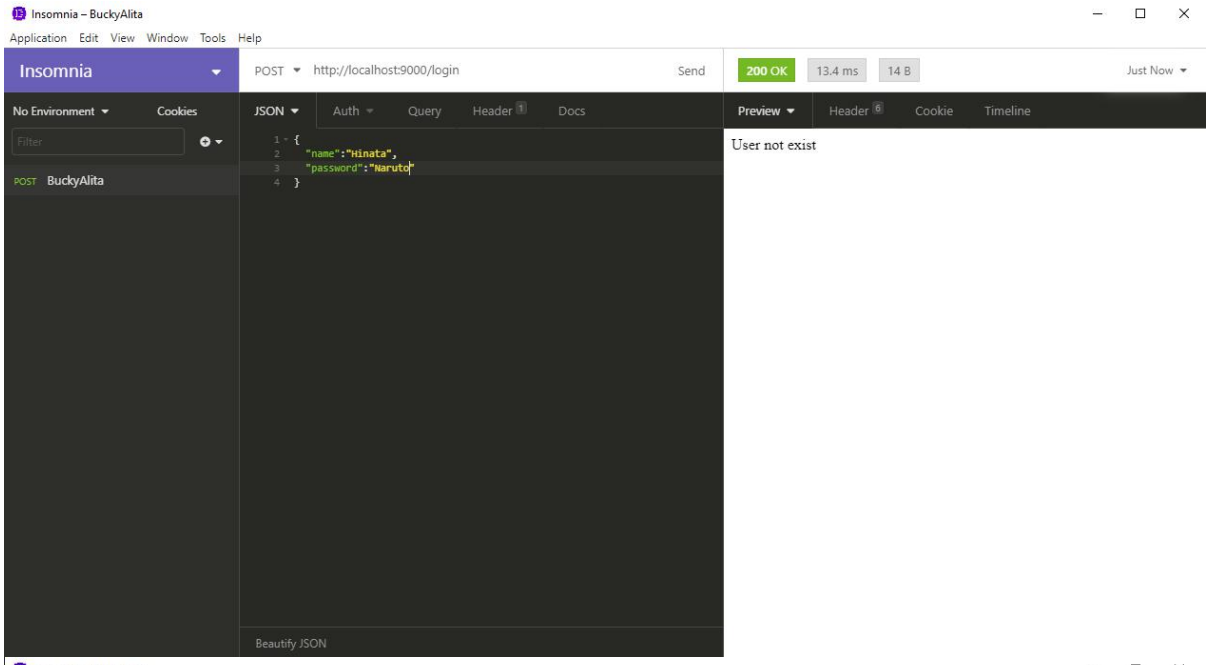
## API SCREENSHOTS : -

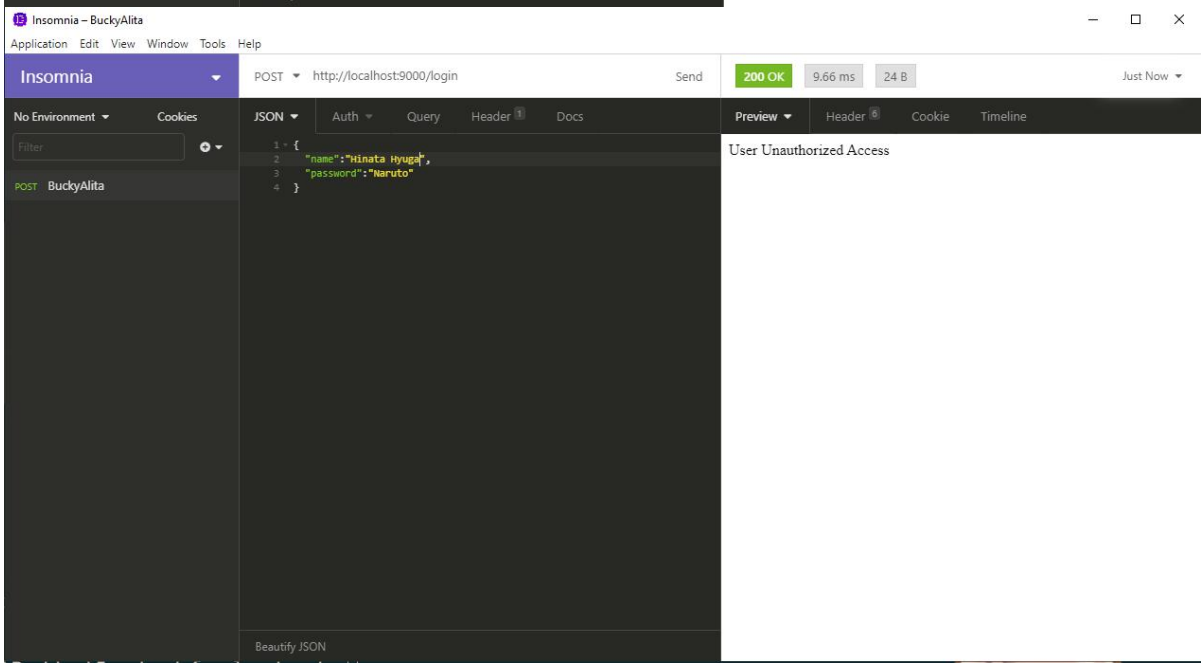



Insomnia – BuckyAlita
Application Edit View Window Tools Help
Insomnia
POST http://localhost:9000/signup
Send
200 OK
12.3 ms
22 B
Just Now
No Environment
Cookies
Filter
POST BuckyAlita
JSON
Auth
Query
Header 1
Docs
Preview
Header 6
Cookie
Timeline
{
"name":"Hinata Hyuga",
"password":"NarutoUzumaki"
}
User already exists...
Beautify JSON


Insomnia – BuckyAlita
Application Edit View Window Tools Help
Insomnia
POST http://localhost:9000/login
Send
200 OK
17.6 ms
18 B
Just Now
No Environment
Cookies
Filter
POST BuckyAlita
JSON
Auth
Query
Header 1
Docs
Preview
Header 6
Cookie
Timeline
{
"name":"Hinata Hyuga",
"password":"NarutoUzumaki"
}
User authenticated
Beautify JSON

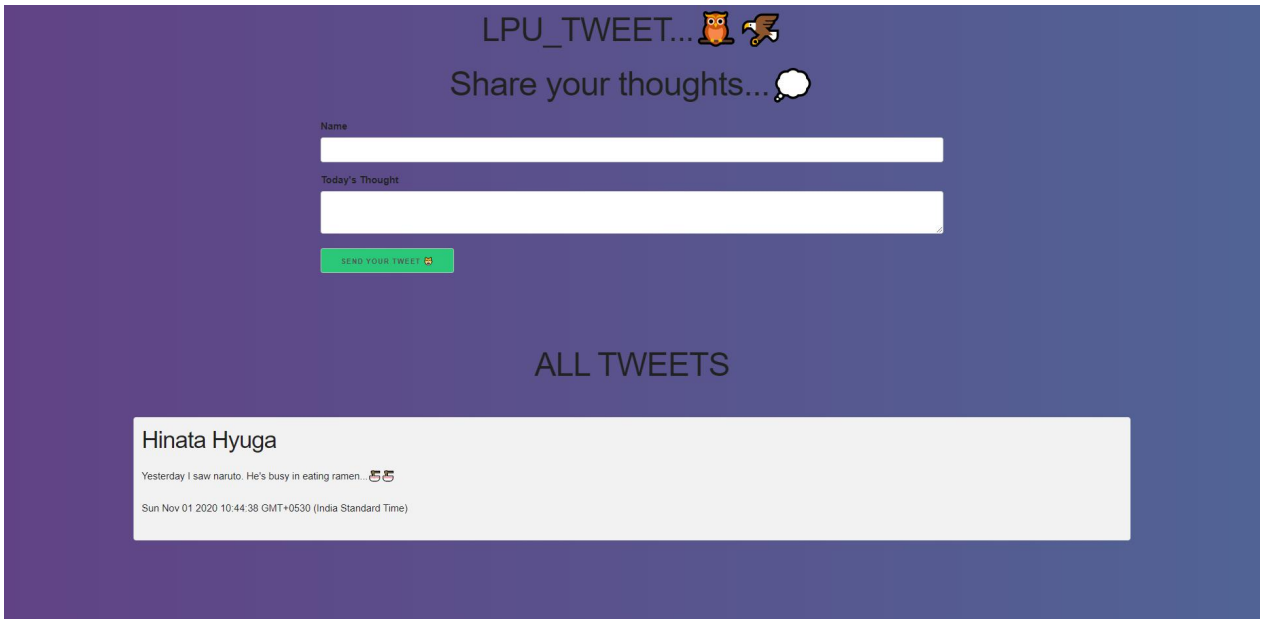

Insomnia – BuckyAlita
Application Edit View Window Tools Help
Insomnia
POST http://localhost:9000/login
Send
200 OK
13.4 ms
14 B
Just Now
No Environment
Cookies
Filter
POST BuckyAlita
JSON
Auth
Query
Header 1
Docs
Preview
Header 6
Cookie
Timeline
{
"name":"Hinata",
"password":"Naruto"
}
User not exist
Beautify JSON


Insomnia – BuckyAlita
Application Edit View Window Tools Help
Insomnia
POST http://localhost:9000/login
Send
200 OK
9.66 ms
24 B
Just Now
No Environment
Cookies
Filter
POST BuckyAlita
JSON
Auth
Query
Header 1
Docs
Preview
Header 6
Cookie
Timeline
{
"name":"Hinata Hyuga",
"password":"Naruto"
}
User Unauthorized Access
Beautify JSON


**HOME PAGE (WITHOUT ANY TWEET)**

**(**click on each button to open the login or signup modal box and check respective functionality )

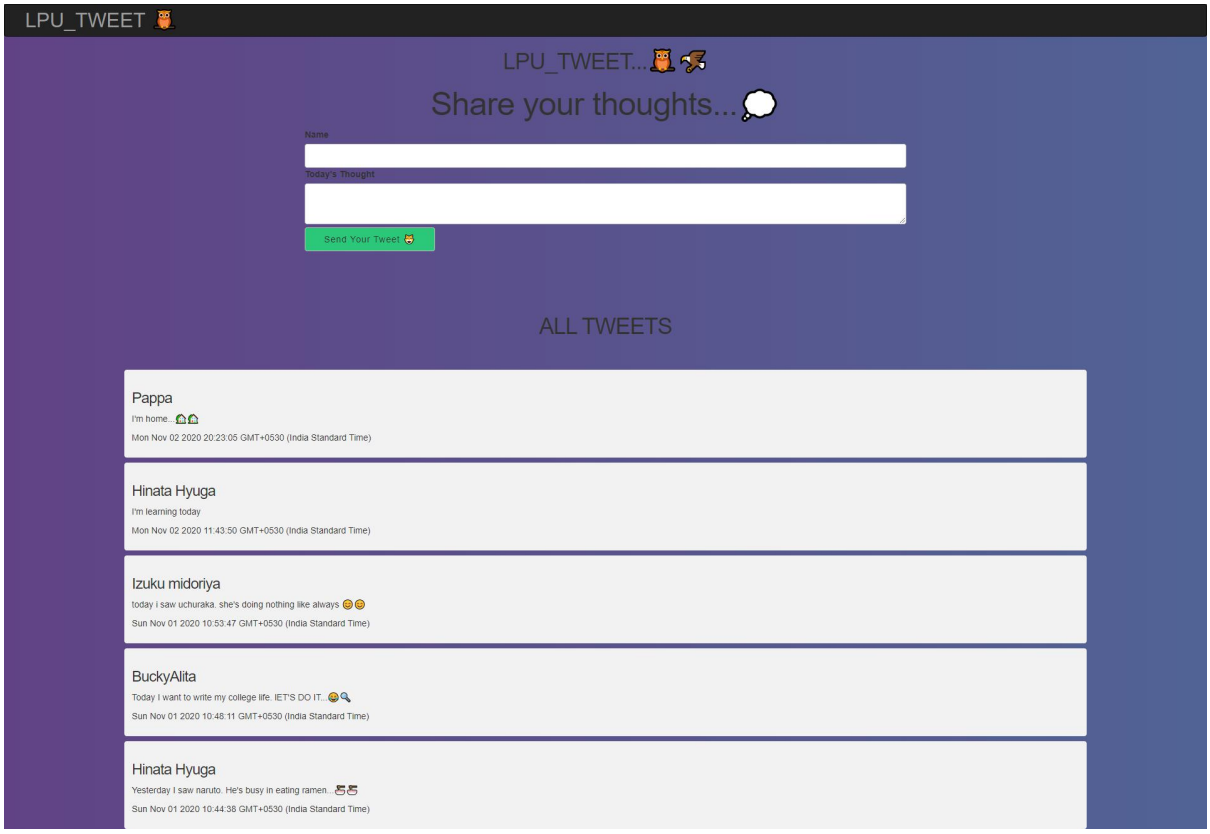

**HOME PAGE (WITH FIRST TWEET)**




# TWEET PAGE

## ( WHERE USER WRITES THERE TWEET AFTER LOGIN OR SIGNUP)


LPU_TWEET
Signup
Login
LPU_TWEET...
Where we all can tweet. whatever in our mind...
Interesting right, then what are you waiting for. Let's tweet
LOGIN (for member)
Your Name*
Your Password *
Login
Forget Password?
Signup (for new User)
Your Name *
Your Password *
Signup
Forget Password?
ALL TWEETS...
Pappa
I'm home...
Mon Nov 02 2020 20:23:05 GMT+0530 (India Standard Time)
Hinata Hyuga
I'm learning today
Mon Nov 02 2020 11:43:50 GMT+0530 (India Standard Time)
Izuku midoriya
today I saw uchuraka. she's doing nothing like always
Sun Nov 01 2020 10:53:47 GMT+0530 (India Standard Time)
BuckyAlita
Today I want to write my college life. IET'S DO IT.
Sun Nov 01 2020 10:48:11 GMT+0530 (India Standard Time)
Hinata Hyuga
Yesterday I saw naruto. He's busy in eating ramen...
Sun Nov 01 2020 10:44:36 GMT+0530 (India Standard Time)


LPU_TWEET
LPU_TWEET...
Share your thoughts...
Name
Today's Thought
Send Your Tweet
ALL TWEETS
Pappa
I'm home...
Mon Nov 02 2020 20:23:05 GMT+0530 (India Standard Time)
Hinata Hyuga
I'm learning today
Mon Nov 02 2020 11:43:50 GMT+0530 (India Standard Time)
Izuku midoriya
today i saw uchuraka. she's doing nothing like always
Sun Nov 01 2020 10:53:47 GMT+0530 (India Standard Time)
BuckyAlita
Today I want to write my college life. IET'S DO IT...
Sun Nov 01 2020 10:48:11 GMT+0530 (India Standard Time)
Hinata Hyuga
Yesterday I saw naruto. He's busy in eating ramen...
Sun Nov 01 2020 10:44:38 GMT+0530 (India Standard Time)

# 3. MAIN – CODES

## app.js

```
const express = require('express');
const cors = require('cors');
const monk = require('monk');
const Filter = require('bad-words');
const rateLimit = require('express-rate-limit');


const db = monk('localhost/BuckyAlita');
const mews = db.get('mews');
const app = express();
const filter = new Filter();

app.use(cors());
app.use(express.json())


app.get('/',(req,res)=>{
   res.json({
      message:"Hello Alita    "
   });
});

app.get('/mews',(req,res) =>{
   mews
      .find()
      .then(mews => {
         res.json(mews)
      });
});
```

```
function isValidAlita(mew)
{
   return mew.name && mew.name.toString().trim() != '' && mew.content &&
mew.content.toString().trim() != '';
}

app.use(rateLimit({
   windowMs: 30 * 1000,
   max: 1
}));

app.post('/mews',(req,res)=>{
   // mews.drop()

   // console.log(req.body);
   if(isValidAlita(req.body)){
      // insert into database
      const mew = {
         name: filter.clean(req.body.name.toString()),
         content: filter.clean(req.body.content.toString()),
         created:new Date()
      }



      mews
         .insert(mew)
         .then(createdMew => {
            res.json(createdMew);


         })
   }
   else{
      res.status(422);
      res.json({
         message:"Hey Buddy Name and Content is require for tweet. tenu inna bhii nii pta.    "
      });
   }
})
```

```
app.listen(5000,()=>{
   console.log("We're Listning BuckyAlita on port http://localhost:5000")
});
```

# client.js

```
const form = document.querySelector('form'); // grabbing an element on the page
const loadingElement = document.querySelector('.loading');
const mewsElement = document.querySelector('.mews');

const ApiUrl = "http://localhost:5000/mews";


loadingElement.style.display = '';

listAllMews();

form.addEventListener('submit', (event) => {
  event.preventDefault();
  const formData = new FormData(form);
  const name = formData.get('name');
  const content = formData.get('content');

    const mew = {
      name,
      content
    };
    loadingElement.style.display = ''
    form.style.display = 'none';

  fetch(ApiUrl,{
      method:'POST',
      body:JSON.stringify(mew),
      headers:{
          'content-type':'application/json'
      }
  }).then(response => response.json())
    .then(createdMew => {
        form.reset();
        form.style.display = '';
        listAllMews();

    })

});


function listAllMews(){
```

```
  mewsElement.innerHTML = '';
  fetch(ApiUrl)
  .then(response => response.json())
  .then(mews => {
     mews.reverse();
     mews.forEach(mew => {
       const div = document.createElement('div');

       const header = document.createElement('h3');
       header.textContent = mew.name;

       const contents = document.createElement('p');
       contents.textContent = mew.content;

       const date = document.createElement('p');
       date.textContent = new Date(mew.created);

       div.appendChild(header);
       div.appendChild(contents);
       div.appendChild(date);

       mewsElement.appendChild(div);
     });
     loadingElement.style.display = 'none'
  });
}


// function listAllMews(){
//   fetch(ApiUrl)
//   .then(response => response.json())
//   .then(mews => {
//     console.log(mews);
//   });
// }
```

## 4. TECHNOLOGY / MODULES USED

Web technologies used in this project are:

- HTML-5
- CSS-3
- JavaScript
- jQuery
- Bootstrap
- NodeJS(javascript runtime environment)
- MongoDB

---

## 6. GitHub – Link

**https://github.com/Tiwari007/NodeJsProject**