PROJECT REPORT

DREAM THOUGHTS-

SHORT STORIES & THESIS

Submitted by

REMYA TR

Roll No.: 012

ADIT (2019-2021)

National Skill Training Institute for Women, Trivandrum

ABSTRACT

THE PROJECT TITLED: *DREAM THOUGHTS-SHORT STORIES & THESIS*

Main objective of this project is to help all writers to write their short stories as well as thesis. In this platform, A user can create their thoughts, Short stories and thesis, read & update these all, finally the mistakes delete as per the writers wish. I often recommend beginning writers to still write on the old fashioned pen and paper. There's something magical about the old-fashioned way that just works. The posting style on Commaful is a bit different than traditional writing, so if you're looking to write just copy and paste a novel onto the site, then this probably isn't the best site for you. Short stories are best for this site .

CONTENTS

1. INTRODUCTION

1.1OBJECTIVE

The short story is often described as one of the most satisfying literary genres, both to read and to write. Digital writing is not simply a matter of learning about and integrating new digital tools into an unchanged repertoire of writing processes, practices, skills, and habits of mind. Digital writing is about the dramatic changes in the ecology of writing and communication and, indeed, what it means to write—to create and compose and share.

2. SYSTEM ANALYSIS

2.1EXISTING SYSTEM

Writers to still write on the old fashioned pen and paper. So due to this lot of problems will happen.

- Lack of storage space. ...
- Security issues. ...
- Prone to damage. ...
- Document transportation. ...
- Editing problems. ...
- High costs. ...
- Limit communication and collaboration. ...
- Environmental damage.

2.2PROPOSED SYSTEM

Typing is a fast and easy way to take the information presented in lectures and textbooks and consolidate them for reference later. But, due to its fast nature, this method leads to the least amount of information retained and will require you to study more later.

- Time Independence.
- Location Independence.
- Say Goodbye to Commuting.
- Sleep When You Want.
- Security. ...
- More Interesting Work.

3. TECHNOLOGY OVERVIEW

The technology selected for implementing writing platform is NodeJS, MySQL,

HTML And CSS.

NodeJS

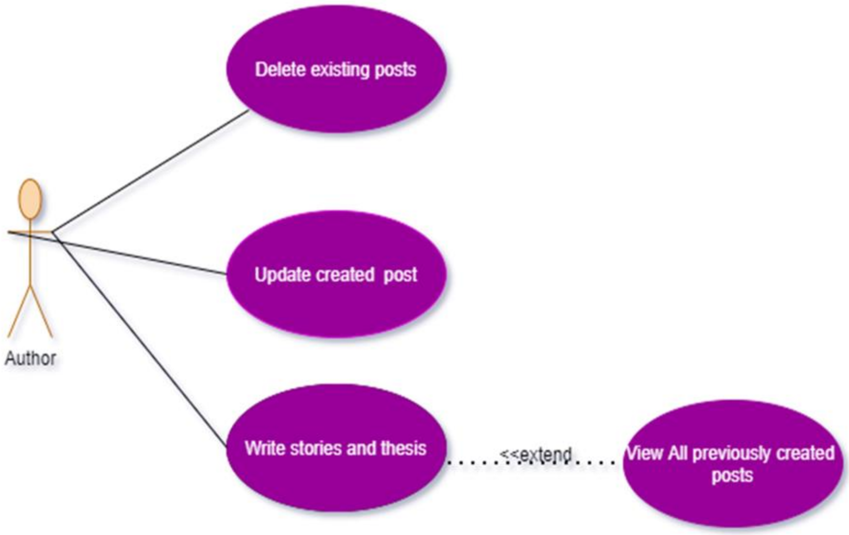
Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its official documentation.

MySQL

MySQL is a relational database management system (RDBMS) [1] that runs as a server providing multi-user access to a number of databases. MySQL is a popular choice of database for use in web applications and is an open source product. The process of setting up a MySQL database varies from host to host, however we will end up with a database name, a user name and a password.

4. SYSTEM DESIGN

4.1. USE CASE DIAGRAM



5. SYSTEM REQUIREMENTS

5.1. SOFTWARE SPECIFICATION

Operating System	:	Windows 7 or above
Front End	:	HTML, CSS, Node.JS
Back End	:	MySQL
Code Editor	:	Notepad++, Visual Code
Software	:	XAMPP Server
Server	:	local host
Web Browser	:	Mozilla Firefox

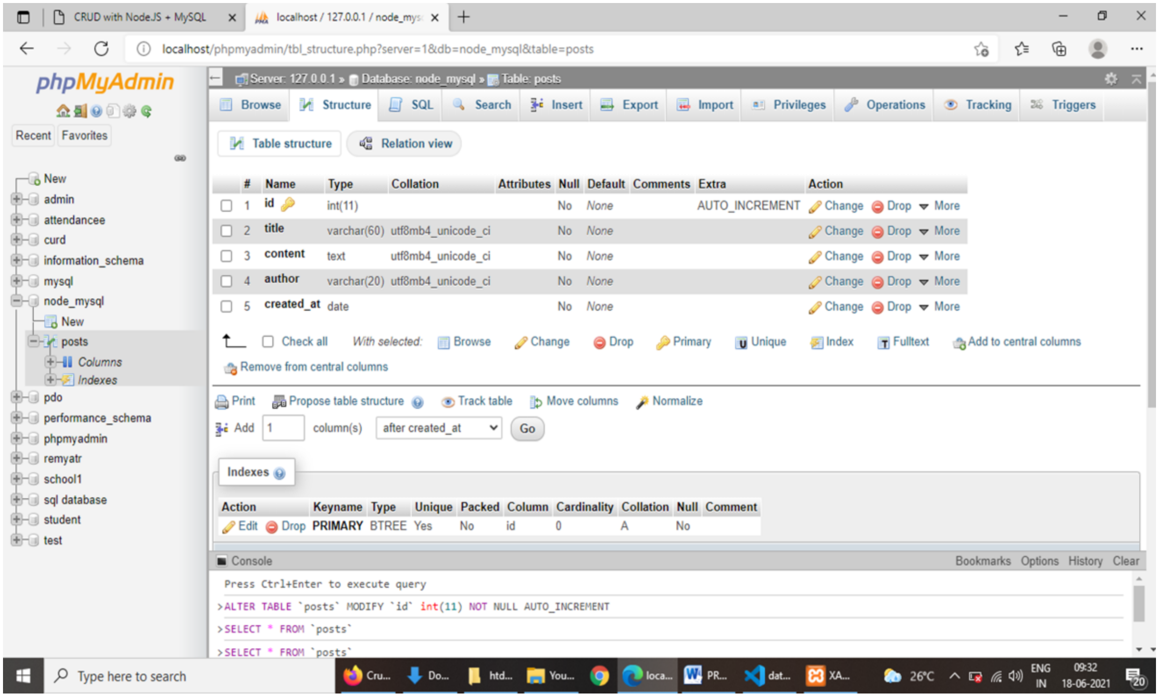
5.2. HARDWARE SPECIFICATION

RAM	:	1 GB or above

Processor	:	1 GHz or more
Hard Drive	:	32 GB or above
Network Connectivity :		LAN or Wi-Fi

6. APPENDICES

6.1. DATABASE TABLES



localhost/phpmyadmin/tbl_structure.php?server=1&db=node_mysql&table=posts

Server: 127.0.0.1 » Database: node_mysql » Table: posts

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
□	2	title	varchar(60)	utf8mb4_unicode_ci		No	None			Change Drop More
□	3	content	text	utf8mb4_unicode_ci		No	None			Change Drop More
□	4	author	varchar(20)	utf8mb4_unicode_ci		No	None			Change Drop More
□	5	created_at	date			No	None			Change Drop More

□ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after created_at Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	

Console Bookmarks Options History Clear

```
Press Ctrl+Enter to execute query
>ALTER TABLE `posts` MODIFY `id` int(11) NOT NULL AUTO_INCREMENT
>SELECT * FROM `posts`
>SELECT * FROM `posts`
```

6.2. SOURCE CODE

1. index.html

```
<!DOCTYPE html>
```

```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Dream Thoughts</title>
    <!-- USING BOOTSTRAP FOR STYLING USER INTERFACE -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">
    <style>
        .customRow{
            background-image: rgb(16, 16, 78);
            padding: 20px;
        }
</style>
</head>
<body bgcolor="red">
    <div class="customRow">
        <div class="container">
            <div class="card">
                <div class="card-body">
                    <h1 class="text-center">Short Stories & Thesis</h1>
                    <hr>
                    <form action="" method="POST">
                        <div class="form-group">
                            <label for="post_title"> Title</label>
                            <input type="text" name="title" class="form-control" placeholder="Title" id="post_title" required>
                            <label for="post_content">Content</label>
                            <textarea name="content" class="form-control" placeholder="Write something" id="post_content" required ></textarea>
                            <label for="author_name">Author Name</label>
                            <input type="text" name="author_name" class="form-
```

```
control" placeholder="Enter author name" id="author_name" required>
                            <br>
                            <input type="submit" value="POST" class="btn btn-primary">
                        </div>
                    </form>
                    <hr>
                    <h2 class="text-center">All Writings</h2>
                    <!-- IF HAVE ANY POSTS -->
                    {% if posts|length > 0 %}
                        <!-- LOOPING ALL THE POSTS -->
                        {% for post in posts %}
                            <div class="card">
                                <div class="card-body">
                                    <h5 class="card-title">{{ post.title | e }} | <small><span>{{ post.created_at | date("d M, Y") }}</span></small></h5>
                                    <p class="card-text text-justify">{{ post.content | e }}</p>
                                    <span class="float-right"><strong>By</strong>, {{ post.author | e }}</span>
                                    <a href="/edit/{{ post.id | e }}" class="btn btn-light">✎ Edit</a>
                                    <a href="/delete/{{ post.id | e }}" class="btn btn-danger">Delete</a>
                                </div>
                            </div>
                            <hr>
                        {% endfor %}
                    {% else %}
                        <h4>No Writings Found !</h4>
                    {% endif %}
                </div>
            </div>
        </div>
    </div>
```

```
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k" crossorigin="anonymous"></script>
</body>

</html>
```

2. edit.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Edit your Writings</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">
  <style>
    .customRow{
      background-color: black;
      padding: 20px;
    }
</style>
</head>
```

```
<body>
    <div class="customRow">
        <div class="container">
            <div class="card">
                <div class="card-body">
                    <h1 class="text-center">Short Stories And Thesis</h1>
                    <hr>
                    {% if post is empty %}
                        <h3>Invalid writing ID</h3>
                    {% else %}
                        <form action="" method="POST">
                            <div class="form-group">
                                <label for="post_title">Title </label>
                                <input type="text" name="title" class="form-control" placeholder="Title" id="post_title" value="{{ post.title | e }}" required>
                                <label for="post_content"> Content</label>
                                <textarea style="height:100px;" name="content" class="form-control" placeholder="Write something" id="post_content" required>{{ post.content | e }}</textarea>
                                <label for="author_name">Author Name</label>
                                <input type="text" name="author_name" class="form-control" placeholder="Enter author name" id="author_name" value="{{ post.author | e }}" required>
                                <br>
                                <input type="submit" value="UPDATE" class="btn btn-success">
                            </div>
                        </form>
                    {% endif %}
                </div>
            </div>
        </div>
    </div>
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k" crossorigin="anonymous"></script>
</body>

</html>
```

3. app.js

```
const express = require('express');
const app = express();
const twig = require('twig');
const bodyParser = require('body-parser');

// IMPORT DB CONNECTION
const connection = require('./config/database');

// SET VIEW ENGINE
app.set('view engine','html');
app.engine('html', twig.__express);
app.set('views','views');

// USE BODY-PARSER MIDDLEWARE
app.use(bodyParser.urlencoded({extended:false}));

app.get('/', (req, res) => {
    // FETCH ALL THE POSTS FROM DATABASE
    connection.query('SELECT * FROM `posts`', (err, results) => {
        if (err) throw err;
    // RENDERING INDEX.HTML FILE WITH ALL POSTS
    res.render('index',{
      posts:results
    });
  });

});

// INSERTING POST
app.post('/', (req, res) => {
  const title = req.body.title;
  const content = req.body.content;
  const author_name = req.body.author_name;
  const post = {
    title: title,
    content: content,
    author: author_name,
    created_at: new Date()
  }
  connection.query('INSERT INTO `posts` SET ?', post, (err) => {
    if (err) throw err;
    console.log('Data inserted');
    return res.redirect('/');
  });
});

// EDIT PAGE
app.get('/edit/:id', (req, res) => {
  const edit_postId = req.params.id;
  // FIND POST BY ID
  connection.query('SELECT * FROM `posts` WHERE id=?',[edit_postI
d] , (err, results) => {
    if (err) throw err;
    res.render('edit',{
      post:results[0]
    });
  });
});
```

```
// POST UPDATING
app.post('/edit/:id', (req, res) => {
  const update_title = req.body.title;
  const update_content = req.body.content;
  const update_author_name = req.body.author_name;
  const userId = req.params.id;
  connection.query('UPDATE `posts` SET title = ?, content = ?, author = ? WHERE id = ?', [update_title, update_content, update_author_name, userId], (err, results) => {
    if (err) throw err;
    if(results.changedRows === 1){
      console.log('Post Updated');
      return res.redirect('/');
    }
  });
});

// POST DELETING
app.get('/delete/:id', (req, res) => {
  connection.query('DELETE FROM `posts` WHERE id = ?', [req.params.id], (err, results) => {
    if (err) throw err;
    res.redirect('/');
  });
});
// SET 404 PAGE
app.use('/',(req,res) => {
  res.status(404).send('<h1>404 Page Not Found!</h1>');
});


// IF DATABASE CONNECTION IS SUCCESSFUL
connection.connect((err) => {
  if (err) throw err;
  app.listen(3000);
});
```

4. package.json

```
      {
 "name": "post_management",
 "version": "1.0.0",
 "description": "Post management system with nodejs and mysql",
 "main": "app.js",
 "scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
 },
 "keywords": [
  "crud",
  "mysql"
 ],
 "author": "remya_tr",
 "license": "ISC",
 "dependencies": {
  "body-parser": "^1.19.0",
  "express": "^4.17.1",
  "mysql2": "^2.2.5",
  "twig": "^1.15.4"
 }
}
```

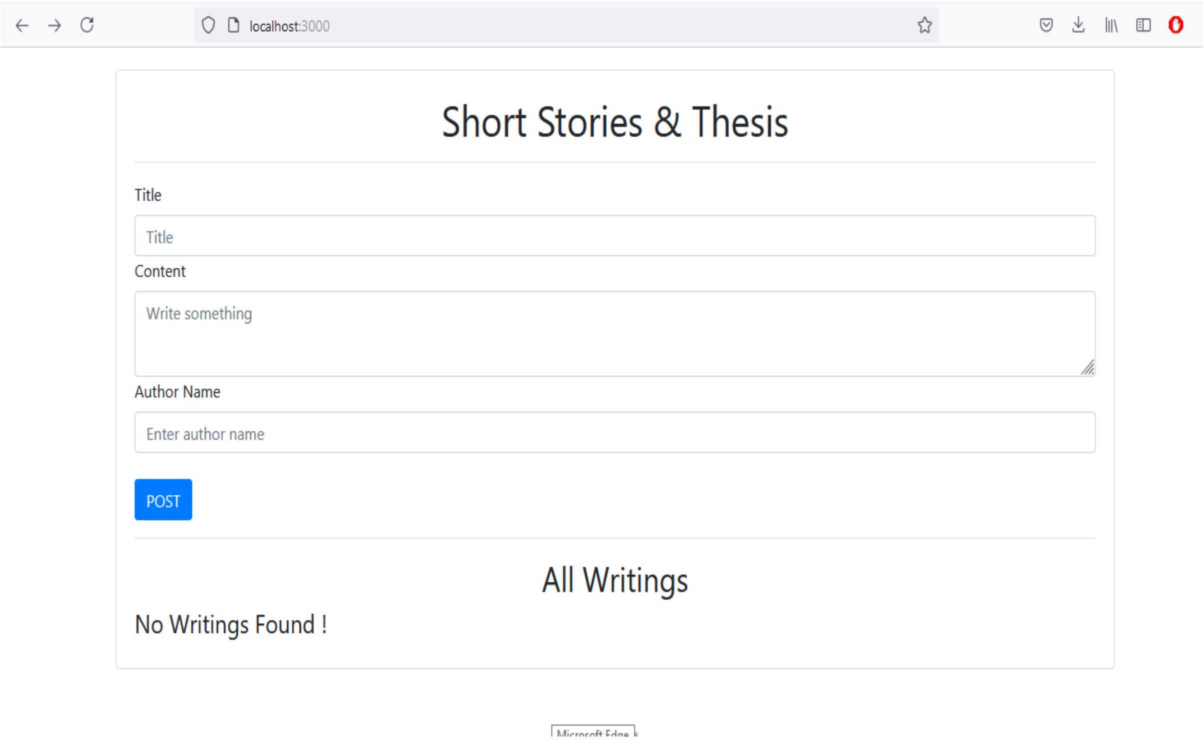
5. database

```
const mysql = require('mysql2');
const connection = mysql.createConnection({
   host     : 'localhost', // MYSQL HOST NAME
   user     : 'root',        // MYSQL USERNAME
   password : '',    // MYSQL PASSWORD
   database : 'node_mysql'      // MYSQL DB NAME
});
```

```
module.exports = connection;
```

6.3. SCREENSHOTS

1.Home page

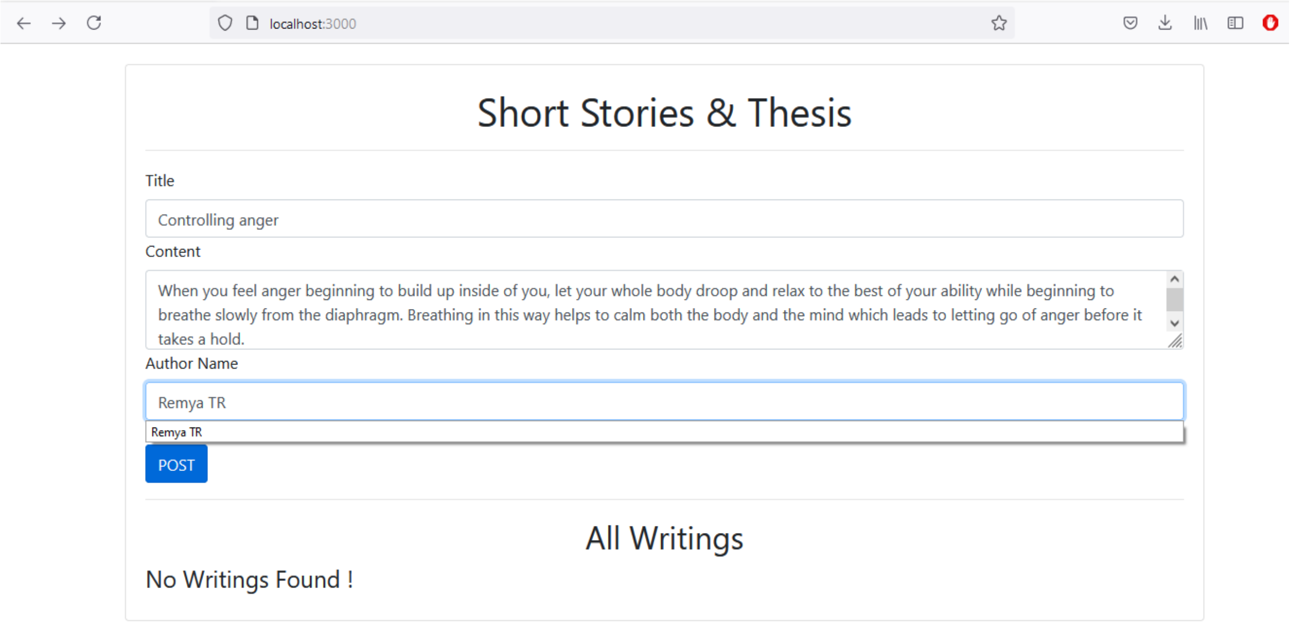


Short Stories & Thesis

All Writings

No Writings Found !

2.Add Title,Content and Author name ,After that POST



Short Stories & Thesis

Title

Controlling anger

Content

When you feel anger beginning to build up inside of you, let your whole body droop and relax to the best of your ability while beginning to breathe slowly from the diaphragm. Breathing in this way helps to calm both the body and the mind which leads to letting go of anger before it takes a hold.

Author Name

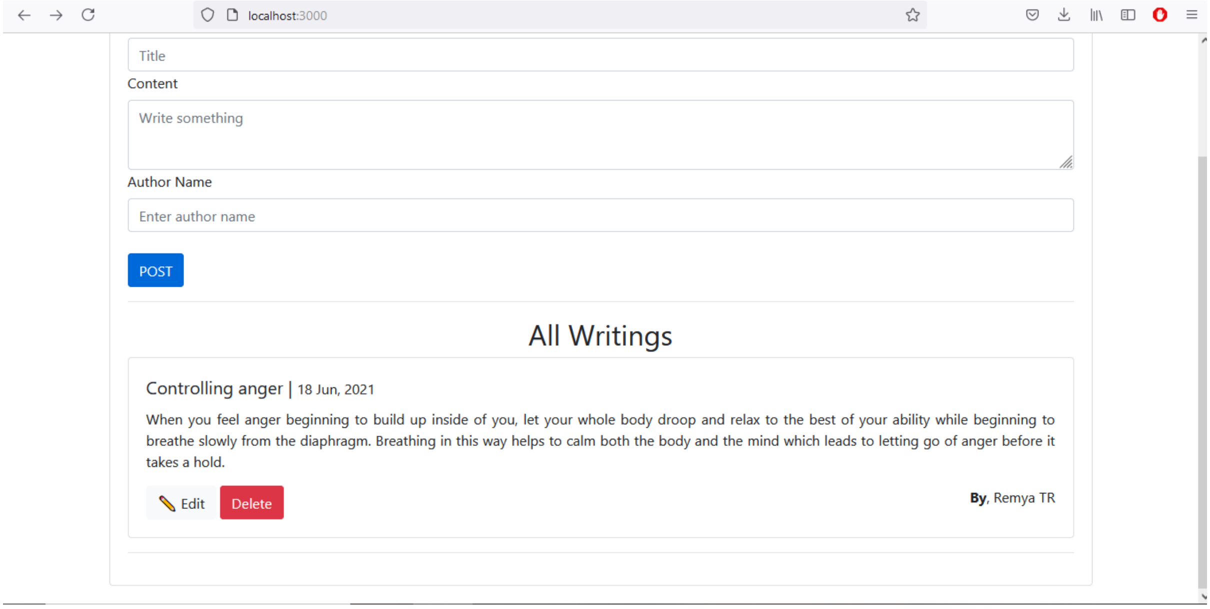
Remya TR

Remya TR

All Writings

No Writings Found !

3.



Title

Content

Write something

Author Name

Enter author name

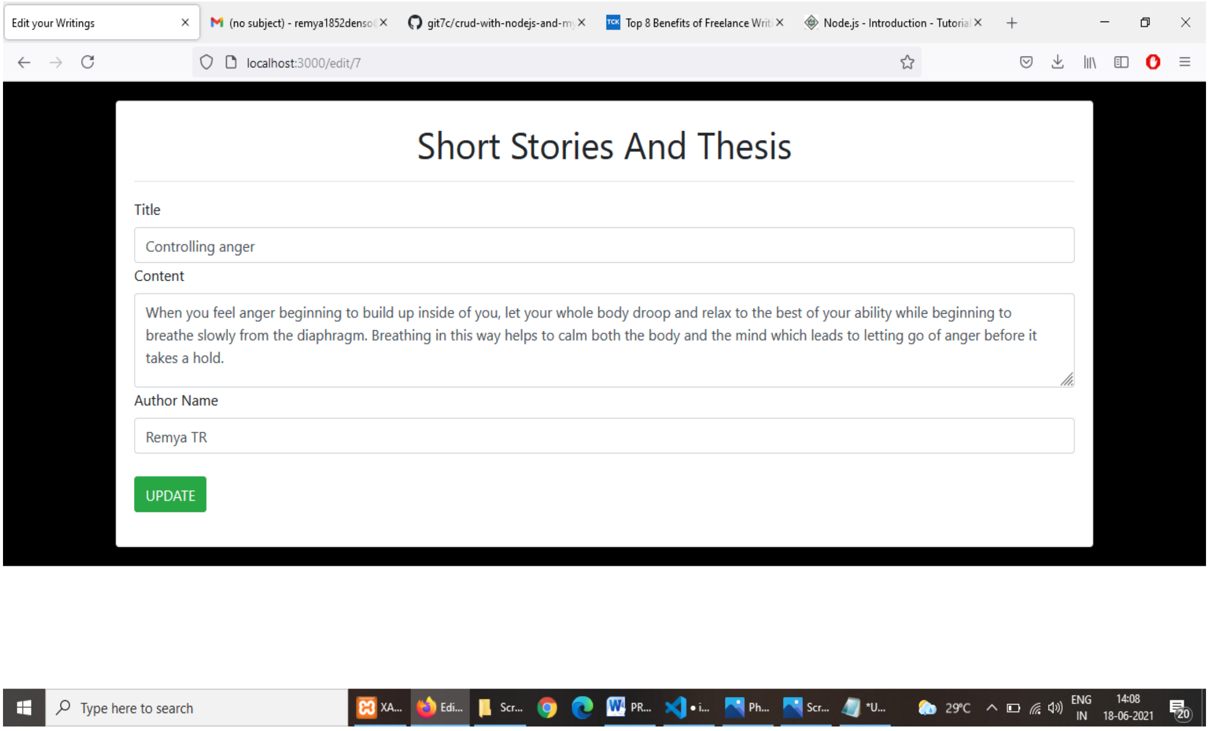
All Writings

Controlling anger | 18 Jun, 2021

When you feel anger beginning to build up inside of you, let your whole body droop and relax to the best of your ability while beginning to breathe slowly from the diaphragm. Breathing in this way helps to calm both the body and the mind which leads to letting go of anger before it takes a hold.

By, Remya TR

4. For edit click on edit, then this will appear ,After edition UPDATE it.



Short Stories And Thesis

Title

Controlling anger

Content

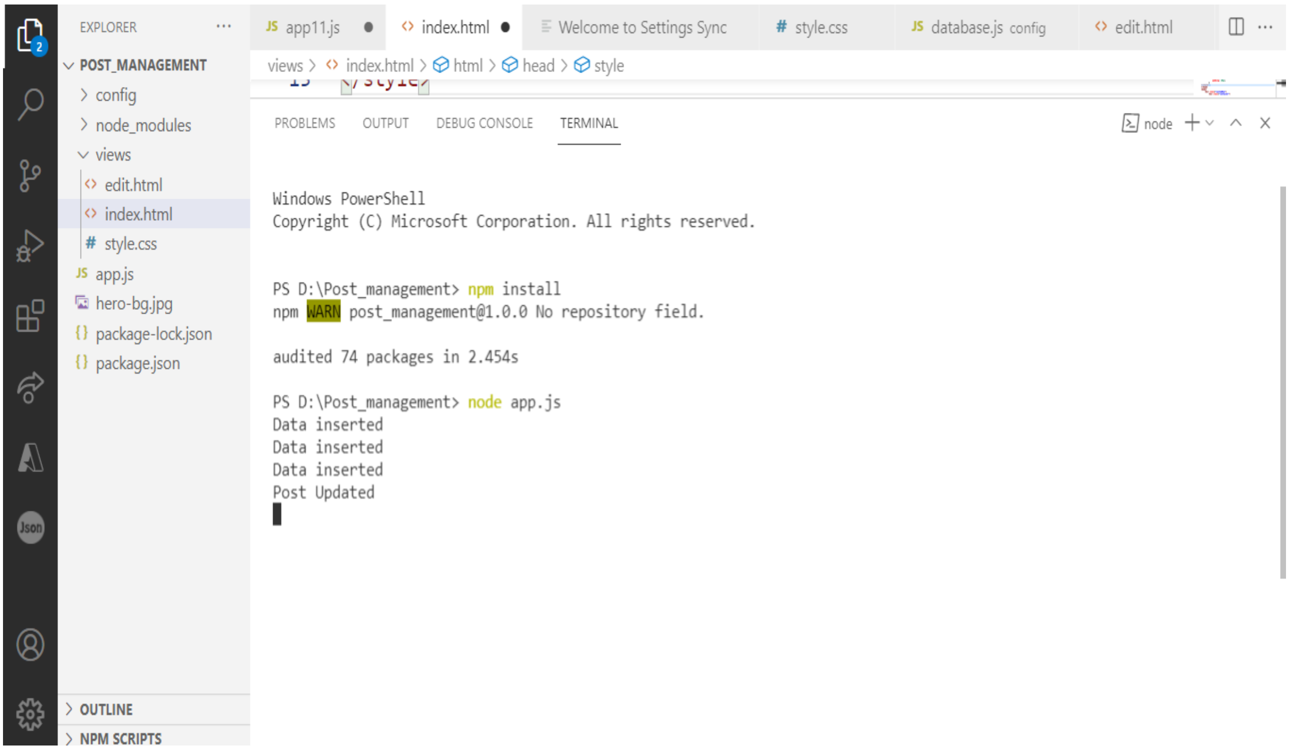
When you feel anger beginning to build up inside of you, let your whole body droop and relax to the best of your ability while beginning to breathe slowly from the diaphragm. Breathing in this way helps to calm both the body and the mind which leads to letting go of anger before it takes a hold.

Author Name

Remya TR

UPDATE

5.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS D:\Post_management> npm install
npm WARN post_management@1.0.0 No repository field.

audited 74 packages in 2.454s

PS D:\Post_management> node app.js
Data inserted
Data inserted
Data inserted
Post Updated
```

7. CONCLUSION

In this writing platform lead to a better inspiration to all writers **.** There's something magical about the old-fashioned way that just works. The posting style on Commaful is a bit different than traditional writing, so if you're looking to write just copy and paste a novel onto the site, then this probably isn't the best plateform for you

8. REFERENCE

- www.google.com.
- www.wikipedia.com
- **https://degree.astate.edu/articles/undergraduate-studies/writing-for-online-and-print.aspx**