

รายงานประกอบรายวิชา
เทคโนโลยีการเขียนโปรแกรมเว็บไซต์
Web Programming Technology
รหัสวิชา 226297

เสนอ เว็บแอปพลิเคชัน

ผู้จัดทำ

65021600 นายณัฐวุฒิ วัฒนมาลา
65021824 นายณฤพนธ์ วงศ์ชัย
65021699 นายธนพล ยะใหม่วงศ์

ผู้สอน

อาจารย์ วัฒนพงศ์ สุทธภักดิ์

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศ
และการสื่อสาร มหาวิทยาลัยพะเยา
ภาคการศึกษา ปลาย ประจำปีการศึกษา 2566

บทที่ 1

ที่มาของเว็บไซต์

1. ที่มาและความสำคัญ

ปัจจุบันการพัฒนาเทคโนโลยีทำให้เป็นไปได้ที่จะสร้างแพลตฟอร์มที่มีความสามารถในการโพสต์ข้อมูลและไฟล์ในรูปแบบต่าง ๆ ออนไลน์ ทำให้ผู้ใช้งานสามารถแบ่งปันข้อมูลและความคิดเห็นได้อย่างสะดวกสบายโซเชียลมีเดียเป็นส่วนสำคัญของวัฒนธรรมออนไลน์ในปัจจุบัน ผู้คนสนใจการแชร์และการสื่อสารออนไลน์มากขึ้น การสร้างเว็บที่มีความคล้ายคลึงกับโซเชียลมีเดียทำให้ผู้ใช้งานรู้สึกคุ้นเคยและสะดวกในการใช้งานผู้ใช้งานในยุคปัจจุบันมีความต้องการที่สูงขึ้นในการแลกเปลี่ยนข้อมูล แชร์ประสบการณ์ต่าง ๆ และสื่อสารกับคนรอบตัว ทำให้เว็บไซต์ที่สามารถทำทั้งหมดนี้กลายเป็นที่ต้องการ

เว็บไซต์นี้ถูกสร้างขึ้นเพื่อให้ผู้ใช้งานสามารถสื่อสารและเชื่อมโยงกับเพื่อนหรือกลุ่มที่สนใจ ผ่านการโพสต์ข้อความ รูปภาพ หรือไฟล์ PDF พร้อมแสดงความคิดเห็นในเนื้อหาของผู้อื่น

2. วัตถุประสงค์

- เพื่อให้ผู้ใช้งานสามารถสื่อสารและเชื่อมโยงกับเพื่อนหรือกลุ่มที่สนใจ
- เพื่อแบ่งปันข้อมูลต่าง ๆ ที่สนใจ ไม่ว่าจะเป็นข่าวสาร ความรู้ หรือเนื้อหาที่น่าสนใจ
- เก็บและค้นหาไฟล์ที่อัปโหลดตลอดจนบนเว็บได้

3. ประโยชน์ที่ได้รับ

- เว็บไซต์นี้ช่วยส่งเสริมการสื่อสารระหว่างบุคคลทั้งในรูปแบบข้อความ รูปภาพ และไฟล์ PDF ทำให้การสื่อสารมีความหลากหลายและเข้าใจได้มากขึ้น
- ผู้ใช้สามารถแบ่งปันข้อมูลที่น่าสนใจ ความรู้ และประสบการณ์ต่าง ๆ กับผู้อื่นในชุมชน (ในเว็บไซต์นี้)
- เก็บไฟล์หรือรูปภาพไว้ดูภายหลังได้

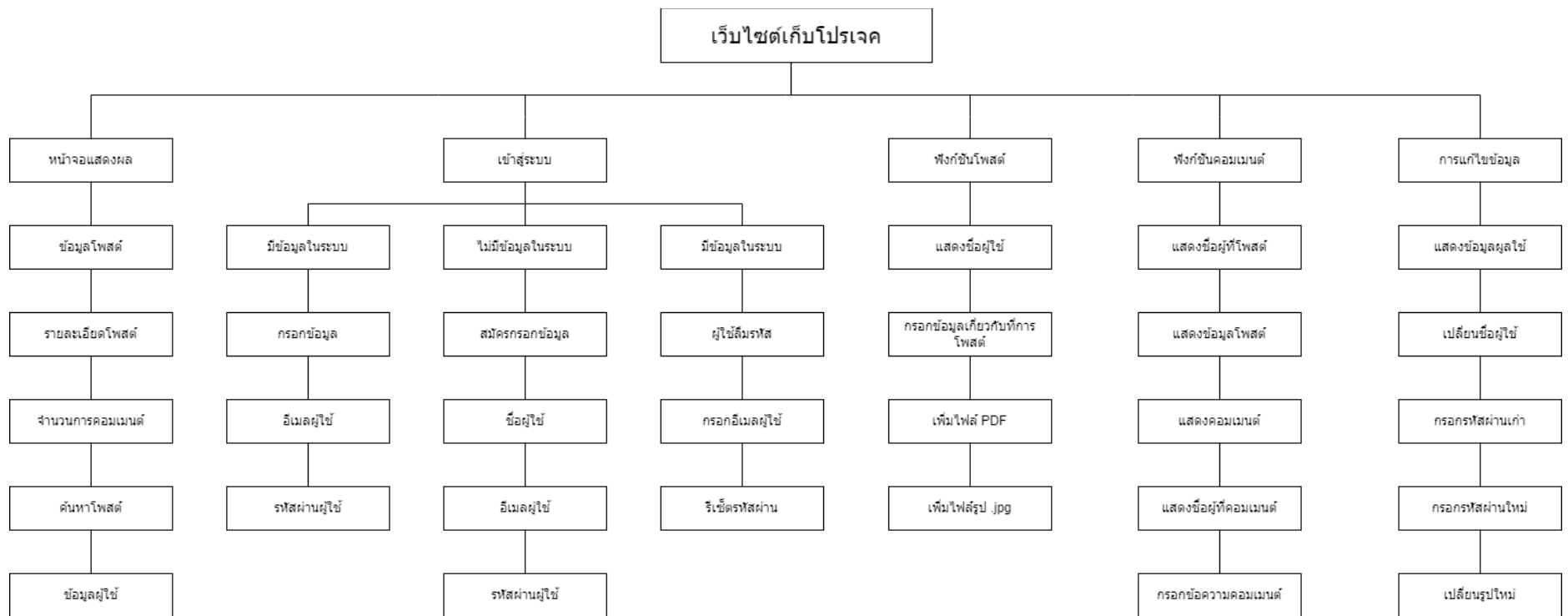
4. เทคโนโลยีที่ใช้

- HTML
- CSS
- JavaScript
- Node.js
- MySQL

บทที่ 2

โครงสร้างเว็บไซต์

แสดงข้อมูลโครงสร้างเว็บไซต์ ด้วย Tree Diagram

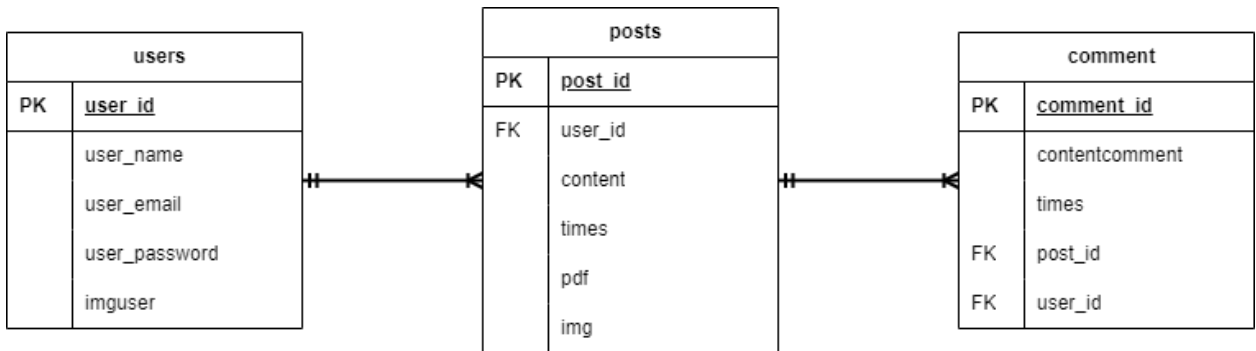


รูปแสดง Tree Diagram ของเว็บไซต์เก็บโปรเจก

บทที่ 3

ข้อมูลฐานข้อมูล

แสดงข้อมูล Database Schema หรือ ER Diagram



รูปแสดง ER Diagram ของเว็บไซต์เก็บโปรเจค

บทที่ 4

การติดตั้งและใช้งานเว็บไซต์

1. การติดตั้ง

ทำการ clone git repository โดยการใช้คำสั่งใน Terminal ดังนี้

- git clone https://github.com/POXKY/Web.git
- cd เข้าไปใน Folder Web
- npm install
- npm run dev

2. การเรียกใช้งาน

2.1 หน้าบ้าน

2.1.1 หน้า home html css

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>My Profile</title>
7   {{! <link rel="stylesheet" href="..."> }}
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgp3LIn9Nao0Yv1ztcQTwfspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
9   <style>
10     {{>style}}
11   {{>bootstrap}}
12 </style>
13 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css" integrity="sha512-DTOQ99RWCH3ppGqChAE1B1ZOC6xxalwEsW9c2Q0eAIf1+VegovIneeIC9QX4TctnMh13T7ye+gI1m8e2LW" crossorigin="anonymous">
14 </head>
15 <body class="bg-dark">
16
17   {{>menuSetting}}
18
19   <div class="container-sm" style="height: 100%; padding-top: 14px; justify-content: center; margin-top: 50px;">
20     {{>set}}
21   </div>
22
23   {{>footer}}
24   <script>
25     var settingsmenu = document.querySelector(".settings-menu");
26     var darkBtn = document.getElementById("dark-btn");
27     function settingMenuToggle(){
28       settingsmenu.classList.toggle("settings-menu-height")
29     }
30
31
32     darkBtn.onclick = function(){
33       darkBtn.classList.toggle("dark-btn-on");
34       document.body.classList.toggle("dark-theme");
35     }
36   </script>
37 </body>
38 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Home</title>
7   <link rel="stylesheet" href="...">
8   <link rel="stylesheet" href="...">
9   <script src="bootstrap.min.js"></script>
10  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css" integrity="sha512-DTOQ99RWCH3ppGqChAE1B1ZOC6xxalwEsW9c2Q0eAIf1+VegovIneeIC9QX4TctnMh13T7ye+gI1m8e2LW" crossorigin="anonymous">
11  <link href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" rel="stylesheet">
12  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" integrity="sha512-x79X99Z1+Z7J1N4M2zpwgStVF73dkbJ5qtkpxjC08thXmM1ArRmAYhY8LMVzjzRRQUM4KbV3ka57RNCwJvTTF" crossorigin="anonymous">
13 </head>
14 <style>
15   .preview-images img {
16     width: 100%
17   }
18 </style>
19 <body class="bg-dark">
20   {{>menu}}
21
22   <div class="container-sm" style="height: 100%; padding-top: 14px; justify-content: center; margin-top: 50px;">
23     {{! <div class="row"> }}
24     {{! <div class="col"> }}
25     {{! <div class="col"> }}
26     {{! </div> }}
27   </div>
28
29   {{>footer}}
30
31   {{>jquery}}
32   <script src="https://cdnjs.cloudflare.com/ajax/libs/mustache.js/4.2.0/mustache.min.js"></script>
33   <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
34   <script>
35     var settingsmenu = document.querySelector(".settings-menu");
36     var darkBtn = document.getElementById("dark-btn");
37     function settingMenuToggle(){
38       settingsmenu.classList.toggle("settings-menu-height")
39     }

```

```

40   {{>footer}}
41
42   {{>jquery}}
43   <script src="https://cdnjs.cloudflare.com/ajax/libs/mustache.js/4.2.0/mustache.min.js"></script>
44   <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
45   <script>
46     var settingsmenu = document.querySelector(".settings-menu");
47     var darkBtn = document.getElementById("dark-btn");
48     function settingMenuToggle(){
49       settingsmenu.classList.toggle("settings-menu-height")
50     }
51
52     darkBtn.onclick = function(){
53       darkBtn.classList.toggle("dark-btn-on");
54       document.body.classList.toggle("dark-theme");
55     }
56
57     function submitPost() {
58       var pdfFileInput = document.getElementById('pdf-upload');
59       var jpgFileInput = document.getElementById('jpg-upload');
60       if (pdfFileInput.files.length > 0 || jpgFileInput.files.length > 0) {
61         console.log("Post submitted successfully!");
62       } else {
63         alert('Please select at least one file to post.');

```

2.1.2 หน้า login html css

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <link rel="stylesheet" href="../../css/login.css">
  <link rel="stylesheet" href="../../css/bootstrap.min.css">
  <link href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Kanit:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Prompt:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Kanit:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900" rel="stylesheet">
</head>
<body class="bg-dark">
  <section class="container">
    <div class="form-login">
      <div class="content">
        <header>Login</header>
        <form action="/member/verify" method="post">
          <div class="field input-field">
            <input type="email" placeholder="Email" class="input" name="user_email" id="user_email" required>
          </div>
          <div class="field input-field">
            <input type="password" placeholder="Password" class="password" name="user_password" id="user_password" required>
            <i class="bx bx-hide eye-icon" style="font-size: 18px;"></i>
          </div>
          <div class="form-link">
            <a href="/reset/reset" class="forgot-pass">Forgot password</a>
          </div>
          <div class="mb-3 center">
            {msg}
          </div>
          <div class="field button-field">
            <button type="submit">Login</button>
          </div>
          <div class="form-link">
            <span>Already have an account? <a href="/register/register" class="signup-link">Signup</a></span>
          </div>
        </form>
      </div>
    </div>
  </section>
</body>
</html>

```

2.1.3 หน้า resetpass.html ใช้ css เหมือนกันกับ login

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <link rel="stylesheet" href="../css/login.css">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <script src="js/bootstrap.min.js"></script>
  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
</head>
<body class="bg-dark">
  <section class="container">
    <div class="form-login">
      <div class="content">
        <header>Reset Password</header>
        <form action="/reset/resetpass" method="post">
          <div class="field input-field">
            <input type="email" placeholder="Email" class="input" name="user_email" id="user_email" required>
          </div>
          <div class="field input-field">
            <input type="password" placeholder="New password" class="password" name="user_password" id="user_password" required>
            <i class='bx bx-hide eye-icon' style="font-size: 18px;"></i>
          </div>
          <div class="field button-field">
            <button type="submit">Reset</button>
          </div>
        </form>
      </div>
    </div>
  </section>
</body>
</html>
```

2.1.3 หน้า sign-up html css

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <link rel="stylesheet" href="...">
  <link rel="stylesheet" href="...">
  <link href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Kanit:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Prompt:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900" rel="stylesheet">
</head>
<body class="bg-dark">
  <section class="container">
    <div class="form-login">
      <div class="content">
        <header>Sign-up</header>
        <form action="/register/re" method="post">
          <div class="field input-field">
            <input type="text" placeholder="Username" class="input" name="user_name" id="user_name" required>
          </div>
          <div class="field input-field">
            <input type="email" placeholder="Email" class="input" name="user_email" id="user_email" required>
          </div>
          <div class="field input-field">
            <input type="password" placeholder="Password" class="password" name="user_password" id="user_password" required>
            <i class="bx bx-hide eye-icon" style="font-size: 18px;"></i>
          </div>
          <div class="field button-field">
            <button type="submit">Signup</button>
          </div>
          <div class="form-link">
            <span>Already have an account? <a href="/member/login" class="signup-link">Login</a></span>
          </div>
        </form>
      </div>
    </div>
  </section>
</body>
</html>
```

2.1.4 ปุ่มกดโพสต์ html css

```
1  {{#user_name}}
2  <div class="write-post-container">
3    <div class="user-profile">
4      
5      <div>
6        <p>{{user_name}}</p>
7      </div>
8    </div>
9    <form action="/member/verify/upload" method="post" enctype="multipart/form-data">
10     <div class="post-input-container">
11       <textarea rows="3" placeholder="What's on your mind?" name="content" id="content"></textarea>
12     </div>
13     <div class="upload">
14       <h1> pdf <i class="fas fa-file-pdf"></i></h1>
15       <input type="file" name="file" id="pdf-upload" accept=".pdf">
16     </div>
17     <div class="upload">
18       <h1> jpg <i class="fas fa-file-image"></i></h1>
19       <input type="file" name="file" id="input-files" accept=".jpg, .jpeg">
20     </div>
21     <div class="post">
22       <button type="submit">โพสต์</button>{{msg}}
23     </div>
24     <div class="col-sm-12">
25       <div class="preview-images">
26       </div>
27     </div>
28   </form>
29   {{/user_name}}
```


2.1.5 เมนู html css

```
1  {{#data}}
2  <nav class="navbar navbar-expand-sm navbar-dark fixed-top" id="allnavbar" style="background: var(--nav-color);">
3    <div class="container-fluid">
4      <a href="/member/verify" class="navbar-brand" style="padding-left: 20px; color: var(--text-color);">Group 7</a>
5      <form action="/member/verify" style="display: flex; align-items: center;">
6
7        <!-- Profile -->
8        <div class="nav-user-icon nav-item" style="display: flex; align-items: center; margin-left: 20px; " onclick="settingMenuToggle()">
9          
10        </div>
11      </form>
12      <!-- Setting -->
13      <div class="settings-menu">
14
15        <div id="dark-btn">
16          <span></span>
17        </div>
18        <div class="settings-menu-inner">
19          <div class="user-profile">
20            
21            <div>
22              <p>{{user_name}}</p>
23              <a href="/member/verify/myposts">See your profile</a>
24            </div>
25          </div>
26          <hr>
27          <div class="setting-links">
28            
29            <a href="/member/logout">Logout </a>
30          </div>
31          <!-- Setting -->
32        </div>
33      </div>
34      <!-- Profile -->
35    </div>
36    <!-- MENU -->
37  </nav>
38  {{/data}}
```

2.2 หลังบ้าน

```
const mysql = require('mysql2');
const dbConnection = require('./database')
const getDataPost = (req,res,next) =>{
  const sqlposts = 'SELECT users.user_name ,posts.content,posts.times,posts.img,posts.pdf,posts.post_id FROM posts INNER JOIN users ON users.user_id = posts.user_id ORDER BY posts.post_id DESC;'
  dbConnection.query(sqlposts,(error,result,fields)->{
    if(error){
      console.error(error);
    }
    res.locals.data = result ;
    //postid = result[0].post_id;
    // for(let i=0;i<result.length;i++){
    //   postid = result[i].post_id;
    //   console.log(postid);
    // }

    next();
  });
}
module.exports = getDataPost;
```

```

1  const mysql = require('mysql2');
2  const dbConection = mysql.createPool({
3      host: 'localhost', //process.env.DB_HOST,
4      user: 'root', //process.env.DB_USER,
5      password: '', //process.env.DB_PASSWORD,
6      database: 'project', //process.env.DB_NAME,
7      port: 3306,
8  });
9
10 module.exports = dbConection;

```

```

web > index.js / ...
1  const express = require('express');
2  const path = require('path');
3  const bodyParser = require('body-parser');
4  const mustacheExpress = require('mustache-express')
5  const app = express();
6  const PORT = 3000;
7  const getDatapost = require('./accout')
8
9
10
11 // Mustache
12 app.set('views', `${__dirname}/static`);
13 app.set('view engine', 'mustache');
14 app.engine('mustache', mustacheExpress());
15 app.use(express.static('static/css'));
16
17 // Config database
18 require('dotenv').config();
19
20 app.use(bodyParser.urlencoded({ extended: true }));
21 app.use(bodyParser.json());
22
23 let root_path = path.resolve(__dirname, 'static')
24 app.use(express.static(root_path));
25 app.use(express.static(path.join(__dirname, 'public')))
26
27 // router
28 const registerRouters = require('./router/register');
29 const memberRouters = require('./router/member');
30 const resetpassRouters = require('./router/resetpass');
31
32
33
34
35
36 // use router
37 app.use('/register', registerRouters);
38 app.use('/member', memberRouters);
39 app.use('/reset', resetpassRouters);
40 app.use(getDatapost);

```

```
41
42 // Home
43 app.get('/', (req, res) => {
44   res.render('home/home', {posts: res.locals.data})
45 });
46
47
48 // Test
49 app.get('/test', (req, res) => {
50   res.setHeader('Content', 'text/html')
51   res.send('<h1>Test</h1>')
52 });
53
54
55 app.listen(PORT, () => {
56   console.log(`http://localhost:${PORT}`);
57 });
```

```

1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const router = express.Router();
4  const dbConection = require('../database');
5  const cookieSession = require('cookie-session');
6  const {body, validationResult} = require('express-validator');
7  const cookie = require('cookie-parser');
8  const md5 = require('md5');
9  const getDataPost = require('../accout');
10 const uploadFile = require('../middleware/upload')
11
12 router.use(cookie());
13 router.use(getDataPost);
14
15 router.use(bodyParser.urlencoded({ extended: true}));
16 router.use(bodyParser.json());
17
18 router.use(express.urlencoded({extended: true}));
19 router.use(express.json({extended: true}));
20
21
22
23 // router.use(getUsers);
24
25 // Login.mustache
26 router.get('/login',(req,res)=>{
27   res.render('member/login')
28 });
29
30
31 let gloResults = null;
32 let userid = null;
33 let username = null;
34 // member/login
35 router.post('/verify',(req,res)=>{
36   const {user_email,user_password} = req.body
37   // const user = req.cookie.user_id;
38   const sqluser = 'SELECT * FROM users WHERE user_email = ? and user_password = ? '
39   dbConection.query(sqluser,[user_email,md5(user_password)],(err,result)=>{
40     if(err){

```

```

    console.log(err)
    res.render('member/login')
  }else{
    if(results.length == 0){
      res.render('member/login',{msg: '!!!Wrong Username or Password!!!'})
    }else{
      res.cookie('user_email',user_email,{maxAge: 600000})
      gloResults = results;
      userid = results[0].user_id;
      username = results[0].user_name;
      res.render('member/home',{data: gloResults,posts: res.locals.data})
    }
  }
});
});

```

```

// click home
router.get('/verify',(req,res)=>{
  const username = req.cookies.user_email;
  if (username){
    res.render('member/home',{user_name:username,posts: res.locals.data})
  }
  else {
    res.redirect('/member/login')
  }
});

```

```

74 // Logout
75 router.get('/logout',(req,res)=>{
76     const username = req.cookies.user_email;
77     if (username){
78         res.clearCookie('username')
79     }
80     res.redirect('/')
81 });
82
83
84
85 // Upload Post
86
87
88 router.post('/verify/upload',uploadFile.array('file',10),(req,res)=>{
89     // if(!req.file){
90     //     return res.status(400).send({msg:'No file uploaded.'});
91     // }
92     const {content} = req.body
93     const sqlinto = 'INSERT INTO posts (post_id,user_id,content,times,pdf,img) VALUES (?,?,,?,?,?)';
94     dbConnection.query(sqlinto,['',userid , content ,'',req.files[0].filename,req.files[1].filename],(error,result)=>{
95         if(error){
96             return res.status(500).json({error: error.message});
97         }else{
98             res.redirect('/member/verify')
99         }
100     });
101
102
103 });
104
105
106

```

```

7 // Setting
8
9
10 router.get('/verify/setting',(req,res)=>{
11     const sqlset = 'SELECT * FROM users WHERE user_id = '+userid
12     dbConnection.query(sqlset,(error,result,fields) => {
13         if(error)
14             console.error(error);
15         res.render('member/setting',{datas: result,data: gloResults});
16     });
17
18 });
19
20
21
22
23
24 router.get('/verify/myposts',(req,res)=>{
25
26     const sqlmyposts = 'SELECT users.user_name ,posts.content,posts.times,posts.img,posts.pdf FROM posts INNER JOIN users ON users.user_id = posts.user_id WHERE posts.user_id ='+userid
27     dbConnection.query(sqlmyposts,(error,result,fields) => {
28         if(error)
29             console.error(error);
30         res.render('member/mypost',{mypost: result,data: gloResults,dattax: userid});
31     });
32
33 });
34
35
36
37 router.get('/test',(req,res)=>{
38     res.render('test',{data: gloResults})
39     // console.log(userid);
40     // console.log(gloResults);
41     // console.log(username);
42     // gloResults.forEach(result => {
43     //     console.log(result);
44     // });
45 })
46
47

```

```

77 // Setting
78
79
80 router.get('/verify/setting',(req,res)=>{
81     const sqlset = 'SELECT * FROM users WHERE user_id = '+userid
82     dbConnection.query(sqlset,(error,results,fields) => {
83         if(error)
84             console.error(error);
85         res.render('member/setting',{datas: results,data: gloResults});
86     });
87 });
88
89
90
91
92
93
94 router.get('/verify/myposts',(req,res)=>{
95
96     const sqlmyposts = 'SELECT users.user_name ,posts.content,posts.times,posts.img,posts.pdf FROM posts INNER JOIN users ON users.user_id = posts.user_id WHERE posts.user_id ='+userid
97     dbConnection.query(sqlmyposts,(error,results,fields) => {
98         if(error)
99             console.error(error);
100         res.render('member/mypost',{mypost: results,data: gloResults,dass: userid});
101     });
102 });
103
104
105 router.get('/test',(req,res)=>{
106     res.render('test',{data: gloResults})
107     // console.log(userid);
108     // console.log(gloResults);
109     // console.log(username);
110     // gloResults.forEach(result => {
111     //     console.log(result);
112     // });
113 })
114
115 module.exports = router;

```

Web > router > JS register.js > ...

```

1  const express = require('express');
2  const router = express.Router();
3  const bodyParser = require('body-parser');
4  const dbConnection = require('../database')
5  const cookieSession = require('cookie-session');
6  const {body,validationResult} = require('express-validator');
7  const md5 = require('md5');
8
9
10 router.use(express.urlencoded({ extended: false}));
11
12 router.use(bodyParser.urlencoded({ extended: true}));
13 router.use(bodyParser.json());
14
15 router.get('/register',(req,res)=>{
16     res.render('register/register')
17 });
18
19
20
21 router.post('/re',(req,res)=>{
22
23     const { user_name } =req.body;
24     const { user_email } =req.body;
25     const { user_password } =req.body;
26     const sql = 'INSERT INTO users(user_id,user_name,user_email,user_password ) VALUES (?,?,,?)';
27     dbConnection.query(sql,['',user_name,user_email,md5(user_password)],(err,results)=>{
28         if(err){
29             return res.status(500).json({error: err.message});
30         }else{
31             res.render('member/login')
32         }
33     });
34 });
35
36 module.exports = router;

```

Web > router > JS resetpass.js > ...

```
1  const express = require('express');
2  const router = express.Router();
3  const dbConection = require('../database')
4  const md5 = require('md5');
5
6  router.get('/reset',(req,res)=>{
7    res.render('home/resetpass');
8  })
9
10 router.post('/resetpass',(req,res)=>{
11   const {user_email , user_password} = req.body;
12   const sqlreset = 'UPDATE users SET user_password = ? WHERE user_email = ?'
13   dbConection.query(sqlreset,[md5(user_password),user_email],(err,results)=>{
14     if(err){
15       return res.status(500).json({error: err.message})
16     }else{
17       // console.log(results)
18       res.render('member/login')
19     }
20   })
21 })
22
23 module.exports = router;
```


Web > middleware > JS upload.js > ...

```
1  const multer = require('multer')
2  const path = require('path');
3
4
5  const storage = multer.diskStorage({
6    destination: (req, file, cb)=>{
7      cb(null, './public/img')
8    },
9    filename: (req, file, cb)=>{
10      cb(null, Date.now() + '-' + file.originalname)
11    }
12  })
13  // const storage = multer.memoryStorage();
14  let uploadFile = multer({
15    storage: storage,
16    limits: { fileSize: 1024 * 1024 * 10 }
17  });
18
19  module.exports = uploadFile;
```