

รายงานประกอบรายวิชา
เทคโนโลยีการเขียนโปรแกรมเว็บไซต์
Web Programming Technology

รหัสวิชา 226297

เสนอ เว็บแอปพลิเคชัน

Groovy Design

ผู้จัดทำ

65025761	นาย ภาณุ	เสมอสูง
65021789	นาย ธีรภัทร์	หงส์ทอง
65025648	นาย ก่อพจน์	จอมแก้ว
65024692	นาย ภคภพ	เขียวขำ
65021352	นางสาว กัลยรักษ์	โรจนเลิศประเสริฐ

ผู้สอน

อาจารย์ วัฒนพงศ์ สุทธภักดิ์

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร

มหาวิทยาลัยพะเยา

ภาคการศึกษา ปลาย ประจำปีการศึกษา 2566

บทที่ 1

ที่มาของเว็บไซต์

1. ที่มาและความสำคัญ

เทคโนโลยีมีประโยชน์มากมายต่อการดำรงชีวิต ปัจจุบันประชาชนมีความสนใจกับการตกแต่งบ้านมากขึ้นตามสไตล์ที่แตกต่างกันออกไป แต่ยังคงขาดไอเดียการตกแต่งบ้านที่ทันสมัย และสไตล์การตกแต่งบ้านต่างๆไม่อยู่ในเว็บไซต์เดียวกัน ทำให้ยุ่งยากต่อการใช้งาน

จากปัญหาข้างต้นคณะผู้จัดทำจึงมีแนวคิดที่จะมีการพัฒนาเว็บไซต์ให้ดีขึ้น โดยมีการนำเทคโนโลยีเข้ามาช่วยในการแก้ไขปัญหาต่างๆ

2. วัตถุประสงค์

1. เพื่อเป็นพื้นที่ที่ช่วยในการแลกเปลี่ยนไอเดียและเคล็ดลับในการตกแต่งบ้าน
2. เพื่อเป็นแหล่งรวบรวมไอเดียตกแต่งบ้านหลายๆมุมมอง
3. เพื่อสร้างแรงบันดาลใจในการตกแต่งบ้าน

3. ประโยชน์ที่ได้รับ

จากโครงการที่ได้ดำเนินการ ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่าเว็บไซต์ Groovy Design ที่ทางคณะผู้จัดทำได้ทำขึ้นมาจะสามารถเป็นประโยชน์ให้แก่ผู้ที่ได้ใช้งานทำให้การเลือกดูไอเดียการตกแต่งบ้านที่น่าสนใจและบนเว็บไซต์นี้จะช่วยปรับแต่งสมาธิและอารมณ์ของผู้ใช้ให้ดีขึ้นเนื่องจากมีความสบายตา และยังสามารถค้นหาสไตล์การตกแต่งบ้านที่เหมาะสมได้

4. เทคโนโลยีที่ใช้

หลังบ้าน

1. express เป็นเฟรมเวิร์กที่ช่วยในการสร้างและจัดการเราเอพีไอ (RESTful API) และเว็บแอปพลิเคชัน Node.js

2. cookie-parser ใช้ในการอ่านคุกกี้จาก HTTP request ในแอปพลิเคชัน Express
3. body-parser ช่วยในการแปลงข้อมูลที่ส่งมาในรูปแบบของ JSON, URL encoded หรือ multipart form-data เป็น object ใน Express
4. dotenv ช่วยในการโหลดตัวแปรสภาพแวดล้อมจากไฟล์ .env ใน Node.js application
5. multer ใช้ในการจัดการการอัปโหลดไฟล์ multipart/form-data ใน Express
6. cors ช่วยในการจัดการกับ Cross-Origin Resource Sharing (CORS) ใน Express application เพื่ออนุญาตการร้องขอที่มาจากโดเมนอื่น
7. path ใช้ในการจัดการและสร้างเส้นทางไฟล์และไดเรกทอรีใน Node.js
8. bcrypt ใช้ในการเข้ารหัสและตรวจสอบรหัสผ่านใน Node.js โดยใช้การเข้ารหัส bcrypt
9. jsonwebtoken ใช้ในการสร้างและตรวจสอบ JSON Web Tokens (JWT) ใน Node.js เพื่อการยืนยันตัวตนแบบ stateless

หน้าบ้าน

1. ant-design เป็นไลบรารี UI สำหรับ React ที่มีคอมโพเนนต์และสไตล์ที่สวยงามและสมบูรณ์
2. ant-design/icons ไอคอนเพิ่มเติมสำหรับ Ant Design ที่สามารถนำมาใช้ในโปรเจกต์ของคุณ
3. axios ไลบรารีที่ใช้สำหรับการจัดการ HTTP request ในแอปพลิเคชัน React
4. @mui/icons-material ไอคอน Material Design สำหรับ Material-UI
5. @mui/material เฟรมเวิร์ก UI สำหรับ React ที่ใช้งานง่ายและมีความสามารถมากมาย
6. @mui/styled-engine-sc เอ็นจินที่ใช้สำหรับการจัดการสไตล์ใน Material-UI โดยใช้ Styled Components
7. moment ไลบรารีสำหรับการจัดการกับเวลาและวันที่ใน JavaScript

8. react เฟรมเวิร์ก JavaScript สำหรับสร้าง UI โดยใช้คอมโพเนนต์แบบเรียลไทม์

9. react-dom ไลบรารีที่ใช้สำหรับการจัดการ DOM ใน React

10. react-quill คอมโพเนนต์ React สำหรับการแสดงและจัดการเนื้อหาในรูปแบบ Rich

Text

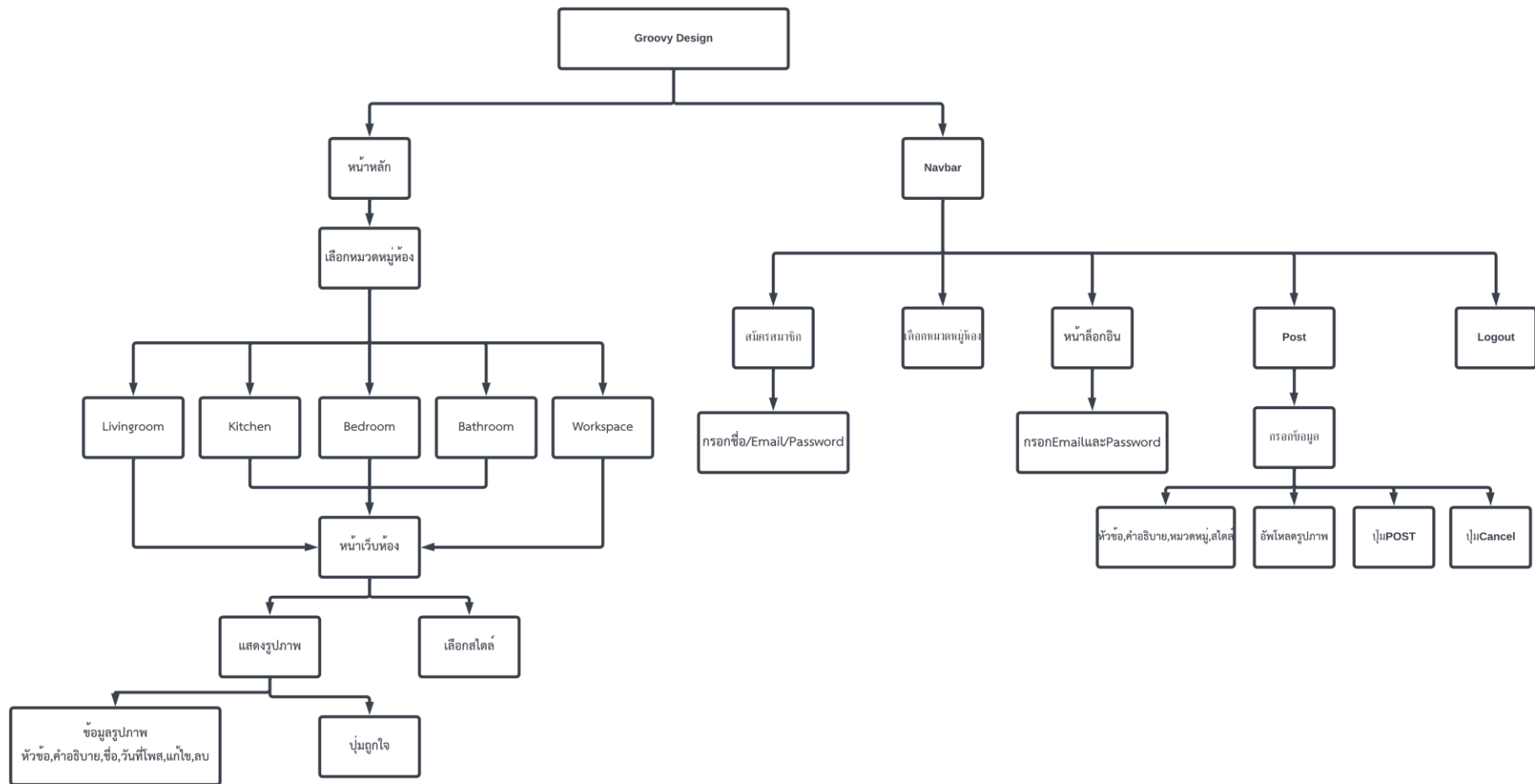
11. react-router-dom ไลบรารีสำหรับการจัดการเส้นทางในแอปพลิเคชัน React

12. fortawesome/free-solid-svg-icons ไอคอนฟรีที่มีคุณภาพจาก Font Awesome
สำหรับใช้ในโปรเจก React

13. fortawesome/react-fontawesome ไลบรารีสำหรับใช้ไอคอน Font Awesome ใน
แอปพลิเคชัน React

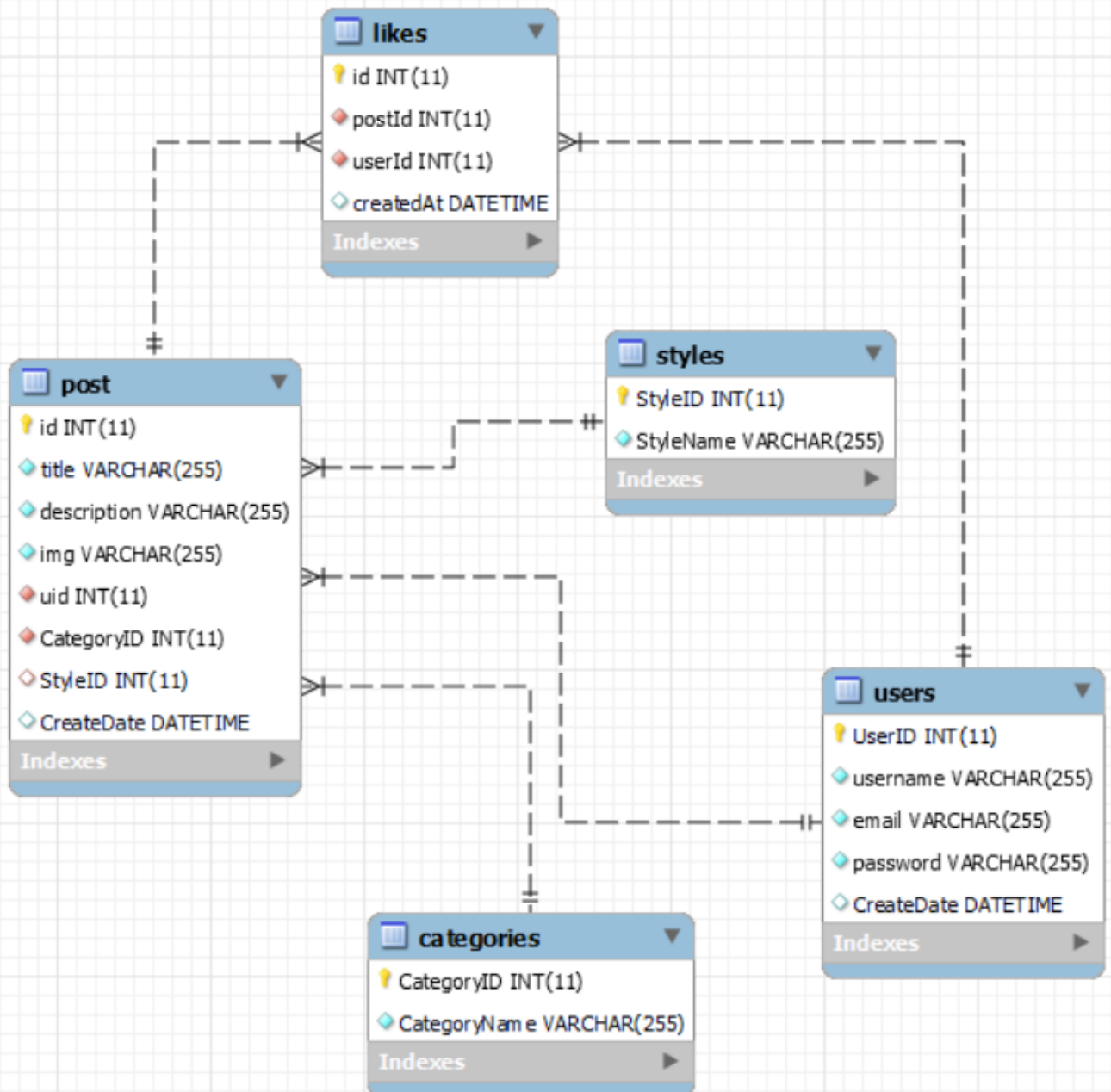
บทที่ 2

โครงสร้างเว็บไซต์



บทที่ 3

ข้อมูลฐานข้อมูล



บทที่ 4

การติดตั้งและการใช้งานเว็บไซต์

1. การติดตั้ง

1.1 ทำการ clone git repository โดยการใช้คำสั่งใน Terminal ดังนี้

1.1.1 git init

1.1.2 git add.

1.1.3 git commit -m “first commit”

1.1.4 git branch -M main

1.1.5 git remote add origin http://github.com/HoxZi/kananek_web

1.1.6 git push -u origin main

1.2 ทำการสร้าง file .gitignore แล้วทำการคอมพิ์ /node_modules เพื่อไม่ให้ส่งไฟล์ node_modules github แล้วใช้ คำสั่งต่อไปนี้ ใน Terminal

1.2.1 git add .

1.2.2 git commit -m “ข้อความที่ต้องการใส่”

1.2.3 git push -u origin main

1.3เปิดไฟล์Frontendในโปรแกรม Visual studio Code พิมพ์ ในterminal: npm install จากนั้นพิมพ์npm startเพื่อเปิดserver

และเปิดไฟล์Backend ทำตามขั้นตอนเหมือนกับFrontend

1.4 ดาวน์โหลดไฟล์ SQL จาก github

1.4.1สร้างฐานข้อมูลชื่อว่า logindata

1.4.2จากนั้นเข้าในส่วนQuery

1.4.3คลิกขวาและเลือก Load file SQL

1.4.4เลือกฐานข้อมูล

1.4.5กดrun

2. การเรียกใช้งาน

2.1. หลังบ้าน

Index.js จะเป็นการเรียกใช้ Library และเปิดServer ในport 8000

มีAPI สำหรับUpload ไฟล์รูปภาพโดยใช้ Multer

```
> node_modules
├── route
│   ├── getroom
│   │   ├── room.js
│   │   ├── post.js
│   │   ├── user.js
│   │   ├── .env
│   │   ├── db.js
│   │   └── index.js
│   ├── package-lock.json
│   └── package.json
└── package-lock.json
└── package.json

1  const express = require('express');
2  const cookieParser = require('cookie-parser');
3  const bodyParser = require('body-parser');
4  const userRoutes = require('./route/user');
5  const postRoutes = require('./route/post');
6  const roomRoutes = require('./route/getroom/room');
7  const dotenv = require('dotenv').config();
8  const app = express();
9  const port = 8000;
10 const multer = require('multer');
11 const cors = require('cors');
12 const path = require('path');
13 app.use(bodyParser.json());
14 app.use(bodyParser.urlencoded({ extended: true }));
15 app.use(cookieParser());
16 app.use(express.json());
17 app.use(express.static('public'));
18
19
20 app.use(cors({
21   origin: ['http://localhost:3000'],
22   methods: ['GET', 'POST', 'PUT', 'DELETE'],
23   credentials: true
24 }));
25
26
27
28
29 const storage = multer.diskStorage({
30   destination: function (req, file, cb) {
31     cb(null, '../Frontend/public/upload/');
32   },
33   filename: function (req, file, cb) {
34     cb(null, Date.now()+file.originalname);
35   },
36 });
37
38 const upload = multer({ storage });
39 app.use('/uploads', express.static('../uploads'));
40
41 app.post('/api/upload', upload.single('file'), function (req, res) {
42   const file = req.file;
43   res.status(200).json(file.filename);
44   console.log(file.filename);
45 });
46
47
48
49 app.use('/api/users', userRoutes);
50 app.use('/api/post', postRoutes);
51 app.use('/api/rooms', roomRoutes);
52
53
54 app.listen(port, () => {
55   console.log(`Server running on port ${port}`);
56 });
```

ในไฟล์db.js เป็นการเชื่อมกับฐานข้อมูลโดยมีการกำหนดตัวแปรใน .env

```
JS index.js JS db.js ×
JS db.js > ...
1  dotenv = require('dotenv').config();
2  const mysql = require('mysql');
3
4  const db = mysql.createConnection({
5    host: process.env.DB_HOST,
6    user: process.env.DB_USER,
7    database: process.env.DB_DATABASE
8  });
9  module.exports = db;
```


user.js เป็นที่เก็บAPI สำหรับสมัครสมาชิกและเข้าสู่ระบบ

```
PROJECT
├── node_modules
├── route
├── getroom
├── roomjs
├── postjs
├── userjs
├── .env
├── dbjs
├── indexjs
├── package-lock.json
└── package.json

route > JS userjs > ...
1  const express = require('express');
2  const bcrypt = require('bcrypt');
3  const bodyParser = require('body-parser');
4  const jwt = require('jsonwebtoken');
5  const app = express();
6  const router = express.Router();
7
8  app.use(express.json());
9  // Database connection
10 const db = require('../db');
11 db.connect((err) => {
12   if (err) throw err;
13   console.log('Connected to database');
14 });
15 const jwtSecret = process.env.JWT_SECRET;
16 app.use(bodyParser.urlencoded({ extended: true }));
17
18 // Route สำหรับการลงทะเบียน
19 router.post('/register', async (req, res) => {
20   const { username, email, password } = req.body;
21   if (!username || !email || !password) {
22     return res.status(400).json({ msg: 'Please enter all fields' });
23   }
24
25   // ตรวจสอบอีเมลซ้ำ
26   db.query('SELECT * FROM users WHERE email = ?', [email], async (err, result) => {
27     if (err) throw err;
28     if (result.length > 0) {
29       return res.status(400).json({ msg: 'Email already exists' });
30     } else {
31       // Hash password
32       const salt = await bcrypt.genSalt(10);
33       const hashedPassword = await bcrypt.hash(password, salt);
34
35       // บันทึกข้อมูลลงในฐานข้อมูล
36       const newUser = { Username: username, Email: email, Password: hashedPassword };
37       db.query('INSERT INTO users SET ?', newUser, (err, result) => {
38         if (err) throw err;
39         res.status(201).json({ msg: 'registered successfully' });
40       });
41     }
42   });
43 });
```

```
JS postjs
JS userjs
.env
dbjs
indexjs
package-lock.json
package.json

40   });
41   }
42   });
43   });
44   router.post('/login', (req, res) => {
45     const q = "SELECT * FROM users WHERE email = ?";
46     db.query(q, [req.body.email], (err, data) => {
47       if (err) return res.status(500).json(err);
48       if (data.length === 0) return res.status(404).json("User not found!");
49       const isPasswordCorrect = bcrypt.compareSync(req.body.password, data[0].password);
50       if (!isPasswordCorrect) return res.status(400).json("Wrong username or password!");
51       const token = jwt.sign({ id: data[0].id }, jwtSecret);
52       const { password, ...other } = data[0];
53       res.status(200).json({ ...other, token });
54     });
55   });
56
57
58   router.post('/logout', (req, res) => {
59     res.status(200).json({ message: "Logged out successfully" });
60   });
61
62
63   module.exports = router;
64
65
66
```

```
PROJECT
├── node_modules
├── route
│   ├── getroom
│   ├── JS room.js
│   ├── JS post.js
│   ├── JS user.js
│   ├── .env
│   ├── JS db.js
│   ├── JS index.js
│   ├── package-lock.json
│   └── package.json
└── .env

route > JS post.js > [0] authenticateToken
1  const express = require('express');
2  const router = express.Router();
3  const db = require('../db');
4  const bodyParser = require('body-parser');
5  const app = express();
6  const jwt = require('jsonwebtoken');
7  const jwtSecret = process.env.JWT_SECRET;
8
9  app.use(express.json());
10 app.use(bodyParser.urlencoded({ extended: true }));
11 app.use(bodyParser.json());
12
13 //=====
14
15 const authenticateToken = (req, res, next) => {
16   const authHeader = req.headers['authorization'];
17   const token = authHeader && authHeader.split(' ')[1]; // "Bearer TOKEN"
18
19   if (token == null) return res.sendStatus(401);
20
21   jwt.verify(token, jwtSecret, (err, user) => {
22     if (err) return res.sendStatus(403);
23     req.user = user;
24     next();
25   });
26 };
27
28 //=====
29
30 router.post('/addPost', authenticateToken, (req, res) => {
31   const { title, description, img, CategoryID, StyleID, uid } = req.body;
32
33   const sql = `INSERT INTO post (title, description, img, CreateDate, uid, CategoryID, StyleID)
34   VALUES (?, ?, ?, NOW(), ?, ?, ?)`;
35
36   db.query(sql, [title, description, img, uid, CategoryID, StyleID], (err, result) => {
37     if (err) {
38       console.error(err);
39       return res.status(500).json({ error: 'Internal server error' });
40     }
41     res.status(201).json({ message: 'Post added successfully', postId: result.insertId });
42   });
43 });
44
```

editPost และ deletePost จะใช้แก้ไขโพสโดยอิงตามpostId

```
JS postjs
JS userjs
.env
JS dbjs
JS indexjs
() package-lockjson
() packagejson

45 router.put('/editPost/:id', authenticateToken, (req, res) => {
46   const { id } = req.params; // ID ของโพสต์ที่จะแก้ไข
47   const { title, description, img, CategoryID, StyleID } = req.body;
48
49   // คำสั่ง SQL สำหรับอัปเดตโพสต์
50   const sql = `UPDATE post SET title = ?, description = ?, img = ?, CategoryID = ?, StyleID = ? WHERE id = ?`;
51
52   // ประมวลผลคำสั่ง SQL พร้อมพารามิเตอร์จากตัวแปร req.body
53   db.query(sql, [title, description, img, CategoryID, StyleID, id], (err, result) => {
54     if (err) {
55       console.error(err);
56       return res.status(500).json({ error: 'Internal server error' });
57     }
58     if (result.affectedRows === 0) {
59       return res.status(404).json({ error: 'Post not found' });
60     }
61     res.json({ message: 'Post updated successfully' });
62   });
63 });
64
65
66
67 router.delete('/deletePost/:id', (req, res) => {
68   const { id } = req.params; // ID ของโพสต์ที่ต้องการลบ
69
70   const deleteSql = `DELETE FROM post WHERE id = ?`;
71   db.query(deleteSql, [id], (err, result) => {
72     if (err) {
73       console.error(err);
74       return res.status(500).json({ error: 'Internal server error' });
75     }
76     if (result.affectedRows === 0) {
77       // ไม่พบโพสต์ที่ต้องการลบ
78       return res.status(404).json({ message: 'Post not found' });
79     }
80     res.json({ message: 'Post deleted successfully' });
81   });
82 });
83
```

/LikeCount/:id จะทำการนับจำนวนการกดถูกใจของโพสต์นั้นๆ และ/Like:id จะเป็นการส่งข้อมูลเมื่อมีการกดถูกใจจะส่ง 1 ไปยังตาราง likes

```
JS dbjs
JS indexjs
() package-lockjson
() packagejson

86 router.get('/likesCount/:postId', (req, res) => {
87   const postId = req.params.postId;
88   const sql = `SELECT COUNT(*) AS likesCount FROM likes WHERE postId = ?`;
89
90   db.query(sql, [postId], (err, result) => {
91     if (err) {
92       console.error('Error fetching likes count:', err);
93       res.status(500).json({ error: 'Internal server error' });
94     } else {
95       const likesCount = result[0].likesCount;
96       res.status(200).json({ likesCount });
97     }
98   });
99 });
100
101 router.post('/like/:postId', authenticateToken, (req, res) => {
102   const { postId, userId } = req.body;
103
104   const sql = `INSERT INTO likes (postId, userId) VALUES (?, ?)`;
105
106   db.query(sql, [postId, userId], (err, result) => {
107     if (err) {
108       console.error('Error liking post:', err);
109       res.status(500).json({ error: 'Internal server error' });
110     } else {
111       res.status(201).json({ message: 'Post liked successfully' });
112     }
113   });
114 });
115
```

ส่วน getImage จะดึงข้อมูลสำหรับแสดงรูปภาพบนเว็บไซต์และรายละเอียดต่างๆ โดยเรียงลำดับจากมากไปน้อยตามจำนวนการกดถูกใจ

```
> node_modules
  route
    getroom
      JS room.js
      JS post.js
      JS user.js
    .env
    JS db.js
    JS index.js
    {} package-lock.json
    {} package.json

1  const express = require('express');
2  const router = express.Router();
3  const db = require('../db');
4  const bodyParser = require('body-parser');
5  const { route } = require('../post');
6  const app = express();
7
8  app.use(express.json());
9  app.use(bodyParser.urlencoded({ extended: true }));
10 app.use(bodyParser.json());
11
12
13
14 router.get('/categories', (req, res) => {
15   const sql = 'SELECT CategoryID, CategoryName FROM Categories';
16   db.query(sql, (err, results) => {
17     if (err) {
18       console.error('Error fetching categories:', err);
19       return res.status(500).json({ error: 'Internal server error' });
20     }
21     res.json(results);
22   });
23 });
24
25 router.get('/styles', (req, res) => {
26   const sql = 'SELECT StyleID, StyleName FROM Styles';
27   db.query(sql, (err, results) => {
28     if (err) {
29       console.error('Error fetching Styles:', err);
30       return res.status(500).json({ error: 'Internal server error' });
31     }
32     res.json(results);
33   });
34 });
35
```

```
> node_modules
  route
    getroom
      JS room.js
      JS post.js
      JS user.js
    .env
    JS db.js
    JS index.js
    {} package-lock.json
    {} package.json

33   });
34 });
35
36 router.get("/getImage/:id", (req, res) => {
37   const categoryId = req.params.id;
38   const sql = `
39     SELECT
40       p.id,
41       p.uid,
42       p.img,
43       p.title,
44       p.description,
45       p.CreateDate,
46       u.Username,
47       c.CategoryName,
48       s.StyleName,
49       COUNT(l.id) AS likesCount
50     FROM
51       post AS p
52     JOIN
53       Categories AS c ON p.CategoryID = c.CategoryID
54     JOIN
55       Styles AS s ON p.StyleID = s.StyleID
56     JOIN
57       users AS u ON p.uid = u.UserID
58     LEFT JOIN
59       likes AS l ON p.id = l.postId
60     WHERE
61       c.CategoryID = ?
62     GROUP BY
63       p.id
64     ORDER BY
65       likesCount DESC`;
66
67   db.query(sql, [categoryId], (err, result) => {
68     if (err) {
69       res.status(500).send('Error fetching posts for the specified CategoryID');
70     } else {
71       res.status(200).json(result);
72     }
73   });
74 });
75
```

2.2. หน้าบ้าน

สร้างโฟลเดอร์สำหรับห้องต่างๆ

```
> node_modules
> public
  src
    component
      login.jsx
      nav.jsx
      post.jsx
      register.jsx
    context
    css
    pages
      bathroom
      bedroom
      kitchen
      livingroom
      workspace
```

สร้าง authContext.JS สำหรับส่วนloginและlogout โดยเมื่อทำการ loginจะทำการเก็บ token,UserIDและข้อมูลuser ไว้ใน Local Storage และเมื่อทำการlog out จะเป็นการลบทุกอย่างที่อยู่ในLocal Storage

```
1 import axios from "axios";
2 import { createContext, useEffect, useState } from "react";
3
4 export const AuthContext = createContext();
5
6 export const AuthProvider = ({ children }) => {
7   const [currentUser, setCurrentUser] = useState(
8     JSON.parse(localStorage.getItem("user")) || null
9   );
10
11   const login = async (inputs) => {
12     const res = await axios.post("http://localhost:8000/api/users/login", inputs);
13
14     // ตรวจสอบก่อนว่า response มี data และ token หรือไม่
15     if (res.data && res.data.token) {
16       localStorage.setItem("token", res.data.token); // เก็บ token ไว้ใน local storage
17       localStorage.setItem("UserID", res.data.UserID);
18       const { token, ...userData } = res.data; // แยก token ออกจากข้อมูลผู้ใช้อื่นๆ
19       setCurrentUser(userData); // เก็บข้อมูลผู้ใช้ (ไม่รวม token) ไว้ใน state
20     }
21   };
22
23   const logout = async () => {
24     await axios.post("http://localhost:8000/api/users/logout");
25     localStorage.removeItem("token");
26     localStorage.removeItem("UserID");
27     localStorage.removeItem("user");
28     setCurrentUser(null);
29   };
30
31   useEffect(() => {
32     localStorage.setItem("user", JSON.stringify(currentUser));
33   }, [currentUser]);
34
35   return (
36     <AuthContext.Provider value={{ currentUser, login, logout }}>
37       {children}
38     </AuthContext.Provider>
39   );
40 };
41
```

Login.jsx จะเป็นส่วนหน้าเว็บไซต์ โดยจะให้ผู้ใช้กรอกemailและpassword หากสำเร็จจะพาผู้ใช้งานไปยังหน้า home

```
▼ FRONTEND
  > node_modules
  > public
  ▼ src
    ▼ component
      ✎ login.jsx
      ✎ nav.jsx
      ✎ post.jsx
      ✎ register.jsx
      ▼ context
        JS authContext.js
      > css
      ▼ pages
        > bathroom
        > bedroom
        > kitchen
        > livingroom
        > workspace
        ✎ checkbox.jsx
        ✎ home.jsx
src > component > ✎ login.jsx > ...
1  import React, { useContext, useState } from 'react';
2  import { Form, Input, Button, message } from 'antd';
3  import { AuthContext } from '../context/authContext';
4  import { useNavigate } from 'react-router-dom'
5  import Nav from './nav'
6
7  const Login = () => {
8    const { login } = useContext(AuthContext);
9    const [loading, setLoading] = useState(false);
10   const navigate = useNavigate();
11   const onFinish = async (values) => {
12     setLoading(true);
13     try {
14       await login(values);
15       message.success('Login successful!');
16       navigate('/home'); // นำทางไปยังหน้าหลัก
17       setLoading(false);
18     } catch (error) {
19       message.error('Failed to login!');
20       setLoading(false);
21     }
22   };
};
```

Register.jsx สำหรับหน้าต่างสมัครสมาชิก

```
14
15 export default function SignUp() {
16   const [formData, setFormData] = useState({
17     username: '',
18     email: '',
19     password: '',
20   });
21
22   const handleSubmit = async (event) => {
23     event.preventDefault();
24
25     try {
26       const response = await axios.post('http://localhost:8000/api/users/register', formData, {
27         headers: {
28           'Content-Type': 'application/json',
29         }
30       });
31
32     };
33
34     console.log('Response from backend:', response.data);
35   } catch (error) {
36     console.error('Error sending data to backend:', error);
37   }
38 };
39
40 const handleInputChange = (event) => {
41   const { name, value } = event.target;
42   setFormData({
43     ...formData,
44     [name]: value,
45   });
46 };
47
```

```

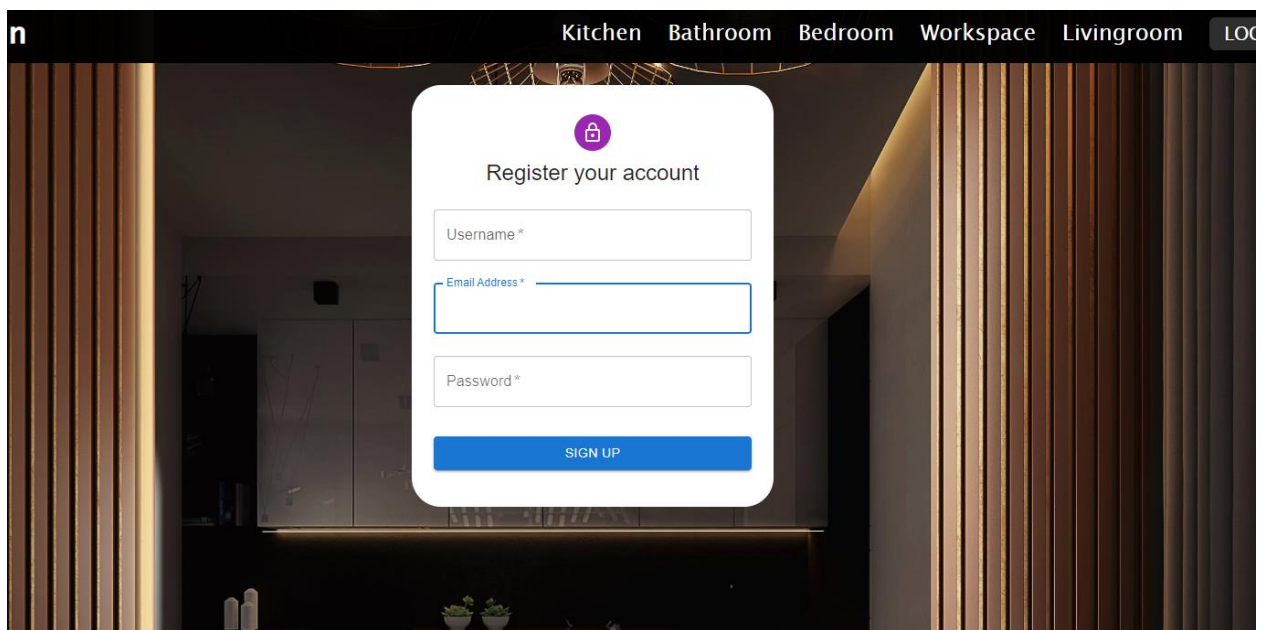
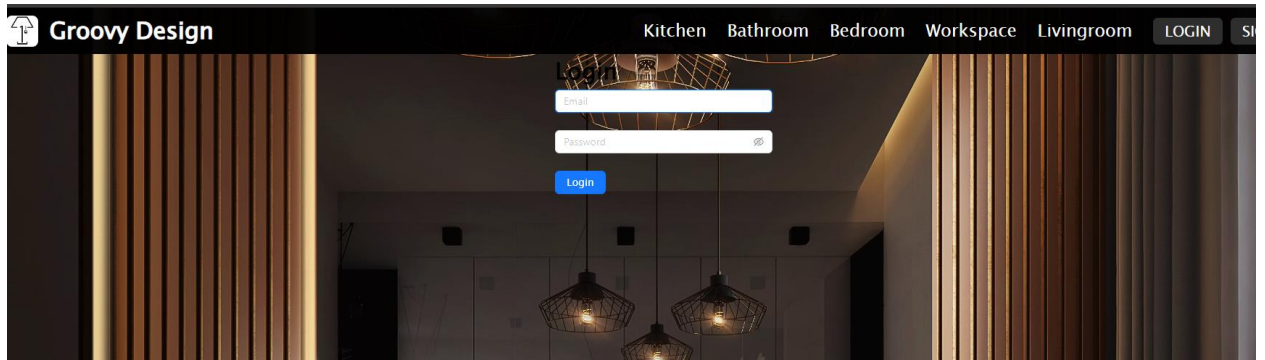
48   return (
49     <div>
50       <Nav />
51       <ThemeProvider theme={defaultTheme}>
52         <Container component="main" maxWidth="xs">
53           <CssBaseline />
54           <Box
55             sx={{
56               marginTop: 3,
57               display: 'flex',
58               flexDirection: 'column',
59               alignItems: 'center',
60               backgroundColor: 'white',
61               padding: 3,
62               borderRadius: 8,
63             }}
64           >
65             <Avatar sx={{ m: 1, bgcolor: 'secondary.main' }}>
66               <LockOutlinedIcon />
67             </Avatar>
68             <Typography component="h1" variant="h5">
69               Register your account
70             </Typography>
71             <Box component="form" onSubmit={handleSubmit} noValidate sx={{ mt: 1 }}>
72               <TextField
73                 margin="normal"
74                 required
75                 fullWidth
76                 id="Username"
77                 label="Username"
78                 name="username"
79                 autoComplete="username"
80                 autoFocus
81                 value={formData.username}
82                 onChange={handleInputChange}
83               />

```

```

83       />
84       <TextField
85         margin="normal"
86         required
87         fullWidth
88         id="email"
89         label="Email Address"
90         name="email"
91         autoComplete="email"
92         autoFocus
93         value={formData.email}
94         onChange={handleInputChange}
95       />
96       <TextField
97         margin="normal"
98         required
99         fullWidth
100        name="password"
101        label="Password"
102        type="password"
103        id="password"
104        autoComplete="current-password"
105        value={formData.password}
106        onChange={handleInputChange}
107      />
108
109      <Button
110        type="submit"
111        fullWidth
112        variant="contained"
113        sx={{ mt: 3, mb: 2 }}
114      >
115        Sign Up
116      </Button>
117    </Box>
118  </Container>
119 </ThemeProvider>
120 </div>
121 );
122 }

```



Nav.jsx สำหรับคลิกเข้าไปที่ห้องต่างๆ หน้าlogin registerและเมื่อทำการloginจะแสดง ปุ่ม logoutขึ้นมา

```
> node_modules
> public
> src
  > component
    login.jsx
    nav.jsx
    post.jsx
    register.jsx
  > context
    JS authContext.js
  > css
  > pages
    > bathroom
    > bedroom
    > kitchen
    > livingroom
    > workspace
    checkbox.jsx
    home.jsx
  > picture
    App.jsx
  # index.css
  JS index.js
  logo.svg
  README.md
  JS reportWebVitals.js
  JS setupTests.js
  webpack.config.js
.gitignore
package-lock.json
package.json

1 import React, { useContext, useEffect, useState, useRef } from 'react';
2 import { Link } from 'react-router-dom';
3 import Backdrop from '@mui/material/Backdrop';
4 import CircularProgress from '@mui/material/CircularProgress';
5 import '../css/home.css';
6 import logogroovy from '../picture/groovy.png';
7 import { AuthContext } from '../context/authContext';
8 import AddPost from './post';
9 import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
10
11
12 function Nav() {
13   const { currentUser, logout } = useContext(AuthContext);
14   const hamIconRef = useRef(null);
15   const mobileMenuRef = useRef(null);
16   const [open, setOpen] = React.useState(false);
17   useEffect(() => {
18     document.title = 'Groovy Design';
19   }, []);
20
21   const handleClose = () => {
22     setOpen(false);
23   };
24   const handleLogout = () => {
25     logout();
26     window.location.href = '/';
27   };
28   const handleOpen = () => {
29     setOpen(true);
30     setTimeout(() => {
31       window.location.href = '/login';
32     }, 1000);
33   };
34   const handleOpen2 = () => {
35     setOpen(true);
36     setTimeout(() => {
37       window.location.href = '/register';
38     }, 1000);
39   };
40   const handleOpen_bedroom = () => {
41     setOpen(true);
42     setTimeout(() => {
43       window.location.href = '/bedroom';
44     }, 1000);
45   };
46 }
```

```

46
47     const handleOpen_livingroom = () => {
48         setOpen(true);
49         setTimeout(() => {
50             window.location.href = '/livingroom';
51         }, 1000);
52     };
53
54     const handleOpen_Bathroom = () => {
55         setOpen(true);
56         setTimeout(() => {
57             window.location.href = '/bathroom';
58         }, 1000);
59     };
60
61     const handleOpen_kitchen = () => {
62         setOpen(true);
63         setTimeout(() => {
64             window.location.href = '/kitchen';
65         }, 1000);
66     };
67
68     const handleOpen_workspace = () => {
69         setOpen(true);
70         setTimeout(() => {
71             window.location.href = '/workspace';
72         }, 1000);
73     };
74
75     const handleOpenhome = () => {
76         setOpen(true);
77         setTimeout(() => {
78             window.location.href = '/';
79         }, 1000);
80     };
81
82     useEffect(() => {
83         const toggleMenu = () => {
84             const display = mobileMenuRef.current.style.display;
85             mobileMenuRef.current.style.display = display === 'block' ? 'none' : 'block';
86             hamIconRef.current.classList.toggle('change');
87         };
88

```

```

88
89     const hamIcon = hamIconRef.current;
90     hamIcon.addEventListener('click', toggleMenu);
91
92     return () => {
93         hamIcon.removeEventListener('click', toggleMenu);
94     };
95 }, []);
96
97 return (
98     <div>
99
100         <div className="container">
101             <div className="nav-con">
102                 <ul className='menu logo-container'>
103                     <Link to="/" onClick={handleOpenhome}>
104                         <li className="logo">
105                             <img src={logogroovy} />
106                             <a>Groovy&nbsp;Design</a>
107                         </li>
108                     </Link>
109                 </ul>
110                 <div className="MENU_NO_LOGO">
111                     <ul className='menu'>
112                         <li>
113                             <Link to="/kitchen" onClick={handleOpen_kitchen}>
114                                 <a>Kitchen</a>
115                             </Link>
116                         </li>
117
118                         <li>
119                             <Link to="/bathroom" onClick={handleOpen_Bathroom}>
120                                 <a>Bathroom</a>
121                             </Link>
122                         </li>
123
124                         <li>
125                             <Link to="/bedroom" onClick={handleOpen_bedroom}>
126                                 <a>Bedroom</a>
127                             </Link>
128                         </li>
129

```

```

</li>
  <Link to="/workspace" onClick={handleOpen_workspace}>
    <a>Workspace</a>
  </Link>
</li>

<li>
  <Link to="/livingroom" onClick={handleOpen_livingroom}>
    <a>Livingroom</a>
  </Link>
</li>
</ul>

<ul className='menu button-container'>
  {currentUser ? (
    <li>
      <div className='auth-box'>
        <button className="nav-login-btn">
          <AddPost />
        </button>
      </div>
      <div className="auth-box">
        <button className="nav-login-btn" onClick={handleLogout}>
          Logout
        </button>
      </div>
    </li>
  ) : (
    <li>
      <div className="auth-box">
        <Link to="/login"><button className="nav-login-btn">LOGIN</button></Link>
      </div>
      <div className="auth-box">
        <Link to="/register"><button className="nav-signup-btn">SIGNUP</button></Link>
      </div>
    </li>
  )}
</ul>

```

```

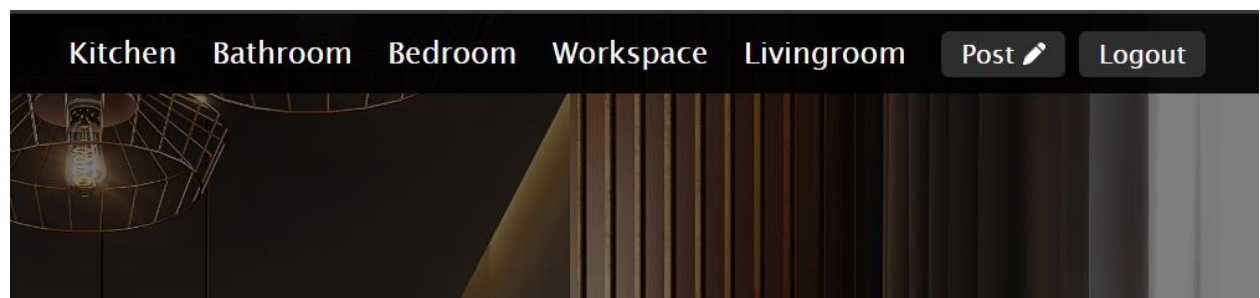
74     </div>
75   </div>
76   <div>
77     <ul className="moblie-menu" ref={mobileMenuRef}>
78       <li>
79         <Link to="/kitchen" onClick={handleOpen_kitchen}>
80           <a>Kitchen</a>
81         </Link>
82       </li>
83
84       <li>
85         <Link to="/bathroom" onClick={handleOpen_Bathroom}>
86           <a>Bathroom</a>
87         </Link>
88       </li>
89
90       <li>
91         <Link to="/bedroom" onClick={handleOpen_bedroom}>
92           <a>Bedroom</a>
93         </Link>
94       </li>
95
96       <li>
97         <Link to="/workspace" onClick={handleOpen_workspace}>
98           <a>Workspace</a>
99         </Link>
100      </li>
101
102      <li>
103        <Link to="/livingroom" onClick={handleOpen_livingroom}>
104          <a>Livingroom</a>
105        </Link>
106      </li>
107    </ul>

```

```

206   </li>
207   <AddPost />
208   {currentUser ? (
209     <li>
210       <Link onClick={handleLogout}>
211         Logout<FontAwesomeIcon icon="fa-solid fa-right-to-bracket" />
212       </Link>
213     </li>
214   ) : (
215     <React.Fragment>
216       <li><Link to="/login" onClick={handleOpen}>Login</Link></li>
217       <li><Link to="/register" onClick={handleOpen2}>Signup</Link></li>
218     </React.Fragment>
219   )}
220 </ul>
221 </div>
222 <Backdrop open={open} onClick={handleClose}>
223   <CircularProgress color="inherit" />
224 </Backdrop>
225 </div>
226 );

```



Post.jsx

```
FRONTEND
  > node_modules
  > public
  > src
    > component
      login.jsx
      nav.jsx
      post.jsx
      register.jsx
    > context
      JS authContext.js
    > css
    > pages
      > bathroom
      > bedroom
      > kitchen
      > livingroom
      > workspace
      checkbox.jsx
      home.jsx
      picture
      App.jsx
      # index.css
      JS index.js
      logo.svg
      README.md
      JS reportWebVitals.js
      JS setupTests.js
      webpack.config.js
      .gitignore
      {} package-lock.json
      {} package.json

src > component > post.jsx > ...
1  import React, { useState, useEffect } from 'react';
2  import axios from 'axios';
3  import { Select, Modal, Button, Upload, Form, Input, AutoComplete } from 'antd';
4  import { UploadOutlined } from '@ant-design/icons';
5  import { Link } from 'react-router-dom';
6  import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
7  import { faPen } from '@fortawesome/free-solid-svg-icons';
8
9
10 const { Option } = Select;
11
12 const PostForm = () => {
13   const [title, setTitle] = useState('');
14   const [description, setDescription] = useState('');
15   const [file, setFile] = useState(null);
16   const [CategoryID, setCategoryID] = useState('');
17   const [StyleID, setStyleID] = useState('');
18   const [error, setError] = useState('');
19   const [categories, setCategories] = useState([]);
20   const [styles, setStyles] = useState([]);
21   const [isModalVisible, setIsModalVisible] = useState(false);
22   const UserID = localStorage.getItem('UserID');
23   const [autoCompleteOptions, setAutoCompleteOptions] = useState([]);
24
25   useEffect(() => {
26     axios.get('http://localhost:8000/api/rooms/categories')
27       .then(response => {
28         setCategories(response.data);
29       })
30       .catch(error => {
31         console.error('Error fetching categories:', error);
32       });
33
34     axios.get('http://localhost:8000/api/rooms/styles')
35       .then(response => {
36         setStyles(response.data);
37       })
38       .catch(error => {
39         console.error('Error fetching styles:', error);
40       });
41   });
42   , []];
```

```
component
  login.jsx
  nav.jsx
  post.jsx
  register.jsx
context
  JS authContext.js
css
pages
  > bathroom
  > bedroom
  > kitchen
  > livingroom
  > workspace
  checkbox.jsx
  home.jsx
  picture
  App.jsx
  index.css
  index.js
  logo.svg
  README.md
  reportWebVitals.js
  setupTests.js
  webpack.config.js
  .gitignore
  package-lock.json
  package.json

44 const showModal = () => {
45   setIsModalVisible(true);
46 };
47
48 const handleOk = async () => {
49   setIsModalVisible(false);
50   await handleSubmit();
51 };
52
53 const handleCancel = () => {
54   setIsModalVisible(false);
55 };
56
57 const uploadProps = {
58   beforeUpload: file => {
59     setFile(file);
60     return false; // Prevent automatic upload
61   },
62   file,
63 };
64 const handleDescriptionChange = (value) => {
65   setDescription(value);
66 };
67 const handleSubmit = async () => {
68   if (!file) {
69     console.error('No file selected.');
```

```

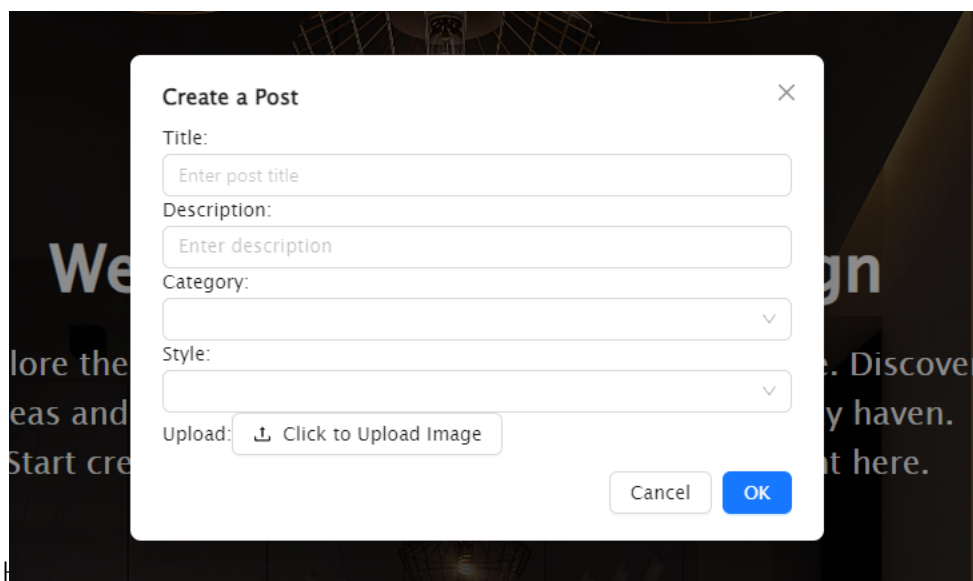
79     await axios.post(
80       'http://localhost:8000/api/post/addPost',
81       { title, description, img: imgUrl, uid: UserID, CategoryID, StyleID },
82       {
83         headers: {
84           Authorization: `Bearer ${localStorage.getItem('token')}`
85         }
86       }
87     );
88     console.log('Post created successfully');
89   } catch (err) {
90     setError(err.response.data.error);
91   }
92 };
93
94 return (
95   <div>
96     <Link type="primary" onClick={showModal}>
97       Post &nbsp; <FontAwesomeIcon icon={faPen} size="sm"/>
98     </Link>
99     <Modal title="Create a Post" open={isModalVisible} onOk={handleOk} onCancel={handleCancel}>
100       <Form>
101         <div>
102           <label>Title:</label>
103           <Input
104             type="text"
105             value={title}
106             onChange={(e) => setTitle(e.target.value)}
107             placeholder="Enter post title"
108           />
109         </div>
110         <div>
111           <label>Description:</label>
112           <AutoComplete
113             value={description}
114             options={autoCompleteOptions}
115             onChange={handleDescriptionChange}
116             placeholder="Enter description"
117             style={{ width: '100%' }}
118           />
119         </div>

```

```

120     <div>
121       <label>Category:</label>
122       <Select
123         value={CategoryID}
124         onChange={value => setCategoryID(value)}
125         placeholder="Select a category"
126         style={{ width: '100%' }}
127       >
128         {categories.map(category => (
129           <Option key={category.CategoryName} value={category.CategoryID}>
130             {category.CategoryName}
131           </Option>
132         ))}
133       </Select>
134     </div>
135     <div>
136       <label>Style:</label>
137       <Select
138         value={StyleID}
139         onChange={value => setStyleID(value)}
140         placeholder="Select a style"
141         style={{ width: '100%' }}
142       >
143         {styles.map(style => (
144           <Option key={style.StyleName} value={style.StyleID}>
145             {style.StyleName}
146           </Option>
147         ))}
148       </Select>
149     </div>
150     <div>
151       <label>Upload:</label>
152       <Upload {...uploadProps}>
153         <Button icon={UploadOutlined />>Click to Upload Image</Button>
154       </Upload>
155     </div>
156   </Form>
157 </Modal>
158 {error && <p>{error}</p>}
159 </div>
160 );
161 };
162

```



Create a Post

Title:

Enter post title

Description:

Enter description

Category:

Style:

Upload:

Click to Upload Image

Cancel

OK


```
1  import React from 'react';
2  import '../css/home.css';
3  import pic from '../picture/pexels-ilya-shakir-2440471.jpg';
4  import pic2 from '../picture/pexels-mark-2724749.jpg';
5  import pic3 from '../picture/pexels-pixabay-271618.jpg';
6  import pic4 from '../picture/bathroom-3563272.jpg';
7  import pic5 from '../picture/office-8503515.png';
8  import Nav from '../component/nav';
9  import { Link } from 'react-router-dom';
10 function Home() {
11   const [open, setOpen] = React.useState(false);
12
13   const handleOpen_bedroom = () => {
14     setOpen(true);
15     setTimeout(() => {
16       window.location.href = '/bedroom';
17     }, 1000);
18   };
19
20   const handleOpen_livingroom = () => {
21     setOpen(true);
22     setTimeout(() => {
23       window.location.href = '/livingroom';
24     }, 1000);
25   };
26
27   const handleOpen_Bathroom = () => {
28     setOpen(true);
29     setTimeout(() => {
30       window.location.href = '/bathroom';
31     }, 1000);
32   };
33
34   const handleOpen_kitchen = () => {
35     setOpen(true);
36     setTimeout(() => {
37       window.location.href = '/kitchen';
38     }, 1000);
39   };
40 }
```

```

41   const handleOpen_workspace = () => {
42     setOpen(true);
43     setTimeout(() => {
44       window.location.href = '/workspace';
45     }, 1000);
46   };
47
48
49   return (
50     <div>
51       <Nav />
52       <div className="main">
53         <div className="h1">
54           <div className="welcome">Welcome to Groovy Design</div>
55           Explore the latest in modern home decor on our website. Discover ideas and
56           products that will turn your home into a cozy haven.
57           Start creating warmth and comfort in your space right here. Enjoy!!!
58         </div>
59       </div>
60
61       <div className="se-main">
62         <div class="image-container">
63           <img src={pic} class="img"/>
64           <div class="overlay">
65             <Link to="/livingroom" onClick={handleOpen_livingroom}>
66               <p>LivingRoom</p>
67             </Link>
68           </div>
69         </div>
70
71         <div class="image-container">
72           <img src={pic2} class="img" />
73           <div class="overlay">
74             <Link to="/kitchen" onClick={handleOpen_kitchen}>
75               <p>Kitchen</p>
76             </Link>
77           </div>
78         </div>
79

```

```

    <div class="image-container">
      <img src={pic3} class="img"/>
      <div class="overlay">
        <Link to="/bedroom" onClick={handleOpen_bedroom}>
          <p>BedRoom</p>
        </Link>
      </div>
    </div>

    <div class="image-container">
      <img src={pic4} class="img"/>
      <div class="overlay">
        <Link to="/bathroom" onClick={handleOpen_Bathroom}>
          <p>Bathroom</p>
        </Link>
      </div>
    </div>

    <div class="image-container">
      <img src={pic5} class="img"/>
      <div class="overlay">
        <Link to="/workspace" onClick={handleOpen_workspace}>
          <p>Workspace</p>
        </Link>
      </div>
    </div>
  </div>
  <div>
    <footer className="footer">
      <div className = 'footer-con'>
        <ul className="menu-footer">
          <li><a href="#"> About Us</a></li>
          <li><a href="#"> Contact Us</a></li>
        </ul>
      </div>
    </footer>
  </div>
);
}

```

Imagelist.jsx

```
src > pages > bedroom > imagelist.jsx > MyGallery
1  import React, { useState, useEffect } from 'react';
2  import Box from '@mui/material/Box';
3  import ImageList from '@mui/material/ImageList';
4  import ImageListItem from '@mui/material/ImageListItem';
5  import { Select, Modal, Button, Upload, Form, Input, AutoComplete, Checkbox } from 'antd';
6  import { UploadOutlined, FormOutlined, DeleteTwoTone, HeartFilled } from '@ant-design/icons';
7  import { useTheme } from '@mui/material/styles';
8  import useMediaQuery from '@mui/material/useMediaQuery';
9  import axios from 'axios';
10 import '../css/img-list.css';
11 import '../css/bedroom.css';
12
13 const { Option } = Select;
14 const MyGallery = () => {
15   const [images, setImages] = useState([]);
16   const [modalVisible, setModalVisible] = useState(false);
17   const [selectedImage, setSelectedImage] = useState(null);
18   const [filterStyles, setFilterStyles] = useState([]);
19   const theme = useTheme();
20   const matchesSm = useMediaQuery(theme.breakpoints.up('sm'));
21   const matchesMd = useMediaQuery(theme.breakpoints.up('md'));
22   const matchesLg = useMediaQuery(theme.breakpoints.up('lg'));
23   const [title, setTitle] = useState('');
24   const [description, setDescription] = useState('');
25   const [file, setFile] = useState(null);
26   const [CategoryID, setCategoryID] = useState('');
27   const [StyleID, setStyleID] = useState('');
28   const [error, setError] = useState('');
29   const [categories, setCategories] = useState([]);
30   const [styles, setStyles] = useState([]);
31   const [isModalVisible, setIsModalVisible] = useState(false);
32   const UserID = localStorage.getItem('UserID');
33   const [autoCompleteOptions, setAutoCompleteOptions] = useState([]);
34   const [canEdit, setCanEdit] = useState(false);
35   const [likesCount, setLikesCount] = useState(0);
36 }
```

```
let cols;
if (matchesLg) {
  cols = 4;
} else if (matchesMd) {
  cols = 3;
} else if (matchesSm) {
  cols = 2;
} else {
  cols = 1;
}

useEffect(() => {
  axios.get('http://localhost:8000/api/rooms/categories')
    .then(response => {
      setCategories(response.data);
    })
    .catch(error => {
      console.error('Error fetching categories:', error);
    });
});
```

```

57     axios.get('http://localhost:8000/api/rooms/styles')
58     .then(response => {
59         setStyles(response.data);
60     })
61     .catch(error => {
62         console.error('Error fetching styles:', error);
63     });
64     const fetchImages = async () => {
65         try {
66             const response = await axios.get('http://localhost:8000/api/rooms/getImage/1');
67             setImages(response.data);
68         } catch (error) {
69             console.error('Error fetching images:', error);
70         }
71     };
72
73     fetchImages();
74
75     const fetchLikesCount = async () => {
76         try {
77             const response = await axios.get(`http://localhost:8000/api/post/likesCount/${selectedImage.id}`);
78             setLikesCount(response.data.likesCount);
79         } catch (error) {
80             console.error('Error fetching likes count:', error);
81         }
82     };
83
84     if (selectedImage) {
85         fetchLikesCount();
86     }
87 }, [selectedImage]);

```

```

91 const handleLike = async () => {
92     try {
93         await axios.post(`http://localhost:8000/api/post/like/${selectedImage.id}`, { postId: selectedImage.id, userId: UserID
94             headers: {
95                 Authorization: `Bearer ${localStorage.getItem('token')}`
96             }
97         });
98         setLikesCount(likesCount + 1); // เพิ่มจำนวนถูกใจทันทีหลังจากกดถูกใจ
99         console.log('Liked successfully');
100     } catch (error) {
101         console.error('Error liking post:', error);
102     }
103 };
104
105 ///////////////////////////////////////////////////LIKE//////////////////////////////////////
106 const uploadProps = {
107     beforeUpload: file => {
108         setFile(file);
109         return false; // Prevent automatic upload
110     },
111     file,
112 };
113 const handleClick = (image) => {
114     setSelectedImage(image);
115     setTitle(image.title);
116     setDescription(image.description);
117     setCategoryID(image.CategoryID);
118     setStyleID(image.StyleID);
119     setModalVisible(true);
120
121     // ตรวจสอบสิทธิ์การแก้ไข โดยเปรียบเทียบ uid ของรูปภาพกับ UserID ใน localStorage
122     const currentUserID = localStorage.getItem('UserID');
123     setCanEdit(image.uid.toString() === currentUserID);
124 };
125 const showModal = () => {
126     setIsModalVisible(true);
127
128     const formatDateTime = (dateTimeString) => {
129         const dateTime = new Date(dateTimeString);
130         const day = dateTime.getDate().toString().padStart(2, '0');
131         const month = (dateTime.getMonth() + 1).toString().padStart(2, '0');
132         const year = dateTime.getFullYear();
133         const hours = dateTime.getHours().toString().padStart(2, '0');
134         const minutes = dateTime.getMinutes().toString().padStart(2, '0');
135
136         return `${day}-${month}-${year}-${hours}:${minutes}`;
137     };
138
139     ///////////////////////////////////////////////////
140     const handleDelete = () => {
141         Modal.confirm({
142             title: 'Are you sure delete this post?',
143             okText: 'Yes, delete it',
144             okType: 'danger',
145             cancelText: 'No, cancel',
146             onOk: async () => {
147

```

```

5 const handleCheckboxChange = (StyleName) => {
6   // ปรับ state ของ checkbox เมื่อถูกเลือกหรือไม่ถูกเลือก
7   if (filterStyles.includes(StyleName)) {
8     setFilterStyles((prevStyles) => prevStyles.filter((filterStyle) => filterStyle !== StyleName));
9   } else {
10    setFilterStyles((prevStyles) => [...prevStyles, StyleName]);
11  }
12 };
13 const handleEdit = async (image) => {
14   if (!file) {
15     console.error('No file selected.');
```

```

175 const handleCheckboxChange = (StyleName) => {
176   // ปรับ state ของ checkbox เมื่อถูกเลือกหรือไม่ถูกเลือก
177   if (filterStyles.includes(StyleName)) {
178     setFilterStyles((prevStyles) => prevStyles.filter((filterStyle) => filterStyle !== StyleName));
179   } else {
180     setFilterStyles((prevStyles) => [...prevStyles, StyleName]);
181   }
182 };
183 const handleEdit = async (image) => {
184   if (!file) {
185     console.error('No file selected.');
```

```

216 const handleDescriptionChange = (value) => {
217   setDescription(value);
218 };
219 return (
220   <Box sx={{}} className="style">
221     <div className='bed-cont'>
222       <div className='bed-cont2'>
223         <div className='checkbox-cont'>
224           <h1>Style&nbsp;rooms</h1>
225           <hr style={{ border: '1px solid #ccc', marginTop: '-3px', marginBottom:'5px'}} />
226           {[ 'Modern', 'Minimal', 'Art deco', 'Boho', 'Country', 'Industrial', 'Rustic' ].map((StyleName) => (
227             <Checkbox key={StyleName} className='checkbox-cont-text' onChange={() => handleCheckboxChange(StyleName)}>
228               {`${StyleName}`}
229             </Checkbox>
230           )))
231         </div>
232       <div className='bed-cont-mygall'>
233         <ImageList variant="masonry" cols={cols} gap={12}>
234           {images
235             .filter((item) => (filterStyles.length === 0 || filterStyles.includes(item.StyleName)))
236             .map((item) => (
237               <ImageListItem key={item.id} className="image-list-item" onClick={() => handleImageClick(item)}>
238                 <img
239                   srcSet={` ${item.img}?w=1080&fit=crop&auto=format&dpr=2 2x`}
240                   src={` ../upload/${item.img}?w=1080&fit=crop&auto=format`}
241                   alt={item.title}
242                   loading="lazy"
243                 />
244               </ImageListItem>
245             )))
246         </ImageList>
247       </div>
248     </div>
249   </div>
250 </div>
251 <Modal
252   title="Image Details"
253   open={modalVisible}
254   onCancel={closeModal}
255   footer={null}
256   style={{ paddingTop: '30px', paddingBottom: '30px', height: '200%' }}
257 >
258

```

```

259 {selectedImage && (
260   <div>
261     <img
262       src={` ../upload/${selectedImage.img}`}
263       alt={selectedImage.title}
264       style={{ maxWidth: '100%', height: 'auto' }}
265     />
266     <p><strong>Title:</strong> {selectedImage.title}</p>
267     <p><strong>Description:</strong> {selectedImage.description}</p>
268     <p><strong>Username:</strong> {selectedImage.Username}</p>
269     <p><strong>Create Date:</strong> {formatDateTime(selectedImage.CreateDate)}</p>
270     <p><strong>Category:</strong> {selectedImage.CategoryName}</p>
271     <p><strong>Style:</strong> {selectedImage.StyleName}</p>
272     {canEdit ? [ // ตรวจสอบสิทธิ์การแก้ไขก่อนแสดงปุ่ม
273       <Button key="edit" type="primary" onClick={showModal}>
274         <FormOutlined /> Edit
275       </Button>,
276       <Button key="delete" type="primary" danger onClick={handleDelete}>
277         <DeleteTwoTone /> Delete
278       </Button>
279     ] : null}
280

```

```

281 <Button
282   className="like-button"
283   type="text"
284   icon={<HeartFilled onClick={handleLike} style={{ color: 'red' }} />}
285 >
286   {likesCount}
287 </Button>
288
289 <Modal title="Edit Post" open={isModalVisible} onOk={handleOk} onCancel={handleCancel}>
290   <Form>
291     <div>
292       <label>Title:</label>
293       <Input
294         type="text"
295         value={title}
296         onChange={(e) => setTitle(e.target.value)}
297         placeholder="Enter post title"
298       />
299     </div>
300     <div>
301       <label>Description:</label>
302       <AutoComplete
303         value={description}
304         options={autoCompleteOptions}
305         onChange={handleDescriptionChange}
306         placeholder="Enter description"
307         style={{ width: '100%' }}
308       />
309     </div>
310     <div>
311       <label>Category:</label>
312       <Select
313         value={CategoryID}
314         onChange={value => setCategoryID(value)}
315         placeholder="Select a category"
316         style={{ width: '100%' }}
317       >
318         {categories.map(category => (
319           <Option key={category.CategoryName} value={category.CategoryID}>
320             {category.CategoryName}
321           </Option>
322         ))}
323     </Select>
324   </div>

```

```

    <div>
      <label>Style:</label>
      <Select
        value={StyleID}
        onChange={value => setStyleID(value)}
        placeholder="Select a style"
        style={{ width: '100%' }}
      >
        {styles.map(style => (
          <Option key={style.StyleName} value={style.StyleID}>
            {style.StyleName}
          </Option>
        ))}
      </Select>
    </div>
    <div>
      <label>Upload:</label>
      <Upload {...uploadProps}>
        <Button icon={<UploadOutlined />}>Click to Upload Image</Button>
      </Upload>
    </div>
  </div>
</Form>
</Modal>
</div>
)}
</Modal>

```


Bedroom.jsx

```
src > pages > bedroom > bedroom.jsx > bedroom
1  import React from 'react';
2
3  import '../css/bedroom.css'
4
5  import Nav from '../component/nav'
6
7  import MyGallery from './imagelist';
8
9
10 export default function Bedroom() {
11   return (
12     <div>
13       <Nav />
14
15       <div className='bed-main'>
16         <div className='bed-cont'>
17           <div className='bed-name'>
18             <p>Bed room</p>
19           </div>
20           <div className='bed-cont-mygal'>
21             <MyGallery/>
22           </div>
23         </div>
24       </div>
25     </div>
26   );
27 }
28
29
```

ใช้ imagelist.jsx และ bedroom.jsx เหมือนกันทุกห้อง

Bathroom

Kitchen

Livingroom

Workspace