เรื่อง คู่มือผู้พัฒนา การติดตั้ง และแนวทางการปฏิบัติการทำงานเป็นทีม

เสนอ

อ.ดร.บุญรัตน์ เผดิมรอด

จัดทำโดย

นางสาว รัตนามน สถิรวัฒนานนท์ 6220500717

นาย ธนกร โพธิปัญญาธรรม 6220503228

นาย จิตติพล คำอุไร 6220504623

นาย ชาญวิชญ์ จำปา 6220504640

นาย อาทฤต เย็นเปรม 6220504801

รายงานนี้เป็นส่วนหนึ่งของวิชา Software Engineering **02204371-60**

ภาคเรียนที่ 2 ปีการศึกษา 2564

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

เว็บไซต์ระบบบริหารจัดการการฝึกงานแบบออนไลน์

เอกสารประกอบโปรแกรมสำหรับผู้เขียนโปรแกรม (Technical Documentation)

ที่มาและความสำคัญ

เนื่องด้วยระบบการยื่นเรื่องฝึกงานในปัจจุบันของภาควิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ ยังต้องรอทางภาควิชาในการนำไฟล์เอกสารมาประกาศทางเว็บไซต์ นิสิตจึงไม่สามารถสืบค้นสถานประกอบการที่มีความประสงค์รับนิสิตฝึกงานล่วงหน้าได้ และยังยากต่อการติดตามผลเนื่องจากนิสิตจะต้องติดต่อเจ้าหน้าที่ภาควิชาเพื่อติดตามผลการยื่นคำร้องขอฝึกงาน และมีการแก้ไขที่บ่อยครั้ง ส่งผลให้การตรวจสอบข้อมูลการฝึกงานของนิสิตเพื่ออนุมัติการจบการศึกษามีความล้าช้าและไม่สะดวกรวดเร็ว

เพื่อออกแบบและพัฒนาระบบเกี่ยวกับการฝึกงานแบบออนไลน์สำหรับภาควิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ให้ดียิ่งขึ้น และยังสามารถจัดทำฐานข้อมูลที่เกี่ยวข้องกับการฝึกงานของนิสิตภาควิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ เพื่อให้สะดวกต่อนิสิต บุคลากร และอาจารย์ที่เกรายวข้อง

วัตถุประสงค์ของการพัฒนาระบบ

1. เพื่อออกแบบและพัฒนาระบบจัดการการฝึกงานแบบออนไลน์สำหรับภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์
2. เพื่อจัดทำฐานข้อมูลที่เกี่ยวข้องการฝึกงานของนิสิตภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์
3. เพื่อปรับปรุงการจัดการการฝึกงานภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์

วิธีการใช้งาน

เว็บไซต์แบ่งการใช้งานแบ่งผู้ใช้ออกเป็น 4 ประเภท คือ

1. นิสิตภาควิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน มีสิทธิ์การใช้งานคือ นิสิตตั้งแต่ปี 3 เป็นต้นไป โดยจะต้องไม่เคยผ่านการฝึกงานหรือสหกิจมาก่อนโดยนิสิต 1 คนจะ 5 – 20 ครั้ง ต่อปี สำหรับการติดต่อการฝึกงานและสหกิจ เช่น การ download เอกสาร, ยื่นคำร้อง, upload เอกสาร, ตรวจสอบผลการยื่นคำร้องฝึกงานและ สหกิจ
2. อาจารย์ ภาควิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน ซึ่งจะสิทธิ์การใช้งานคือ professorโดยบุคลากร 1 คนจะใช้ 30 – 50 ครั้งต่อปี ใช้สำหรับการ เพื่อใช้สำหรับการจัดการการฝึกงานและสหกิจ เช่น จัดการสถานประกอบการ (เพิ่ม, แก้ไข, ลบ), พิจารณาอนุมัติ/ไม่อนุมัติคำร้องขอฝึกงาน
3. เจ้าหน้าที่ เจ้าหน้าที่ ภาควิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศตร์ วิทยาเขต กำแพงแสน ซึ่งจะมีสิทธิ์ คือ staff โดยบุคลากร 1 คนจะใช้ 30 - 50 ครั้งต่อปี เพื่อใช้สำหรับการ จัดการเกี่ยวกับ ข้อมูลการฝึกงาน, เอกสารเกี่ยวกับการฝึกงาน, ติดตามผลการฝึกงานของนิสิต แต่ไม่สามารถ พิจารณา อนุมัติ/ไม่อนุมัติคำร้องขอฝึกงานได้4.บุคคลทั่วไป
4. บุคคลทั่วไป จะมีสิทธิ์การใช้งานคือ General จะสามารถดูเอกสารและประกาศต่างๆเกี่ยวกับการฝึกงานได้ แต่ไม่สามารถ login เพื่อทำการดำเนินเรื่องเกี่ยวกับการฝึกงานได้

แนวคิดเกี่ยวกับการออกแบบโปรแกรม

ทางผู้พัฒนามองเห็นถึงปัญหาของการค้นหาข้อมูลของสถานที่ฝึกงานและสหกิจซึ่งมีความยุ่งยาก ล่าช้าและมีรายละเอียดที่ไม่ครบถ้วนตามความต้องการของนิสิต ซึ่งอาจต้องทำการค้นหาหรือติดต่อกับสถานที่ฝึกงานและสหกิจจึงจะได้ข้อมูลที่ครบถ้วนตามความต้องการ ทางผู้พัฒนาจึงจัดทำเว็บไซต์เพื่อช่วยให้ผู้ใช้งานสามารถค้นหาข้อมูลหรือสถานที่ฝึกงานและสหกิจต่างๆได้ง่ายยิ่งขึ้น

รายละเอียดภาษาและโปรแกรมที่ใช้

ภาษา Java Script

ทางผู้พัฒนาใช้ภาษา java script ในรูปแบบ jsx สำหรับสร้างหน้าเว็บไซต์ ส่วน < head> ประกอบด้วยข้อมูล เช่น ชื่อ คำอธิบาย และลิงก์ไปยัง CSS ส่วน <body> ประกอบด้วยส่วนที่เป็นเนื้อหาของเว็บเพจ เช่น ข้อความ รูปภาพ พื้นหลัง สีตัวอักษร และแบบฟอร์มต่าง ๆ

React

เป็น Library ที่ใช้ในการพัฒนาเว็บไซต์

ภาษา CSS

ทางผู้พัฒนาใช้ภาษา CSS เพื่อพัฒนาลักษณะรูปแบบ ใส่พื้นหลัง หรือเพิ่มกรอบข้อความ การจัดเลย์เอาท์ และการทำให้ภาพลักษณ์ของเว็บสวยงามมากขึ้น

โปรแกรม GitHub

เป็นเว็บบริการที่เก็บ Repository บนออนไลน์แพลตฟอร์ม พื้นที่ทางอินเทอร์เน็ต (hosting service) สำหรับเก็บการควบคุมการปรับปรุงแก้ไข (version control) โดยใช้ Git สามารถเข้าถึงและทำงานร่วมกับคนอื่นได้ผ่านหน้าเว็บไซต์ได้ทุกที่ทุกเวลา

โปรแกรม Visual Studio Code

เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด

Project Team lesson learn

- การวางแผน - วางแผนการทำงาน ทำเพื่อใคร ทำเพื่ออะไร ความต้องการเป็นแบบไหน
- ขอบเขต - วางขอบเขตของงานเพื่อให้เข้าถึงปัญหางานได้ตรงประเด็น
- กำหนดการ - วางแผนและจัดสรรวันและเวลาของการทำงานทั้งหมด
- ค่าใช้จ่าย - กำหนดค่าใช้จ่ายของงานทั้งหมด
- คุณภาพ - ดำเนินงานตามที่กำหนด และทำการทดสอบสิ่งงานที่ได้รับมา
- ทรัพยากรบุคคล - แบ่งบุคลากรให้เป็นส่วน ๆ ตามหน้าที่
- การสื่อสาร - สื่อสารกับบุคลากรด้วยกันเองเพื่อให้งานดียิ่งขึ้น

Function ส่วน client

ในส่วนของ Function AddHospitalComponent นี้เป็นการเพิ่มข้อมูลของ สถานที่ เพื่อมาแสดงผลในส่วนต่างๆ

```
const AddHospitalComponent = () => {
  const [allprovinces, setAllProvinces] = useState([]);
  const [alldistrict, setAllDistrict] = useState([]);
  const fetchData = () => {
    axios
      .get(`https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces`)
      .then((res) => {
        console.log(res.data.data);
        setAllProvinces(res.data.data);
      });
  };
  const fetchDistrict = (pro) => {
    axios
      .get(`https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces/${pro}`)
      .then((res) => {
        console.log(res.data.data);
        setAllDistrict(res.data.data);
      });
  };
```

ในส่วนนี้เป็นการดึงข้อมูลในส่วนของการดึงประกาศ

```
const AnnounceComponent=()=>{
    const [searchAnnounce,setSearchAnnounce]=useState('');
    const [district,setDistrict]=useState([]);
    const [provinces,setProvinces]=useState([]);
    const [announce,setAnnounce]=useState([]);
    const fetchData=()=>{
        axios.get(`http://localhost:5000/api/announces`)
        .then((res)=>{
            setAnnounce(res.data)
        }).catch((err)=>{
            console.log(err)
        })
        axios.get(`https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces`)
        .then((res) => {
        console.log(res.data.data);
        setProvinces(res.data.data);
      });
    }
    const fetchDistrict=(pro)=>{
        axios
          .get(`https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces/${pro}`)
          .then((res)=>{
            console.log(res.data.data);
            setDistrict(res.data.data);
          })
      }
```

ในส่วน Function ApproveStatusForSuperComponent นี้เป็นส่วนอนุมัติของส่วนต่างๆของส่วนอาจารย์

```
const ApproveStatusForSuperComponent = (props) => {
  const [searchAnnounce, setSearchAnnounce] = useState("");
  const [requests, setRequest] = useState([]);
  const [detail, setDetail] = useState([""]);

  const [num, setNum] = useState(0);

  useEffect(() => {
    // axios.get(`http://localhost:5000/api/requests/${props.match.params._id}`)
    axios
      .get(`http://localhost:5000/api/requests`)
      .then((response) => {
        setRequest(response.data);
        // setNum(response.data._id);
        console.log("response.data");
        console.log(response.data);
      })
      .catch((err) => alert(err));
    // eslint-disable-next-line
```

ในส่วน Function CheckStatusForNisitComponent นี้เป็นการดึงข้อมูลในส่วนของ ตัวเช็คข้อมูลต่างๆของนิสิต

```
const CheckStatusForNisitComponent = () => {

  const [searchAnnounce, setSearchAnnounce] = useState("");
  const [requests, setRequest] = useState([]);

  const fetchData = () => {
    axios
      .get(`http://localhost:5000/api/requests`)
      .then((res) => {
        setRequest(res.data);
        console.log(res);
      })
      .catch((err) => {
        console.log(err);
      });
  };
```

ในส่วน Function CheckStatusForComponent นี้เป็นการดึงข้อมูลในส่วนของ ตัวเช็คข้อมูลต่างๆของอาจารย์

```
const CheckStatusForSuperComponent = () => {
  const [searchAnnounce, setSearchAnnounce] = useState("");
  const [requests, setRequest] = useState([]);
  const [users, setUser] = useState([]);

  const [filteredResults, setFilteredResults] = useState([]);
  const [searchInput, setSearchInput] = useState('');


  const searchItems = (searchValue) => {
    setSearchInput(searchValue)
    if (searchInput !== '') {
        const filteredData = requests.filter((item) => {
            return Object.values(item).join('').toLowerCase().includes(searchInput.toLowerCase())
        })
        setFilteredResults(filteredData)
        console.log(filteredData);
    }
    else{
        setFilteredResults(requests)
    }
}
```

ในส่วนนี้เป็นส่วนของ Function DocumentComponent เป็นการดึงข้อมูลในส่วนของเอกสารต่างๆ

```
const DocumentComponent = () => {
  const [allprovinces, setAllProvinces] = useState([]);
  const [alldistrict, setAllDistrict] = useState([]);
  const fetchData = () => {
    axios
      .get(`https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces`)
      .then((res) => {
        console.log(res.data.data);
        setAllProvinces(res.data.data);
      });
  };
  const fetchDistrict = (pro) => {
    axios
      .get(
        `https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces/${pro}`
      )
      .then((res) => {
        console.log(res.data.data);
        setAllDistrict(res.data.data);
      });
  };
```

เป็นการดึงข้อมูลในส่วนของหน้า login

```
const signinForm=(event)=>{
    event.preventDefault();
    axios.post(`http://localhost:5000/api/login`,{studentID,password}).then(res=>{
        console.log(res.data)
        setUser(res.data)
        console.log(user)
        setUser({...user,
            studentID:"",
            password:""})
            Swal.fire(
                'เข้าสู่ระบบสำเร็จ',
            ).then(()=>{
                authenticate(res,()=>props.history.push("/"))
            })
            // ).then(()=>{
            //     window.location.href = "/"
            // })
    })
    .catch((err)=>{
        Swal.fire(
            'เข้าสู่ระบบไม่สำเร็จ',
            'StudentID or Password is wrong',
            'error',
            err.response.data.error,
          )
    })
}
```

ในส่วนของ Function NewsAddConponent เป็นการเพิ่มข้อมูลประกาศต่างๆเข้าไป

```
const NewsAddComponent=(props)=>{

  const [announces, setannounces] = useState([]);

    useEffect(()=>{
      axios.get(`http://localhost:5000/api/announces/${props.match.params._id}`)
      .then(response=>{
          const {_id,titleName,detail,phoneNumber,email,type}= response.data
          setState({...state,_id,titleName,detail,phoneNumber,email,type})
      })
      .catch(err=>alert(err))
      // eslint-disable-next-line
    },[])
```

ในส่วนของ Function OrganizationComponent และ OrganizationComponent1 เป็นการดึงข้อมูลของสถานประกอบการ

```
import { Link, withRouter } from "react-router-dom";
const OrganizationComponent = () => {
  const [companies, setcompanies] = useState([]);
  const [provinces, setProvinces] = useState([]);
  const [subDistrict,setSubDistrict] = useState([]);
  const [searchHospital,setSearchHospital] =useState('');
  const [district, setDistrict] = useState([]);
  const fetchData = () => {
    axios
      .get(`http://localhost:5000/api/companies`)
      .then((response) => {
        console.log(response.data)
        setcompanies(response.data);

      })
      .catch((err) => console.log(err));
  };
```

```
const OrganizationComponent1 = () => {

  const [companies, setcompanies] = useState([]);

  const fetchData = () => {
    axios
      .get(`http://localhost:5000/api/companies`)
      .then((res) => {
        setcompanies(res.data);
        console.log(res.data);
        console.log("ANNOUCE  = ")
        console.log(companies)
      })
      .catch((err) => {
        console.log(err);
      });
  };
```

ในส่วนของ Function RegisterComponent เป็นการดึงข้อมูลส่วนของการลงทะเบียนผู้ใช้

```
const RegisterComponent=()=>{
    const [hospital,setHospital]=useState([])

    const fetchData=()=>{
        axios.get(`http://localhost:5000/api/hospitals`).then(res=>{
          setHospital(res.data)
          console.log(res.data)
          console.log(res.status)
        })

    }
    useEffect(()=>{
        fetchData()
    },[])
```

ในส่วนของ Function ResultComponent ใช้สำหรับ get ค่าผลการฝึกงาน

```
const ResultComponent = () => {
  const [allprovinces, setAllProvinces] = useState([]);
  const [alldistrict, setAllDistrict] = useState([]);
  const fetchData = () => {
    axios
      .get(`https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces`)
      .then((res) => {
        console.log(res.data.data);
        setAllProvinces(res.data.data);
      });
  };
  const fetchDistrict = (pro) => {
    axios
      .get(
        `https://thaiaddressapi-thaikub.herokuapp.com/v1/thailand/provinces/${pro}`
      )
      .then((res) => {
        console.log(res.data.data);
        setAllDistrict(res.data.data);
      });
  };
```

Function ส่วน Server

ในส่วนนี้จะเป็นส่วนของ Announces ซึ่งใน Function getannounces ทำหน้าที่รับค่าของ announces

```
const getannounces = asyncHandler(async (req, res) => {
    const announces = await Announce.find().select('-__v')
    res.status(200).json(announces)
})
```

```
const getannounce = asyncHandler(async (req, res) => {
    var request
    if (mongoose.Types.ObjectId.isValid(req.params.id)) {
        request = await Announce.findById(req.params.id)
    }
    else {
        res.status(400)
        throw new Error('ใส่ object id มาผิด')
    }

    if (request) {
        res.status(200).json(request)
    }
    else {
        res.status(400)
        throw new Error(' เอา object id ไปหาแล้ว หาไม่เจออะ พยายามละ โทษหวะ')
    }
})
```

ต่อมาในส่วน **Function setannounce ก็จะทำการเซ็ตค่าที่ได้มาจาก getannounces**

```
const setannounce = asyncHandler(async (req, res) => {
    if(!req.body.titleName){
        res.status(400)
        throw new Error('ใส่ titleName ด้วย')
    }
    if(!req.body.detail){
        res.status(400)
        throw new Error('ใส่ detail ด้วย')
    }
    if(!req.body.phoneNumber){
        res.status(400)
        throw new Error('ใส่ phoneNumber ด้วย')
    }
    if(!req.body.email){
        res.status(400)
        throw new Error('ใส่ email ด้วย')
    }
    if(!req.body.type){
        res.status(400)
        throw new Error('ใส่ type ด้วย')
    }
    const announce = await Announce.create({
        titleName: req.body.titleName,
        detail: req.body.detail,
        phoneNumber: req.body.phoneNumber,
        email: req.body.email,
        type: req.body.type
```

ในส่วนของ Function putannounce เป็นการทำใส่ในส่วนปุ่มที่เอาไว้เพิ่มค่า

```
const putannounce = asyncHandler(async (req, res) => {
    const announce = await Announce.findById(req.params.id)
    if (!announce) {
        res.status(400)
        throw new Error('annouce id not found')
    }

    const updatedannounce = await Announce.findByIdAndUpdate(req.params.id, req.body, { new: true })
    res.status(200).json(updatedannounce)
})
```

ในส่วนของ Function deleteannounce เป็นการลบค่าที่มีอยู่

```
const deleteannounce = asyncHandler(async (req, res) => {
    const announce = await Announce.findById(req.params.id)
    if(!announce){
        res.status(400)
        throw new Error('announce id not found')
    }
    const deleteannounce = await Announce.findByIdAndDelete(req.params.id)
    res.status(200).json({id:req.params.id})
})
```

ในส่วนนี้จะเป็นส่วนของ Company ซึ่งในแต่ละ Function จะทำหน้าที่คล้ายในส่วนของ Announce

ในภาพคือ Function getCompanys และ getCompany

```
const getCompanys = asyncHandler(async (req, res) => {
    const companys = await Company.find().select('-__v')
    res.status(200).json(companys)
})
```

```
const getCompany = asyncHandler(async (req, res) => {
    var company
    if (mongoose.Types.ObjectId.isValid(req.params.id)) {
        company = await Company.findById(req.params.id)
    }
    else {
        res.status(400)
        throw new Error('ใส่ object id มาผิด')
    }

    if (company) {
        res.status(200).json(company)
    }
    else {
        res.status(400)
        throw new Error(' เอา object id ไปหาแล้ว หาไม่เจออะ พยายามละ โทษหวะ')
    }
})
```

ในภาพคือ Function setcCmpany

```
const setCompany = asyncHandler(async (req, res) => {
    if(!req.body.companyName){
        res.status(400)
        throw new Error('ใส่ companyName ด้วย')
    }
    if(!req.body.businessType){
        res.status(400)
        throw new Error('ใส่ businessType ด้วย')
    }
    if(!req.body.address){
        res.status(400)
        throw new Error('ใส่ address ด้วย')
    }
    if(!req.body.phoneNumber){
        res.status(400)
        throw new Error('ใส่ phoneNumber ด้วย')
    }
    if(!req.body.tel){
        res.status(400)
        throw new Error('ใส่ tel ด้วย')
    }
    const company = await Company.create({
        companyName: req.body.companyName,
        businessType: req.body.businessType,
        address: req.body.address,
        phoneNumber: req.body.phoneNumber,
        tel: req.body.tel
```

ในภาพคือ Function putCompany

```
const putCompany = asyncHandler(async (req, res) => {
    const company = await Company.findById(req.params.id)
    if (!company) {
        res.status(400)
        throw new Error('user id not found')
    }

    const updatedCompany = await Company.findByIdAndUpdate(req.params.id, req.body, { new: true })
    res.status(200).json(updatedCompany)
})
```

ในภาพคือ Function deleteCompany

```
const deleteCompany = asyncHandler(async (req, res) => {
    const company = await Company.findById(req.params.id)
    if(!company){
        res.status(400)
        throw new Error('Company id not found')
    }
    const deleteCompany = await Company.findByIdAndDelete(req.params.id)
    res.status(200).json({id:req.params.id})
})
```

ต่อมาเป็นในส่วนของการ Login ก็จะมี Function login เพียงอันเดียวเพื่อเป็นการดึงข้อมูลของ username และ password มาเพื่อทำการ login

```
const login = asyncHandler(async (req, res) => {
    const { studentID, password } = req.body
    console.log("login "+studentID+password)
    //validate user input
    if (!(studentID && password)) {
        res.status(400)
        throw new Error('studentID or password are  required')
    }
    const user = await User.findOne({ studentID }).select('+password')
    if (user) {
        // if ((await bcrypt.compare(password, user.password))) {
        if (password == user.password) {
            const token = jwt.sign(
                { user_id: user._id, studentID },
                process.env.TOKEN_KEY, {
                expiresIn: "24h"
            })
            //save token in uuser
            const oldUser = await User.findOne({ studentID },'-createdAt -updatedAt -__v +role')
            //if want to deselect _id await User.findOne({ studentID }, '-_id')
            oldUser.token = token
            res.status(200).json(oldUser)
        }
        else{
            res.status(400)
            throw new Error('wrong userName or password')
        }
```

ในส่วนนี้จะเป็นส่วนของ Company ซึ่งในแต่ละ Function จะทำหน้าที่คล้ายในส่วนของ Announce

ในภาพคือ Function getRequests และ getRequest

```
const getRequests = asyncHandler(async (req, res) => {
    const requests = await Request.find().select('-__v')
    res.status(200).json(requests)
})
```

```
const getRequest = asyncHandler(async (req, res) => {
    var request
    if (mongoose.Types.ObjectId.isValid(req.params.id)) {
        request = await Request.findById(req.params.id)
    }

    if (!request) {
        request = await Request.find({ 'studentID': req.params.id })
        // request = await Request.findOne({'studentID':req.params.id})
    }
    if (request) {
        res.status(200).json(request)
    }
    else {
        res.status(404)
        throw new Error("หาไม่เจอ")
    }
})
```

ในส่วนนี้คือ Function setRequest

```
const setRequest = asyncHandler(async (req, res) => {
    if (!req.body.companyName) {
        res.status(400)
        throw new Error('ใส่ companyName ด้วย')
    }
    if (!req.body.typeRequest) {
        res.status(400)
        throw new Error('ใส่ typeRequest ด้วย')
    }
    if (!req.body.jobTitle) {
        res.status(400)
        throw new Error('ใส่ jobTitle ด้วย')
    }
    if (!req.body.studentID) {
        res.status(400)
        throw new Error('ใส่ studentID ด้วย')
    }
    if (!req.body.assistanceName) {
        res.status(400)
        throw new Error('ใส่ assistanceName ด้วย')
    }
    if (!req.body.assistanceRole) {
        res.status(400)
        throw new Error('ใส่ assistanceRole ด้วย')
    }
    if (!req.body.address) {
```

ในส่วนนี้คือ Function putRequest

```
const putRequest = asyncHandler(async (req, res) => {

    const request = await Request.findById(req.params.id)
    if (!request) {
        res.status(400)
        throw new Error('request id not found')
    }

    const updatedrequest = await Request.findByIdAndUpdate(req.params.id, req.body, { new: true })
    res.status(200).json(updatedrequest)
})
```

ในส่วนนี้คือ Function deleteRequest

```
const deleteRequest = asyncHandler(async (req, res) => {
    const request = await Request.findById(req.params.id)
    if (!request) {
        res.status(400)
        throw new Error('request id not found')
    }
    const deleterequest = await Request.findByIdAndDelete(req.params.id)
    res.status(200).json({ id: req.params.id })
})
```

และสุดท้าย เป็นในส่วนของ User **ซึ่งในแต่ละ Function จะทำหน้าที่คล้ายในส่วนของ Announce**

ในภาพคือ Function getUsers และ Function getUser

```
const getUsers = asyncHandler(async (req, res) => {
    const users = await User.find().select('+password').select('+role').select('-__v').select('-createtime')
    res.status(200).json(users)
})
```

```
const getUser = asyncHandler(async (req, res) => {

    var   user
    if( mongoose.Types.ObjectId.isValid(req.params.id) ) {
        user = await User.find({'_id':req.params.id}).select('+password').select('+role').select('-__v').select('-createtime')
    }
  else{
      user = await User.find({'studentID':req.params.id}).select('+password').select('+role').select('-__v').select('-createtime')
  }
   if (user.length >0) {
      res.status(200).json(user)
   }
   else{
      res.status(404)
      throw new Error("หาไม่เจอ")
   }

})
```

ในส่วนนี้คือ Function setUser

```
const setUser = asyncHandler(async (req, res) => {

    const { email,studentID, password } = req.body


    if (email) {
        var oldUser = await User.findOne({ email })
        if (oldUser) {
            res.status(400)
            throw new Error('email user is aleady use')
        }
        oldUser = await User.findOne({ studentID })
        if (oldUser) {
            res.status(400)
            throw new Error('studentID user is aleady use')
        }
    }
    else {
        res.status(400)
        throw new Error(' please add email value')
    }

    encryptedPassword = await bcrypt.hash(password, 10)

    if(!req.body.firstName){
        res.status(400)
        throw new Error('ใส่ firstName ด้วย')
```

ในส่วนนี้คือ Function putUser

```
const putUser = asyncHandler(async (req, res) => {
    const user = await User.findById(req.params.id)
    if (!user) {
        res.status(400)
        throw new Error('user id not found')
    }

    const updateduser = await User.findByIdAndUpdate(req.params.id, req.body, { new: true })
    res.status(200).json(updateduser)
})
```

และสุดท้ายคือ Function deleteUserByGmail และ deleteUser

```
const deleteUserByGmail = asyncHandler(async (req, res) => {
    const {email} =  req.body
    const user = await User.findOne({email})
    if (!user) {
        res.status(400)
        throw new Error('user not found')
    }
    user.remove()


    res.status(200).json({ email:email })
})
```

```
const deleteUser = asyncHandler(async (req, res) => {
    const user = await User.findById(req.params.id)
    if (!user) {
        res.status(400)
        throw new Error('user id not found')
    }
    const deleteuser = await User.findByIdAndDelete(req.params.id)
    res.status(200).json({ id: req.params.id })
})
```