**คู่มือการพัฒนาโปรแกรม Curriculum Output Prediction From Student Academic Data**

1. A picture containing text, diagram, plan, technical drawing

   Description automatically generatedDiagram ของระบบ

โครงสร้างการทำงานของระบบได้อธิบายถึงการเชื่อมต่อระหว่างส่วนต่างๆของระบบ เริ่ม

ตั้งแต่ส่วนของ Input ที่ทำหน้าที่รับข้อมูลเข้ามาจากการสำรวจ และจากกรรมการหลักสูตร แล้วเก็บไว้ในส่วนของ Database จากนั้น ส่วนของ Prepare Data จะนำข้อมูลจากส่วนของ Database มาเตรียมข้อมูลให้พร้อม เมื่อทำเสร็จแล้ว จะทำการส่งกลับไปอัปเดตยัง Database ส่วนของ Process and Prediction จะนำข้อมูลที่ได้ไปเข้าอัลกอริทึมเพื่อ Processผลลัพธ์ออกมาแสดงผลบน Web Application ซึ่งในส่วนการทำงานของระบบแบ่งได้เป็น Backend และ Frontend ซึ่งในส่วนของ Backend นั้นจะรับผิดชอบการทำงานในส่วนของการนำ Input ไป process และติดต่อกับส่วนของ Database รวมถึงส่วนของ Prepare Data และ Model Job Classification และ Similarity Model และการทำงานของ Elective Subject แล้วนำผลลัพธ์การประมวลผลส่งออกไปยัง Frontend ในส่วนของ Frontend นั้นจะรับผิดชอบในการนำผลลัพธ์การประมวลผลที่ Backend ส่งมานำมาแสดงเป็นหน้าใช้งานที่สวยงามและเข้าใจง่ายตาม Diagram ในส่วนของ Output on Web Application และรับผิดชอบในส่วนการแสดงหน้ารับข้อมูลจากผู้ใช้งานแล้วติดต่อไปยัง Backend ในส่วนของ Input

1. โครงสร้างของ Directory ในระบบ
   1. A screenshot of a computer

      Description automatically generatedBack End
   2. Front End

A screenshot of a computer

Description automatically generated

1. Back End Description
   1. ประกอบไปด้วย 2 app 1 backend 2 recommend
   2. Docker File

3.2.1 docker-compose.yml ใช้งานในการ config ค่าต่างๆใน compose เช่น image ในของ DB

3.2.2 Dockerfile ใช้งานสำหรับการ config ค่า version ของ python และการติดตั้ง lib

3.3. requirements.txt ใช้งานสำหรับการ list lib ที่จำเป็นต่อระบบ

3.4. manage.py ใช้สำหรับการ run application

3.5. โครงสร้างของ App

3.5.1. \_\_init\_\_.py ใช้สำหรับ init app เสริมที่จำเป็นเช่น Mysql

3.5.2. asgi.py ใช้สำหรับการตั้งค่าและรัน ASGI (Asynchronous Server Gateway Interface) server สำหรับการรับและตอบกลับ HTTP requests แบบ asynchronous

3.5.3. setting.py ใช้สำหรับการตั้งค่า App, Middleware, Database, CORS ในระบบ

3.5.4. urls.py ใช้ในการกำหนดค่า url ไปยัง function ที่อยู่ใน file view.py

3.5.6. view.py ใช้สำหรับการสร้าง fuction ให้กับระบบ

3.6.7. ใช้สำหรับการตั้งค่าและรัน WSGI (Web Server Gateway Interface) server สำหรับการรับและตอบกลับ HTTP requests แบบ synchronous

3.6. data ใช้สำหรับเก็บ log, rollback, migrate file ของ Database Maria DB

3.7. การทำงานของแต่ละ Function ใน views.py และ url ที่ถูกกำหนดตาม Usecase ของระบบ

3.7.1. Use Case ดูผลลัพธ์สายงานของบัณฑิตย้อนหลัง

3.7.1.1. URL : <http://localhost:8000/getCareerResult/>

3.7.1.2. Function 1 : get\_career\_result

3.7.1.2.1. Parameter : request

3.7.1.2.2. Return : Json(res{“message”: dict, “status”: int})

**A picture containing text, diagram, plan, technical drawing

Description automatically generated** 3.7.1.3. Sequence Diagram :

3.7.1.4. Description : function จะรอรับ request ที่เป็น Method GET แล้วจะทำการ query ข้อมูลจากฐานข้อมูลมาแล้ว Filter ตาม Body ที่ได้จาก request คือ “curriculum” หรือหลักสูตร แล้วจะนำข้อมูลสร้างเป็น Dict แล้วส่งออกในรูปแบบ Json

3.7.2. Use Case คาดการณ์ผลลัพธ์สายงานที่ทำได้ของบัณฑิตในอนาคต

3.7.2.1.URL : <http://localhost:8000/req_pred_many>

3.7.2.2. Function 1 : reqPredict\_career\_manyUser

3.7.2.2.1. Parameter : request

3.7.2.2.2. Return : Json(res{“message”: dict, “status”: int})

3.7.2.3 Fucntion 2 : transfromGrade

3.7.2.3.1. Parameter : dataframe

3.7.2.3.2. Return : dataframe

` 3.7.2.4 Fucntion 3 : transpost\_df

3.7.2.4.1. Parameter : dataframe

3.7.2.4.2. Return : dataframe

**Diagram

Description automatically generated** 3.7.2.5. Sequence Diagram :

3.7.2.6. Description : function จะรอรับ request ที่เป็น Method POST แล้วจะทำการดึงค่าต่างๆ ที่จำเป็นจาก Body คือ “curriculum” หรือ หลักสูตร, “year” หรือปีที่นักศึกษาลงเรียนโดยทั้งคู่จะมีค่า Default อยู่เมื่อไม่ส่ง Body มาคือ 'วิศวกรรมคอมพิวเตอร์' และ '2562' ตามลำดับ โดยหลังจากนั้น function จะทำการคำนวณปีการศึกษาของหลักสูตรจากตัวแปล “year” แล้วจะนำเอาค่าดังกล่าวและ “curriculum” เพื่อไป query model ในฐานข้อมูลโดยเลือก version ล่าสุดหลังจากนั้น function จะทำการ query ข้อมูลของนักศึกษาในระบบแล้วทำการแปลงให้พร้อมต่อการ predict โดยใช้ function ที่ 2 และ 3 โดย function 2 จะทำการ map เกรดในรูปของตัวอักษรไปเป็น float และ function 3 จะทำการ transpose dataframe และจะส่งเข้าไป predict แล้วทำการ map ค่าผลลัพธ์ออกมาเป็น Dict แล้วส่งออกในรูปแบบ Json

3.7.3. Use Case โหลดแบบฟอร์มสำหรับการวิเคราะห์ข้อมูล

3.7.3.1. URL : <http://localhost:8000/reqAna>

3.7.3.2. Function 1 : csv\_template\_generator

3.7.3.2.1. Parameter : request

3.7.3.2.2. Return : file content

**Diagram

Description automatically generated** 3.7.3.3. Sequence Diagram

3.7.3.4. Description : function จะรอรับ request ที่เป็น Method POST แล้วจะทำการดึงค่าต่างๆ ที่จำเป็นจาก Body คือ “curriculum” หรือ หลักสูตร, “year” หรือปีที่นักศึกษาลงเรียนโดยทั้งคู่จะมีค่า Default อยู่เมื่อไม่ส่ง Body มาคือ 'วิศวกรรมคอมพิวเตอร์' และ '2562' ตามลำดับ โดยหลังจากนั้น function จะทำการคำนวณปีการศึกษาของหลักสูตรจากตัวแปล “year” แล้วจะนำเอาค่าดังกล่าวและ “curriculum” เพื่อไป query ข้อมูลรายวิชา แล้วสร้าง file csv ขึ้นจากข้อมูลนั้นแล้วนำส่งออกไป

3.7.4. Use Case อัปโหลดแบบฟอร์มสำหรับการวิเคราะห์ข้อมูล

3.7.4.1. URL : <http://localhost:8000/reqPredict>

3.7.4.2. Function 1 : gradeUploader

3.7.4.2.1. Parameter : request

3.7.4.2.2. Return : Json(res{“message”: dict, “status”: int})

3.7.4.3. Function 2 : reqPredictPerUser\_Production

3.7.4.3.1. Parameter : dataframe, student\_id(str), curriculum(str), year(str)

3.7.4.3.2. Return : list of dictionaries

3.7.4.4. Function 3 : numerical\_to\_alphabetical\_grade

3.7.4.4.1. Parameter : grade(float)

3.7.4.4.2. Return : grade(str)

3.7.4.5. Function 4 : transfromGrade (รายละเอียดตามข้อที่ 3.7.2.3)

3.7.4.6. Function 5 : generate\_data\_set

3.7.4.6.1. Parameter : curriculum(str), year(str)

3.7.4.6.2. Return : dataframe

3.7.4.7. Function 6 : transpost\_df (รายละเอียดตามข้อที่ 3.7.2.4)

3.7.4.8. Function 7 : train\_rec\_model

3.7.4.8.1. Parameter : dataframe

3.7.4.8.2. Return : model(sci-kit-learn\_object)

3.7.4.9. Function 8 : prediction\_grade\_user

3.7.4.9.1. Parameter : model(sci-kit-learn\_object), student\_id(str), selected\_value(list)

3.7.4.9.2. Return : dict

**Diagram

Description automatically generated** 3.7.4.10. Sequence Diagram

3.7.4.11. Description : function 1 จะรอรับ request ที่เป็น Method POST หลังจากนั้นจะทำการนำไฟล์ที่อยู่ใน body แล้วทำการแปลงเป็น dataframe หลังจากตรวจสอบความถูกต้องของไฟล์แล้วจะทำการแกะ student\_id, curriculum, year ออกมาจากไฟล์แล้วส่ง dataframe รวมทั้ง parameter ทั้งหมดเข้าไปใน function 2 หลังจากนั้นตัว function จะทำการส่ง dataframe ไปแปลงค่า grade ผ่าน function 4 แล้วจะทำการสร้าง dataset ขึ้นมาจาก parameter curriculum, year ผ่าน function 5 หลังจากนั้นจะทำการรวม dataframe ที่เกิดขึ้นจาก function 5 และ dataframe ที่รับเข้ามาตอนแรกแล้วส่งไปใน function 7 เพื่อสร้าง model ขึ้นมาในเวลาเดียวกันจะทำการ transpose dataframe ที่รับเข้ามาตอนแรกผ่าน function 6 ต่อมาจะนำ dataset ไปเรียกผล prediction grade ผ่าน function 8 ในเวลาเดียวกันจะทำการไป query model career แล้วนำเอา dataframe ที่ transpose แล้วไปหาผลลัพธ์อาชีพ หลังจากได้ผลลัพธ์เกรดมาจะนำผลลัพธ์มารวมกันแล้วส่งกลับไป function 1 แล้วสร้างเป็น JSON แล้วส่งค่าออกไป

3.7.5. Use Case แนะนำวิชาเลือกภาคให้แก่นักศึกษา

3.7.5.1. URL : [http://localhost:8000/recommendSubject](http://localhost:8000/reqPredict)

3.7.5.2. recommendSubject

3.7.5.2.1.Parameter : request

3.7.5.2.2. Return : Json(res{“message”: dict, “status”: int})

**Diagram

Description automatically generated** 3.7.5.3. Sequence Diagram

3.7.5.4. Description : function 1 จะรอรับ request ที่เป็น Method POST หลังจากนั้นจะทำการนำค่าที่อยู่ body key, year มาทำการ query ข้อมูลรายวิชาในฐานข้อมูลแล้วส่งออกในรูปของ JSON

3.7.6. Use Case ให้ข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร

3.7.6.1. URL : [http://localhost:8000/fileUpload](http://localhost:8000/reqPredict)/<id>

3.7.6.2. Function 1 : csv\_upload

3.7.6.2.1. Parameter : request, id(int), type\_data(str)

3.7.6.2.2. Return : Json(res{“message”: str, “status”: int})

**Diagram

Description automatically generated** 3.7.6.3. Sequence Diagram

3.7.6.4. Description : function 1 จะรอรับ request ที่เป็น Method POST เพื่อรองรับการอัปโหลดไฟล์และ Method PUT เพื่อรองรับการแก้ไขไฟล์โดยในส่วน POST function 1 จะทำการนำข้อมูลใน body หรือไฟล์ที่อัปโหลดเข้ามาหลังจากนั้นจะทำการแปลงเป็น dataframe แล้วแปลงให้อยู่ในรูป pickle field หรือ binary แล้วเก็บลงฐานข้อมูล ส่วน PUT จะทำการไป query ไฟล์ตาม ID ที่อยู่ใน URL หลังจากนั้นจะใช้ข้อมูลใน body มาแก้ไขไฟล์แล้วทำการ save ในรูปแบบเดิม โดยทั้ง 2 Method จะ Return ผลลัพธ์การทำงานของ Function ออกเป็น JSON

3.7.8. Use Case ลบข้อมูลนักศึกษาโดยไม่ระบุตัวตนและข้อมูลหลักสูตร

3.7.8.1. URL : [http://localhost:8000/getFile/<id>](http://localhost:8000/getFile/%3cid%3e)

3.7.8.2. Function 1 : file\_api

3.7.8.2.1. Parameter : request, id(int)

3.7.8.2.2. Return : Json(res{“message”: str, “status”: int})

**Diagram

Description automatically generated** 3.7.8.3. Sequence Diagram

3.7.8.4. Description : function 1 จะรอรับ request ที่เป็น Method GET เพื่อรองรับการเรียกข้อมูลไฟล์ ส่วนในของ Method DELETE จะทำการนำ ID ที่อยู่ใน URL มาทำการ query หาไฟล์แล้วทำการแปลงค่า field del\_flag เป็น 1 แล้วจะ Return ผลลัพธ์การทำงานของ Function ออกเป็น JSON

3.7.9. Use Case ล็อกอินเข้าใช้งาน

3.7.9.1. URL : <http://localhost:8000/signin>

3.7.9.2. Class 1 : LoginUser

3.7.9.2.1 Method 1 : post

3.7.9.2.1.1. Parameter : request, args, kwargs

3.7.9.2.1.2. Return : Json(res{“message”: str, “status”: int})

**Diagram

Description automatically generated** 3.7.9.3. Sequence Diagram

3.7.9.4. Description : method 1 จะรองรับ username, password จาก body ของ request หลังจากนั้นจะทำการ query หา user เมื่อเจอ user แล้วจะทำการ map pwd ที่รับเข้ามากับ pwd ที่เข้ารหัสในฐานข้อมูลแล้วจะ Return ผลลัพธ์การทำงานของ Function ออกเป็น JSON

A screenshot of a computer program

Description automatically generated with low confidence 3.8. การออกแบบฐานข้อมูล

3.8.1. ตาราง Rec\_User

เป็นตารางที่ใช้เก็บข้อมูลผู้ใช้งานในระบบ โดยจะมี Attribute ดังนี้

3.8.1.1. user\_id : UUID สำหรับเก็บ id ของผู้ใช้

3.8.1.2. username : VARCHAR สำหรับเก็บ username ของผู้ใช้

3.8.1.3. password : VARCHAR สำหรับเก็บ password ของผู้ใช้

3.8.2 ตาราง Subject\_Data

เป็นตารางที่ใช้เก็บข้อมูลหลักสูตรวิชา โดยจะมี Attribute ดังนี้

3.8.2.1. subject\_id : UUID สำหรับเก็บ id ของรหัสวิชา

3.8.2.2. subject\_name\_thai : VARCHAR สำหรับเก็บชื่อรายวิชาภาษาไทย

3.8.2.3. subject\_name\_eng : VARCHAR สำหรับเก็บชื่อรายวิชาภาษาอังกฤษ

3.8.2.4 abstract : VARCHAR สำหรับเก็บบทคัดย่อรายวิชา

3.8.2.5. subject\_key : VARCHAR สำหรับเก็บ Keyword ของรายวิชาเลือกภาค

3.8.2.6. subject\_class : VARCHAR สำหรับเก็บหมวดหมู่ของแต่ละวิชา

3.8.2.7. year : VARCHAR สำหรับเก็บปีของเล่มหลักสูตรวิชา

3.8.3 ตาราง Student\_Grade

เป็นตารางที่ใช้เก็บข้อมูลเกรดนักศึกษาโดยจะมี Attribute ดังนี้

3.8.3.1. student\_id : UUID สำหรับเก็บ id encrypt ของนักศึกษา

3.8.3.2. subject\_id : VARCHAR สำหรับเก็บ id ของรหัสวิชา

3.8.3.3. grade : VARCHAR สำหรับเก็บเกรดแต่ละรายวิชา

3.8.3.4. semester : VARCHAR สำหรับเก็บเทอมที่ลงเรียนรายวิชาของนักศึกษา

3.8.3.5. year : VARCHAR สำหรับเก็บปีที่ลงเรียนรายวิชาของนักศึกษา

3.8.4 ตาราง Student\_Data

เป็นตารางที่ใช้เก็บข้อมูลนักศึกษาโดยจะมี Attribute ดังนี้

3.8.4.1. student\_id : UUID สำหรับเก็บ id encrypt ของนักศึกษา

3.8.4.2. curriculum : VARCHAR สำหรับเก็บรายชื่อหลักสูตรที่เรียนของนักศึกษา

3.8.4.3. status : VARCHAR สำหรับเก็บสถานะการเป็นนักศึกษา

3.8.4.4. career : VARCHAR สำหรับเก็บรายชื่อตำแหน่งอาชีพแรกของนักศึกษา

3.8.4.5. start\_year : VARCHAR สำหรับเก็บปีการศึกษาของนักศึกษา

3.8.4.6. curriculum\_year : VARCHAR สำหรับเก็บปีของหลักสูตรของนักศึกษา

3.8.5 ตาราง CSV\_File

เป็นตารางที่ใช้เก็บข้อมูลไฟล์ CSV โดยจะมี Attribute ดังนี้

3.8.5.1. data\_id : UUID สำหรับเก็บ id ของไฟล์ CSV

3.8.5.2. name : VARCHAR สำหรับเก็บชื่อของไฟล์ CSV

3.8.5.3. upload\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปโหลดไฟล์ CSV

3.8.5.4. update\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปเดตไฟล์ CSV

3.8.5.5. del\_flag : VARCHAR สำหรับเก็บสถานะไฟล์ CSV

3.8.5.6. type\_data : VARCHAR สำหรับเก็บประเภทข้อมูลของ Data

3.8.5.7. file : BINARY สำหรับเก็บไฟล์ CSV

3.8.6 ตาราง SurpriseModel

เป็นตารางที่ใช้เก็บ Surprise Model โดยจะมี Attribute ดังนี้

3.8.6.1. model\_id : UUID สำหรับเก็บ id ของไฟล์ Model

3.8.6.2. name : VARCHAR สำหรับเก็บชื่อของ Model

3.8.6.3. curriculum : VARCHAR สำหรับเก็บรายชื่อหลักสูตรที่เรียนของนักศึกษา

3.8.6.4. year : VARCHAR สำหรับเก็บปีของหลักสูตรของชุดข้อมูลที่ใช้สร้าง

3.8.6.5. rmse : VARCHAR สำหรับเก็บค่าความแม่นยำของ Model

3.8.6.6. create\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่สร้าง Model

3.8.6.7. update\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปเดต Model

3.8.6.8. del\_flag : VARCHAR สำหรับเก็บสถานะ Model

3.8.6.9. type\_pred : VARCHAR สำหรับเก็บประเภทของ Model

3.8.6.10. rec\_model : BINARY สำหรับเก็บตัว Model

3.8.7 ตาราง CareerModel

เป็นตารางที่ใช้เก็บ Job Classification Model โดยจะมี Attribute ดังนี้

3.8.7.1. model\_id : UUID สำหรับเก็บ id ของไฟล์ Model

3.8.7.2. name : VARCHAR สำหรับเก็บชื่อของ Model

3.8.7.3. curriculum : VARCHAR สำหรับเก็บรายชื่อหลักสูตรที่เรียนของนักศึกษา

3.8.7.4. year : VARCHAR สำหรับเก็บปีของหลักสูตรของชุดข้อมูลที่ใช้สร้าง

3.8.7.5. accuracy : VARCHAR สำหรับเก็บค่าความแม่นยำของ Model

3.8.7.6. create\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่สร้าง Model

3.8.7.7. update\_date : TIMESTAMP สำหรับเก็บเวลาและวันที่อัปเดต Model

3.8.7.8. del\_flag : VARCHAR สำหรับเก็บสถานะ Model

3.8.7.9. train\_set\_cols : BINARY สำหรับเก็บ Columns ของชุดข้อมูลที่ใช้สร้าง Model

3.8.7.10. career\_model : BINARY สำหรับเก็บตัว Model

1. Front End Description

4.1 node\_modules สำหรับเก็บ dependency ข้างในจะมี folder อื่น ๆ เพื่อเขียนโปรแกรมแล้วจะใช้งาน library หรือ package ต้องทำการ import มาจาก folder นี้

4.2 public สำหรับเก็บ index.html และ assets พวก icon

4.3 src เป็น folder หลักที่จะใช้เขียนโปรแกรม

4.2.1 Assests เป็นโฟลเดอร์สำหรับเก็บไฟล์รูปภาพ

4.2.2 Component เป็นโฟลเดอร์เก็บส่วนประกอบ หรือองค์ประกอบที่ตัวหน้าเว็บทำการเรียกไปใช้ ซึ่งมีปุ่มต่างๆ ปุ่มเพิ่ม ลบ แก้ไข มีตาราง Table รวมไปถึง navbar และ footer

4.2.3 hooks เป็น useContext Hook API ที่ช่วยในการจัดการกับ global state เป็นการไปบอก component ที่ชั้น parent หรือ top level component ว่าให้ค่า context เป็นอย่างไร แล้ว component ลูกที่อยู่ใต้มันจะสามารถนำค่า context ไปใช้ได้ ไว้จัดการ state ของการเพิ่ม/เเก้ไข/ลบข้อมูล

4.2.4 Pages หน้า .js สำหรับพัฒนาหน้าเว็บต่างๆ

4.2.4.1 outputCareer เก็บ index.js หน้าสำหรับพัฒนาหน้าโชว์ผลลัพธ์อาชีพของนักศึกษา โดยส่วน StaticCareer.js ผลลัพธ์สถิติอาชีพในอดีตจะมีการใช้ Axios ดึงจาก API [http://localhost:8000/getCareerResult/ ส่วน](http://localhost:8000/getCareerResult/%20ส่วน) PredictCareer.js ผลลัพธ์พยากรณ์อาชีพจะมีการใช้ Axios ดึงจาก <http://localhost:8000/req_pred_many/>

4.2.4.2 dataUpload.js หน้าสำหรับพัฒนาหน้าอัปโหลดไฟล์จากกรรมการหลักสูตร โดยดึงข้อมูลไฟล์จากฐานข้อมูล ผ่าน API http://localhost:8000/getFile/ และกรรมการหลักสูตรสามารถอัปโหลดไฟล์เพิ่มได้ผ่าน API [http://localhost:8000/fileUpload/](http://localhost:8000/req_pred_many/)

4.2.4.3 home.js หน้าสำหรับพัฒนาหน้าหลัก โดยจะมีลิงก์ที่เชื่อมโยงไปยัง Google Form

4.2.4.4 index.js หน้า main หลัก โดยเรียกฟังก์ชัน Home จาก home.js มาใช้ในการแสดงผล

4.2.4.5 login.js หน้าสำหรับพัฒนาหน้าเข้าสู่ระบบ โดยจะทำการเข้าสู่ระบบผ่าน API [http://localhost:8000/signin/](http://localhost:8000/req_pred_many/)

4.2.4.6 predictStudent.js หน้าสำหรับพัฒนาหน้าพยากรณ์เกรดและอาชีพของนักศึกษา โดยจะมีการโหลดฟอร์มจาก API <http://localhost:8000/reqAna/> และมีการอัปโหลดเพื่อทำการพยากรณ์ผ่าน API <http://localhost:8000/reqPredict/> หลังจากกดปุ่มพยากรณ์ จะเรียกใช้ฟังก์ชัน handleSubmit และทำการ set ค่า ShowPredicts เป็น true เพื่อทำการเรียกใช้ฟังก์ชัน ResultPredict จาก resultPredict.js หน้าสำหรับพัฒนาหน้าแสดงผลลัพธ์จากการพยากรณ์

4.2.4.7 recommendSubject.js หน้าสำหรับพัฒนาหน้าแนะนำวิชาเลือก โดยจะมีการเรียกคำสำคัญของแต่ละวิชาเลือกภาคออกมาโชว์บนหน้าเว็บผ่าน API [http://localhost:8000/ keysubject/](http://localhost:8000/%20keysubject/) หลังจากกดปุ่มแนะนำวิชาเลือกภาค จะเรียกใช้ฟังก์ชัน handleSubmit และทำการเรียกวิชาเลือกภาคที่ฟิลเตอร์ตรงกับการเลือกผ่าน API [http://localhost:8000/ recommendSubject](http://localhost:8000/%20keysubject/) และทำการ set ค่า ShowSubjects เป็น true เพื่อทำการเรียกใช้ฟังก์ชัน RecommendedSubjectsList จาก recommendedSubjectsList.js หน้าสำหรับพัฒนาหน้ารายชื่อวิชาเลือกภาค

4.2.5 App.css กับ index.css ใช้ในการตกแต่งสิ่งที่ทุก component

4.2.6 App.js เป็นไฟล์หลักที่จะใช้รวม component ทั้งหมด

4.2.7 index.js ไฟล์ที่จะนำ App component จาก App.js ไป render

4.2.8 App.test.js เป็นไฟล์ทำ Unit test สำหรับ App.js

4.4 .gitignore ใช้สำหรับใช้งาน Git ใน React Project Git จะไม่ commit ไฟล์หรือโฟลเดอร์ ที่ระบุใน .gitignore

4.5 package.json เป็นไฟล์ประเภท JSON ที่บอกข้อมูลต่าง ๆ ของ project ที่ใช้ node package manager โดยบอกถึง dependencies, version ของ dependencies และอื่น ๆ ที่ใช้ใน React app

4.6 README.md เป็นไฟล์ประเภท markdown ที่ใช้อธิบายเกี่ยวกับ create-react-app

4.7 yarn.lock เป็นไฟล์ที่ใช้ lock version ของ dependencies สำหรับคนที่ต้องการใช้ yarn แทน npm เพื่อใช้ version นี้ทุกครั้ง เวลาส่งมอบให้ team อื่นทำต่อ ทำให้ไม่เกิดปัญหา version ของ dependencies ไม่ตรงกัน