



Web-based IDE

โดย

นาย ธีรช ประสิทธิ์เวช

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์
ปีการศึกษา 2564
ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

Web-based IDE

โดย

นาย ธีรช ประสิทธิ์เวช

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์
ปีการศึกษา 2564
ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

Web-based IDE

By

Mr. Teerat Prasitwet

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING
IN COMPUTER ENGINEERING
FACULTY OF ENGINEERING
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2021
COPYRIGHT OF THAMMASAT UNIVERSITY**

มหาวิทยาลัยธรรมศาสตร์

คณะวิศวกรรมศาสตร์

โครงการ

ของ

นาย ธีรช ประสิทธิ์เวช

เรื่อง

Web-based IDE

ได้รับการตรวจสอบและ อนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

เมื่อวันที่ 20 มิถุนายน พ.ศ. 2565

อาจารย์ที่ปรึกษาโครงการ

ร.ล

(รองศาสตราจารย์ ดร. วีรชัย อโณทัยไพบูรณ์)

หัวหน้าภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

พิศาล

(ผู้ช่วยศาสตราจารย์ ดร. พิศาล แก้วประภา)

หัวข้อโครงงาน	Web-based IDE
ชื่อผู้เขียน	นาย ชีรัช ประสิทธิ์เวช
ชื่อปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา/คณะ	วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์
อาจารย์ที่ปรึกษาโครงงาน	รองศาสตราจารย์ ดร. วีรชัย อโณทัยไพบลุย
ปีการศึกษา	2564

บทคัดย่อ

โครงงานเรื่อง Web-based IDE ในรายวิชาโครงงานของนักศึกษาระดับปริญญาตรี ชั้นปีที่ 4 มีวัตถุประสงค์เพื่อสร้างเว็บไซต์ที่ทำให้ผู้ใช้สามารถเข้าใช้งาน IDE ได้ทันทีผ่านการเข้าใช้งานเว็บไซต์ในฝั่ง client โดย environment ของตัว IDE จะเป็น environment ของเครื่องคอมพิวเตอร์ฝั่ง server ที่มีการติดตั้งภาษาที่ต้องการใช้ในการรันโค้ดเอาไว้แล้ว ผู้ใช้จึงสามารถศึกษาและพัฒนาซอฟต์แวร์ในภาษาต่างๆได้ โดยไม่จำเป็นต้องติดตั้งภาษานั้นๆไว้ในเครื่องของตนเอง และสร้างเว็บไซต์ที่ทำให้ผู้ใช้สามารถดูผลลัพธ์การตรวจโจทย์โปรแกรม markdown

ผลการดำเนินงานที่ได้คือ ได้ Web-based IDE ที่ฝั่ง client มี terminal ที่มีการทำงานอยู่ใน environment ของตัวเครื่องคอมพิวเตอร์ฝั่ง server โดยตัว terminal นั้นสามารถทำงานได้ในรูปแบบเดียวกับ powershell ของตัวเครื่อง server เอง มีหน้าต่างเขียนโค้ดที่สามารถไฮไลต์โค้ดได้ตามภาษาที่ผู้ใช้เลือก โดยมีให้เลือกทั้งหมด 2 ภาษา คือ Python และ Java สามารถบันทึกโค้ดในหน้าต่างเขียนโค้ดให้กลายเป็นไฟล์ในภาษานั้นๆลงบนเครื่อง server ได้ ผู้ใช้จะสามารถรันโค้ดที่ตนเองบันทึกไปได้โดยการรันผ่าน Web-based Terminal และได้ Web-based IDE ที่สามารถส่งโจทย์โปรแกรม markdown ของกลุ่มโครงงานที่มีอาจารย์ที่ปรึกษาร่วมกัน ไปตรวจสอบที่ API ของกลุ่มโครงงานที่มีอาจารย์ที่ปรึกษาร่วมกัน แล้วนำผลลัพธ์การตรวจสอบมาแสดงผลบนหน้าเว็บไซต์ได้

คำสำคัญ : Web-based, IDE, terminal, client, server, Python, Java, markdown

Title	Web-based IDE
Author	Mr. Teerat Prasitwet
Degree	Engineering
Major	Electrical and Computer Faculty of Engineering Thammasat University
Advisor	Associate Prof. Dr. Weerachai Anotaiapaiboon
Academic Year	2021

Abstract

Web-based IDE project in Project course of undergraduate student year 4, has a purpose to create website that help users can access to use IDE immediately, through website on client side. IDE environment will be environment of PC in server side that already installed language that users want to use to run their source code, so the users can study and develop software in language that they don't need to install in their PC. And create website that help users can checking correctness of markdown.

The operating result is Web-based IDE that client side has terminal that works in server side environment, and works in the same format as server side powershell. Web-based IDE has source code editor that can highlight the code in form of language that users selected, which available 2 languages, namely Python and Java. Users can save source code in source code editor to become a file of that language in server side PC, and run their source code from Web-based Terminal. And Web-based IDE can send markdown of project group with advisor together to API of project group with advisor together then bring result to display on website.

Keywords: Web-based, IDE, terminal, client, server, Python, Java, markdown

กิตติกรรมประกาศ

โครงการเรื่อง Web-based IDE นี้ สามารถดำเนินการจนประสบความสำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์และสนับสนุนเป็นอย่างดีจาก รองศาสตราจารย์ ดร. วีรชัย อโณทัยไพบูลย์ อาจารย์ที่ปรึกษาโครงการ ที่ได้กรุณาให้การเสนอแนะหัวข้อโครงการ คำปรึกษา แนวทางในการทำโครงการ และปรับปรุงแก้ไขข้อบกพร่องต่างๆ จนกระทั่งโครงการนี้สำเร็จเรียบร้อยด้วยดี ผู้จัดทำโครงการขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบคุณ อ.นาวัน สมญาติ และ ผศ.ดร.พิศาล แก้วประภา อาจารย์กรรมการคุมสอบโครงการ ที่ได้ให้คำวิจารณ์และติชมโครงการในระหว่างการสอบ ซึ่งเป็นประโยชน์อย่างมากในการนำมาปรับปรุงแก้ไขโครงการให้ดียิ่งขึ้น

และขอขอบคุณเพื่อนๆในคณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ได้ให้ความช่วยเหลือในการแก้ไขปัญหาที่เกิดขึ้นระหว่างการทำโครงการ

ท้ายนี้ ผู้จัดทำโครงการหวังว่าโครงการชิ้นนี้จะเป็นประโยชน์ต่อผู้ใช้ที่ต้องการใช้งาน IDE บนเว็บไซต์ และรองศาสตราจารย์ ดร. วีรชัย อโณทัยไพบูลย์ อาจารย์ที่ปรึกษาโครงการ ที่อาจนำโครงการชิ้นนี้ไปใช้ประโยชน์ได้ในภายภาคหน้า

ผู้จัดทำโครงการ

นายธีรวัช ประสิทธิ์เวช

สารบัญ

หน้า

บทคัดย่อ	(1)
Abstract	(2)
กิตติกรรมประกาศ	(3)
สารบัญ	(4)
สารบัญรูป	(7)
สารบัญตาราง	(8)
สัญลักษณ์และคำย่อ	(9)
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตการดำเนินงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	2
1.6 ตารางการดำเนินงาน	3
บทที่ 2 ทฤษฎี หรืองานที่เกี่ยวข้อง	4
2.1 IDE	4
2.1.1 Source Code Editor	5
2.1.2 Compiler หรือ Interpreter	5
2.1.3 IDE ที่ได้รับความนิยมในปัจจุบัน	5
2.2 Web-based Application	7
2.2.1 การพัฒนา Web-based Application	7
2.2.2 จุดเด่น Web-based Application	7
2.3 Repl.it	8
2.4 Xterm.js	9
2.4.1 Features	9
2.5 Ace	10

2.5.1 Features	10
2.6 Node.js	11
2.7 Socket.io	11
บทที่ 3 การดำเนินงาน/วิธีวิจัย	12
3.1 ศึกษาการใช้งานเครื่องมือที่จำเป็น	12
3.1.1 Repl.it	12
3.1.2 Xterm.js	12
3.1.3 Ace	12
3.1.4 Node.js	12
3.1.5 Socket.io	12
3.2 ดาวน์โหลดและติดตั้งเครื่องมือที่จำเป็น	12
3.2.1 โปรแกรมที่ใช้	12
3.2.2 Package ที่ใช้	12
3.3 สร้าง Web-based Terminal	13
3.3.1 สร้างส่วน Server	13
3.3.2 สร้างส่วน Client	13
3.4 สร้าง Web-based IDE	14
3.4.1 เพิ่มส่วน Server	14
3.4.2 เพิ่มส่วน Client	14
บทที่ 4 ผลการดำเนินงานและอภิปรายผล	15
4.1 หลักการทำงานของ Web-based IDE	15
4.1.1 เริ่มต้นการทำงาน Server	15
4.1.2 เริ่มต้นการทำงาน Client	16
4.1.3 เลือกเปลี่ยนภาษา	16
4.1.4 กดปุ่ม save code	16
4.1.5 กดปุ่ม check code	16
4.2 ผลการสร้างตัว Web-based Terminal	17
4.2.1 การทำงานของ terminal	17

	(6)
4.3 ผลการสร้างตัว Web-based IDE	20
4.3.1 การทำงานของ IDE	20
บทที่ 5 สรุปผลการดำเนินงาน อุปสรรค และการพัฒนาในอนาคต	22
5.1 สรุปผลการดำเนินงาน	22
5.2 อุปสรรค	22
5.3 การพัฒนาในอนาคต	22
บรรณานุกรม	23
ภาคผนวก	24

สารบัญรูป

รูปที่	หน้า
2.1 หน้า UI IDE ของ Repl.it	8
2.2 หน้าเว็บไซต์ Xterm.js	9
2.3 หน้าเว็บไซต์ Ace	10
2.4 หน้าเว็บไซต์ Node.js	11
4.1 Sequence Diagram แสดงหลักการทำงานของ Web-based IDE	15
4.2 หน้าเว็บไซต์เริ่มต้นของ Web-based IDE	16
4.3 แสดงตัวอย่าง path และการใช้งาน command ใน terminal	17
4.4 แสดงตัวอย่างการเข้าถึง path อื่นๆในเครื่อง server	18
4.5 แสดงตัวอย่างโค้ดรับ input ภาษา Python ในไฟล์ code.py ผ่าน Visual Studio Code	18
4.6 แสดงตัวอย่างการรันไฟล์ การรับ input และแสดงผล output ไฟล์ภาษา Python ผ่าน terminal	18
4.7 แสดงตัวอย่างโค้ดรับ input ภาษา Java ในไฟล์ code.java ผ่าน Visual Studio Code	18
4.8 แสดงตัวอย่างการรันไฟล์ การรับ input และแสดงผล output ไฟล์ภาษา Java ผ่าน terminal	19
4.9 แสดงตัวอย่างบัคการแสดงผลตัวอักษรตัวแรกของ input ซ้ำหลายครั้ง	19
4.10 แสดงตัวอย่างการไฮไลต์โค้ดภาษา Python ในหน้าต่างเขียนโค้ด	20
4.11 แสดงตัวอย่างการไฮไลต์โค้ดภาษา Java ในหน้าต่างเขียนโค้ด	20
4.12 แสดงไฟล์ภาษา Python และ Java ที่ถูกบันทึกลงในโฟลเดอร์ storage	20
4.13 แสดงตัวอย่างการไฮไลต์ไวยากรณ์โปรแกรม markdown ภาษา Python ในหน้าต่างเขียนโค้ด	21
4.14 แสดงตัวอย่างการไฮไลต์ไวยากรณ์โปรแกรม markdown ภาษา Java ในหน้าต่างเขียนโค้ด	21
4.15 แสดงตัวอย่างผลลัพธ์ที่หน้าต่างแสดงผลการตรวจไวยากรณ์โปรแกรม markdown	21

สารบัญตาราง

	หน้า
ตารางที่ 1.1 การดำเนินโครงการ	3
ตารางที่ 1.2 การดำเนินโครงการ (ต่อ)	3

สัญลักษณ์และคำย่อ

สัญลักษณ์/คำย่อ	คำเต็ม/คำจำกัดความ
IDE	Integrated Development Environment
GPL	General Public License
MS	Microsoft
WWW	World Wide Web
etc	อื่นๆ
CJK	การใช้คำศัพท์ ภาษาจีน, ภาษาญี่ปุ่น และภาษาเกาหลี ร่วมกัน สำหรับ ใช้ในด้านซอฟต์แวร์และการสื่อสารสากล
IME	Input Method Editor
API	Application Programming Interface
PTY	Pseudo-terminal

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบัน เทคโนโลยีสารสนเทศนั้นมีความสำคัญต่อเกือบทุกวงการทั่วโลก เนื่องจากเทคโนโลยีสารสนเทศสามารถช่วยอำนวยความสะดวกให้กับการทำงานและการดำเนินชีวิตต่าง ๆ มากมาย อาทิเช่น

- ช่วยให้การติดต่อสื่อสารระหว่างกันเป็นไปอย่างสะดวกรวดเร็ว
- ช่วยในการจัดระบบข่าวสารจำนวนมากที่ผลิตออกมาในแต่ละวัน
- ช่วยให้เก็บสารสนเทศไว้ในรูปที่สามารถเรียกใช้ได้ครั้งแล้วครั้งเล่าอย่างสะดวก
- ช่วยเพิ่มประสิทธิภาพการผลิตสารสนเทศ เช่น ช่วยนักวิทยาศาสตร์ วิศวกร ด้วยการช่วยคำนวณ

ตัวเลขที่ยู่งยากซับซ้อนซึ่งไม่สามารถทำให้สำเร็จได้ด้วยมือ ฯลฯ

- ช่วยในการจัดระบบอัตโนมัติเพื่อการเก็บ เรียกใช้และประมวลผลสารสนเทศ
- สามารถจำลองแบบระบบการวางแผนและทำนาย เพื่อทดลองกับสิ่งที่ยังไม่เกิดขึ้น
- อำนวยความสะดวกในการเข้าถึงสารสนเทศดีกว่าสมัยก่อน ทำให้ผู้ใช้สารสนเทศมี ทางเลือกที่ดีกว่า

มีประสิทธิภาพกว่า และสามารถแข่งขันกับผู้อื่นได้ดีกว่า

- ลดอุปสรรคเกี่ยวกับเวลาและระยะทางระหว่างประเทศ

ด้วยความสำคัญของเทคโนโลยีสารสนเทศข้างต้น ทำให้เห็นได้ว่าการศึกษาด้านที่เกี่ยวกับเทคโนโลยีสารสนเทศ และการเขียนโปรแกรมใหม่ๆ เพื่อใช้งานในเทคโนโลยีสารสนเทศนั้น จะก่อให้เกิดประโยชน์กับหลายภาคส่วนในสังคมอย่างมาก

จากเหตุผลดังกล่าว ผู้จัดทำจึงสนใจที่จะพัฒนา Web-based IDE เพื่อเป็นประโยชน์ต่อผู้ที่ต้องการศึกษา และผู้ที่ต้องการพัฒนาและทดสอบการใช้งาน source code ของซอฟต์แวร์ แต่ไม่สะดวกในการติดตั้ง IDE หรือภาษาที่จำเป็นลงบนเครื่องคอมพิวเตอร์ของตนเอง สามารถเข้าใช้งาน IDE ได้ทันทีผ่านการเข้าใช้งานเว็บไซต์ โดย IDE นั้นมีความแตกต่างจากคอมไพเลอร์ หลักๆคือ IDE เป็นชุดซอฟต์แวร์ที่ประกอบด้วยเครื่องมือที่จำเป็นในการพัฒนาและทดสอบการใช้งานซอฟต์แวร์ ในขณะที่คอมไพเลอร์เป็นโปรแกรมที่แปลซอร์สโค้ดที่เขียนด้วยภาษาโปรแกรมระดับสูงให้เป็นรหัสเครื่องระดับต่ำ

1.2 วัตถุประสงค์

1. เพื่อพัฒนา Web-based IDE ที่สามารถใช้งาน terminal ได้
2. เพื่อพัฒนา Web-based IDE ที่สามารถตรวจจอยโปรแกรม markdown ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาพร้อมกันได้
3. เพื่อพัฒนา Web-based IDE ให้ผู้ที่ต้องการศึกษา พัฒนา และทดสอบการใช้งาน source code ของซอฟต์แวร์ แต่ไม่สะดวกในการติดตั้ง IDE หรือภาษาที่จำเป็นลงบนเครื่องคอมพิวเตอร์ของตนเอง ได้ใช้งาน

1.3 ขอบเขตการดำเนินงาน

1. สร้าง terminal ที่สามารถทำงานได้บนเว็บไซต์
2. สร้างหน้าต่างแสดงผลการตรวจโจทย์โปรแกรม markdown ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาพร้อมกัน โดยผลลัพธ์นี้ได้มาจากการส่งโจทย์ไปตรวจที่ API ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาพร้อมกัน
3. สร้างหน้าต่างเขียนโค้ดที่ทำงานได้ 2 ภาษา คือ Python และ Java

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาการใช้งาน Xterm.js
2. ศึกษารูปแบบการทำงานของ IDE ของ Repl.it
3. ศึกษา Ace ซึ่งเป็น code editor บนเว็บไซต์
4. ศึกษาการสร้างการทำงานระหว่าง client และ server
5. ศึกษาการใช้งาน Node.js
6. ออกแบบและพัฒนา terminal สำหรับการตอบโต้กันระหว่าง client และ server
7. ออกแบบและพัฒนา IDE สำหรับทำงานร่วมกับโจทย์โปรแกรม markdown ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาพร้อมกัน
8. ออกแบบ สร้าง และตกแต่งตัวเว็บไซต์สำหรับเป็น Web-based IDE
9. ทดสอบการทำงานของ Web-based IDE ร่วมกับโจทย์โปรแกรม markdown ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาพร้อมกัน และแก้ไขข้อผิดพลาด
10. จัดทำรายงานฉบับสมบูรณ์

1.5 ผลที่คาดว่าจะได้รับ

1. ตัวเว็บไซต์ดูเข้าใจง่าย ใช้งานง่าย
2. terminal สามารถทำงานได้บนเว็บไซต์
3. หน้าต่างเขียนโค้ดทำงานได้ทั้ง 2 ภาษา คือ Python และ Java
4. สามารถส่งโจทย์โปรแกรม markdown ไปตรวจที่ API ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาพร้อมกัน แล้วนำผลลัพธ์ของการตรวจสอบมาแสดงผลได้
5. สามารถนำ Web-based IDE มาใช้เพื่อการศึกษา พัฒนาและทดสอบการใช้งาน source code ของซอฟต์แวร์บนเว็บไซต์ได้ในอนาคต

1.6 ตารางการดำเนินงาน

แผนการดำเนินงาน	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน				ธันวาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ศึกษาการใช้งาน Xterm.js	x	x	x	x																
ศึกษารูปแบบการทำงานของ IDE ของ Repl.it			x	x	x	x														
ศึกษา Ace ซึ่งเป็น code editor บนเว็บไซต์						x	x	x												
ศึกษาการสร้างการทำงานระหว่าง client และ server								x	x	x										
ศึกษาการใช้งาน Node.js											x	x								
ออกแบบและพัฒนา terminal สำหรับการตอบโต้กันระหว่าง client และ server													x	x	x	x				
ออกแบบและพัฒนา IDE สำหรับทำงานร่วมกับไจทียโปรแกรม markdown ของกลุ่ม โครงการที่มีอาจารย์ที่ปรึกษาร่วมกัน																	x	x	x	x

ตารางที่ 1.1: การดำเนินโครงการ

แผนการดำเนินงาน	มกราคม				กุมภาพันธ์				มีนาคม				เมษายน				พฤษภาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ออกแบบและพัฒนา IDE สำหรับทำงานร่วมกับไจทียโปรแกรม markdown ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาร่วมกัน	x	x	x	x	x															
ออกแบบ สร้าง และตกแต่งตัวเว็บไซต์สำหรับเป็น Web-based IDE					x	x	x	x	x											
ทดสอบการทำงานของ Web-based IDE ร่วมกับไจทียโปรแกรม markdown ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาร่วมกัน และแก้ไขข้อผิดพลาด										x	x	x	x							
จัดทำรายงานฉบับสมบูรณ์														x	x					

ตารางที่ 1.2: การดำเนินโครงการ (ต่อ)

บทที่ 2

ทฤษฎี หรืองานที่เกี่ยวข้อง

2.1 IDE

IDE ย่อมาจาก Integrated Development Environment คือ เครื่องมือที่ช่วยในการพัฒนาโปรแกรม โดยมีสิ่งอำนวยความสะดวกต่างๆ เช่น คำสั่ง Compile, Run ตัวอย่างของ IDE เช่น NetBeans Editplus, JCreator, Eclipse แต่ Notepad ไม่นับว่าเป็น IDE เนื่องจากตัว Notepad เองไม่มีเครื่องมืออำนวยความสะดวกสำหรับการเขียนโปรแกรมใดๆเลย IDE เป็นโปรแกรมที่รวมคำสั่ง เมนู และ GUI ต่างๆมาสร้างเป็นโปรแกรมที่มีรูปร่างหน้าตาเหมาะแก่การเขียนโปรแกรม หรือพูดอีกอย่างว่า เป็นโปรแกรมที่จะสร้างสภาพแวดล้อม (environment) ให้เหมาะแก่การเขียนโปรแกรม

หน้าที่ของโปรแกรม IDE คือการเปิดไฟล์ที่เขียนภาษาโปรแกรม เช่น ภาษา C, Pascal, Java และเซตข้อมูลการคอมไพล์โปรแกรมเก็บไว้ในไฟล์โปรเจกต์ รวมถึงจัดการ Directory และมีปุ่มสร้างโปรแกรมแบบกดทีเดียวทำงานอัตโนมัติจนเสร็จ โปรแกรม IDE 1 โปรแกรม มักจะมีความสามารถเฉพาะบางภาษา ตัวอย่างเช่น โปรแกรมตระกูล Visual ของบริษัทไมโครซอฟท์ ที่มี Visual Basic สำหรับภาษาเบสิก Visual C++ สำหรับภาษา C++

โดยปกติ IDE จะต้องประกอบด้วย

1. Source code editor ที่ใช้สำหรับเขียนคำสั่ง หรือแก้ไขคำสั่ง
2. compiler หรือ interpreter ใช้สำหรับ แปลภาษาคอมไพเตอร์ต่าง ๆ เช่น ภาษาซี จาวา ให้เป็นภาษาเครื่อง
3. Debugger ใช้สำหรับช่วยในการตรวจสอบ หรือหาจุดผิดพลาดของโปรแกรม
4. GUI หรือ Graphics User Interface เป็นเครื่องมือที่ช่วยให้โปรแกรมเมอร์ใช้งาน หรือเขียนโปรแกรมได้ง่ายขึ้น รวดเร็วขึ้น

นอกเหนือจากนั้นจะเป็นความต่างของ IDE แต่ละตัว ว่า IDE ตัวใดจะทำมาเอื้อต่อการทำงานของ developer ได้มากกว่ากัน เช่นบางตัวจะมีส่วน debugger หรือว่าส่วน synchronize ไฟล์ หรือส่วนการจัดการ project file ฯลฯ แล้วแต่ว่า IDE นั้นจะใส่ option มากเท่าใด

2.1.1 Source Code Editor

Text Editor ที่ติดมากับ IDE มีความสามารถ เช่น

- highlight สีของคำสั่ง ตัวแปร function หรือค่าต่างๆในโค้ดได้ เช่น ตัวแปรสีส้ม comment เป็นสีเขียว
- auto suggest เช่น หากกำลังจะพิมพ์ function strtolower แต่พิมพ์แค่พิมพ์ str ตัว text editor ก็จะขึ้น function ทั้งหมดที่ขึ้นต้นด้วย str ขึ้นมาให้เลือกทันที
- ตรวจสอบ syntax ได้ สมมติว่าหากเราพิมพ์ function strtolower เป็น strolower โดยที่เราไม่ทันสังเกต แต่ text editor ก็จะเตือนเราทันทีว่าบรรทัดนี้ผิด หรือว่าลืมปิดคำสั่งด้วยเครื่องหมาย ; ก็เตือนเราได้เช่นกัน
- จัดการ format การเขียนโปรแกรมได้ โดยหากเราพิมพ์อะไรใน notepad แล้วเรากด enter ขึ้นบรรทัดใหม่มันก็จะมาชิดซ้ายสุด แต่ว่า text editor ใน IDE จะจัดย่อหน้าให้อย่างสวยงาม
- ฯลฯ

2.1.2 Compiler หรือ Interpreter

ปกติแล้วคอมพิวเตอร์ทำงานด้วยสัญญาณไฟฟ้า คือเวลาอุปกรณ์ในเครื่องแต่ละชิ้นทำงานพูดคุยกัน มันจะคุยกันด้วยสัญญาณไฟ 0 หรือ 1 โดย 0 ก็คือไม่มีไฟ 1 คือมีไฟ โดยจะสลับกับ เช่น 0001 1000 1101 หรืออื่นๆ เป็นต้น ที่คอมพิวเตอร์เข้าใจจริงๆ แต่ว่าเวลาที่คนเราเขียนโปรแกรม จะไม่สามารถเขียนให้เป็น 00010111 11011100 ได้ จึงเริ่มมีการแปลงตัวเลขพวกนี้ให้เป็นภาษาคนมากขึ้น คือภาษา Assembly ซึ่งเป็นภาษาที่คนเริ่มอ่านออกแล้ว และ simple ที่สุดที่จะใช้เพื่อการแปลงลงไปเป็นคำสั่งของ computer ให้ทำงานต่างๆ เช่น

```
if($haveuser){
    echo "มีคนอยู่";
}
```

เป็นต้น ซึ่งภาษานี้คนเข้าใจแต่คอมพิวเตอร์ไม่เข้าใจ จึงต้องอาศัย Compiler หรือ interpreter ช่วยในการแปลงจากภาษาคนตัวอย่างไปเป็น Assembly เพื่อให้คอมพิวเตอร์เข้าใจคำสั่งของเราและทำงานได้ต่อไป

2.1.3 IDE ที่ได้รับความนิยมในปัจจุบัน

เว็บไซต์ <http://www.comptalks.com/> ซึ่งได้เขียนขึ้นในวันที่ 20 มกราคม 2558 ได้จัดเรียงลำดับ IDE ที่ได้รับความนิยม 10 ลำดับดังนี้

1. Netbeans เป็น IDE ประเภท GPL ของบริษัท Sun Microsystem ปัจจุบัน Oracle ได้ซื้อกิจการของบริษัท Sun Microsystem แล้วจึงยังให้บริการดาวน์โหลดไปใช้ฟรีเหมือนเดิม จุดเด่นของ NetBeans คือ มี Third Party หลายกลุ่มเข้ามามีส่วนร่วมในการพัฒนาเพิ่มเติมเข้าไปได้ นอกจากนี้ยังมี GUI ที่ออกแบบมาสวยงาม และมี syntax Highlight ที่ชัดเจน และสุดท้ายคือ รองรับการใช้งานได้หลายภาษา

2. Eclipse เป็น IDE ประเภท GPL เช่นกัน รองรับภาษาคอมพิวเตอร์หลายภาษา เช่น Java, C, C++, Python เป็นต้น

3. Komodo IDE เป็น IDE ที่มีลิขสิทธิ์ที่ผู้ซื้อต้องซื้อ มีข้อดี ในด้านรองรับหลายภาษา เช่น Perl, PHP, Python, C, C++, JavaScript, HTML, XML
4. Monodevelop เป็น IDE ที่สร้างขึ้นมาเพื่อเป็นทางเลือก ในกรณีที่โปรแกรมเมอร์ต้องการเขียนภาษา C# และ ASP.NET เพราะว่า Monodevelop เป็น IDE ที่เป็น Open source ที่ไม่ต้องซื้อเช่นกัน
5. Aptana เป็น IDE ที่ไม่ต้องซื้อเช่นเดียวกัน รองรับภาษาหลายภาษา เช่น Java, HTML, PHP, JavaScript, และอื่น ๆ ใช้ได้ทั้ง MS Windows, Mac OS, และ Linux
6. QtCreator เป็น IDE ที่เป็นทั้งให้ฟรี และชนิดจำหน่าย รองรับภาษาหลายภาษา มี GUI ที่สวยงาม
7. Code::Blocks เป็น IDE ที่ได้รับความนิยมในการเขียนโปรแกรมในสถานศึกษา รองรับภาษา C, C++ หากติดตั้ง plug-in จะสามารถใช้ภาษา Python ได้ ใช้ดีมาก ๆ กับ C และ C#
8. PSPad เป็น IDE ที่พัฒนาขึ้นมาจากโปรแกรมเมอร์เพียงคนเดียว รองรับเฉพาะ MS Windows เหมาะสำหรับเขียนโปรแกรมสร้างเว็บไซต์
9. Geany เป็น IDE ที่รองรับหลายภาษา เช่น C, Java, PHP, HTML, Python และอื่น ๆ
10. Ulzard เป็น IDE ชนิด Online ที่ให้โปรแกรมเมอร์พัฒนาโปรแกรมบนเว็บไซต์โดยตรง

ในขณะเดียวกัน ในยุคของ Cloud Computing ที่รองรับการทำงานหลาย ๆ platform จึงมี IDE แบบ online เกิดขึ้นอย่างมากมาย ซึ่ง Cloud computer เองมี programming language as a service โดยเว็บไซต์ <http://codecall.net/> ได้เขียนขึ้นเมื่อวันที่ 24 กรกฎาคม 2557 ได้จัดลำดับ IDE Online 10 ชื่อ เรียงตามลำดับดังนี้

1. Cloud9
2. Codenvy
3. Code Anywhere
4. Koding
5. Neutron Drive
6. Collide
7. Orion
8. Python Fiddle
9. Erbix
10. SourceKit

2.2 Web-based Application

คือโปรแกรมหรือกลุ่มของโปรแกรมที่ได้รับการพัฒนาขึ้นมาเพื่อใช้งานในบริการ WWW ของระบบเครือข่ายอินเทอร์เน็ตหรือเครือข่ายอินทราเน็ต ที่ใช้โปรโตคอล TCP/IP เป็นมาตรฐานในการสื่อสารข้อมูล โดยผู้ใช้สามารถติดต่อสื่อสาร หรือเรียกใช้งานโปรแกรม Web-based Application ได้โดยใช้โปรแกรมเว็บเบราว์เซอร์

2.2.1 การพัฒนา Web-based Application

สามารถทำได้โดยการเขียนโปรแกรมในภาษาที่ถูกออกแบบมาสำหรับการพัฒนา Application บนระบบเครือข่ายอินเทอร์เน็ต เช่น Perl , PHP, ASP , JavaScript , VB Script , JSP, JAVA ฯลฯ และใน Application บางชนิดจะต้องมีการติดต่อกับระบบฐานข้อมูลด้วย

2.2.2 จุดเด่นของ Web-based Application

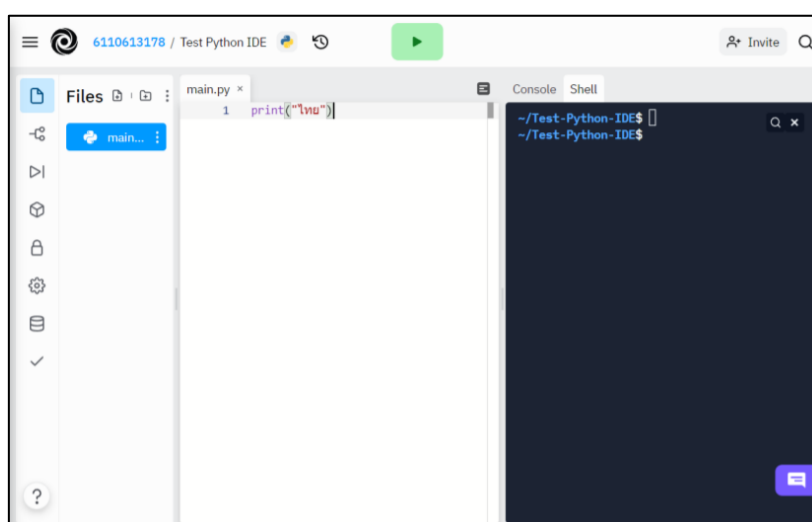
ข้อมูลบนเว็บสามารถเข้าถึงได้จากผู้ชมจำนวนมากโดยไม่มีข้อจำกัดเรื่องชนิดของระบบคอมพิวเตอร์ การนำเสนอข้อมูลบนเว็บเป็นการสื่อสารโดยตรงจากผู้ส่งสารไปยังผู้รับสารโดยใช้เวลานั้น รูปแบบการนำเสนอข้อมูลมีลักษณะเป็นแบบ Hypertext และ Hypermedia ทำให้สามารถนำเสนอข้อมูลที่นำเสนอในรูปแบบมัลติมีเดีย ที่สามารถเชื่อมโยงไปยังข้อมูลอื่น ๆ ที่เกี่ยวข้องได้

แนวโน้มของการนำเสนอข้อมูลบนอินเทอร์เน็ตมีลักษณะ Interactive คือมีกิจกรรมที่ทำให้ผู้เข้าชมมีส่วนร่วมเว็บไซต์มากขึ้น เช่น Guestbook, Message board , forums etc.

แนวโน้มของการนำเสนอข้อมูลบนอินเทอร์เน็ตมีลักษณะ Dynamic คือมีการปรับปรุงข้อมูลให้ทันสมัยอยู่เสมอโดยอัตโนมัติ

2.3 Repl.it

Repl.it เป็นเว็บไซต์ที่ทำให้ผู้ใช้งานสามารถใช้งาน IDE บนเว็บไซต์ได้ Repl.it นั้นมีคุณสมบัติที่จำเป็นและช่วยอำนวยความสะดวกให้กับผู้ที่ต้องการใช้งาน IDE ประกอบด้วย การรองรับภาษามากกว่า 50 ภาษา เช่น Clojure, Haskell, Kotlin(beta), QBasic, Forth, LOLCODE, BrainF, Emoticon, Bloop, Unlambda, JavaScript, CoffeeScript, Scheme, APL, Lua, Python 2.7, Ruby, Roy, Python, Nodejs, Golang, C++, C, C#, F#, HTML, CSS, JS, Rust, Swift, Python (with Turtle), R, Bash, Quil, Crystal, Julia, Elixir, Nim, Dart, Reason, NodeJs, Erlang, TypeScript, Pygame, Love2D, Tkinter, Java, Swing, Emacs Lisp (Elisp), PHP Web Server, SQLite, Java, PHP, CLI และ Pyxel มีทั้ง console และ shell ที่สามารถรับอินพุตและแสดงเอาต์พุตได้ สามารถสร้างไฟล์และไดเรกทอรีสำหรับโปรเจกต์ได้ ดังรูปที่ 2.1



รูปที่ 2.1: หน้า UI IDE ของ Repl.it

2.4 Xterm.js

Xterm.js เป็นส่วนประกอบของ front-end ที่ถูกเขียนขึ้นด้วย TypeScript ซึ่งช่วยให้แอปพลิเคชันสามารถนำ terminal ที่มีคุณสมบัติหลากหลายมาใช้บนเบราว์เซอร์ได้ รูปที่ 2.2 แสดงหน้าเว็บไซต์ Xterm.js



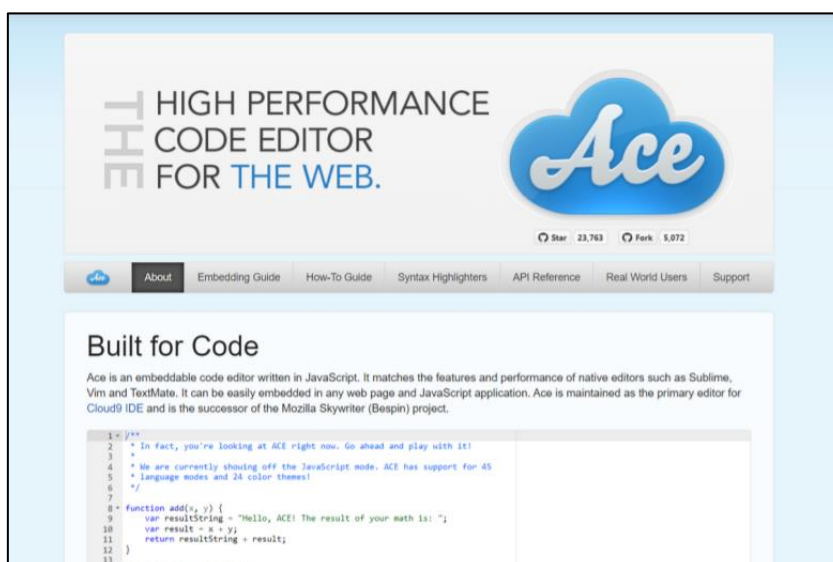
รูปที่ 2.2: หน้าเว็บไซต์ Xterm.js

2.4.1 Features

- terminal app ที่ทำงานร่วมกับ: Xterm.js นั้นสามารถทำงานร่วมกับ terminal app เช่น bash, vim และ tmux
- ประสิทธิภาพ: Xterm.js สามารถทำงานได้อย่างรวดเร็ว และใช้งาน GPU-accelerated renderer ได้
- รองรับ Unicode ที่หลากหลาย: รองรับ CJK, emoji และ IME
- สามารถทำงานได้ด้วยตัวเอง: ไม่ต้องการ dependencies ในการทำงาน
- เข้าถึงได้ง่าย: สามารถเปิดด้วย Screen reader และอัตราส่วนคอนทราสต์ขั้นต่ำได้
- อื่นๆอีกมากมาย: เช่น Link, theming, addons, API ที่ถูกทำมาเป็นเอกสารอย่างดี ฯลฯ

2.5 Ace

Ace เป็น code editor แบบฝังที่ถูกเขียนขึ้นด้วย JavaScript มีคุณสมบัติและประสิทธิภาพตรงกับ editor ดั้งเดิม เช่น Sublime, Vim และ TextMate สามารถฝังลงบนหน้าเว็บไซต์และแอปพลิเคชันได้อย่างง่ายดาย Ace นั้น เป็น editor หลักของ Cloud9 IDE และพัฒนามาจากโครงการ Mozilla Skywriter (Bespun) รูปที่ 2.3 แสดงหน้าเว็บไซต์ Ace



รูปที่ 2.3: หน้าเว็บไซต์ Ace

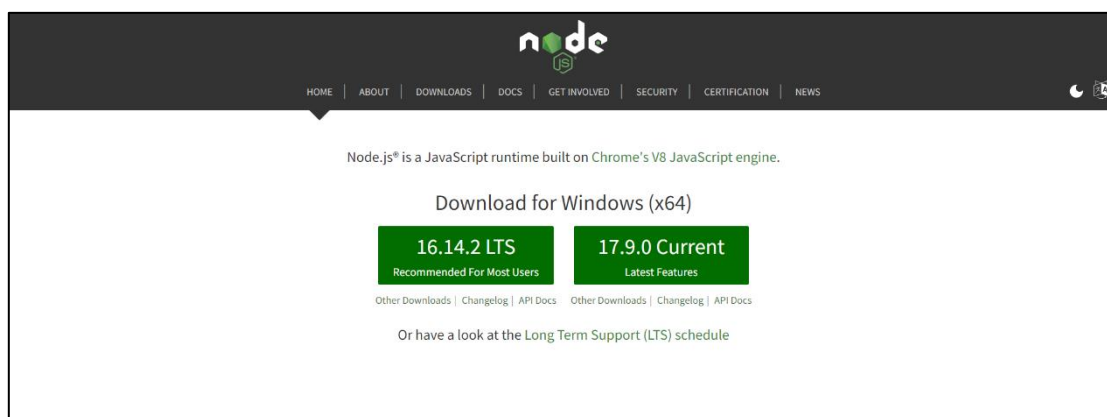
2.5.1 Features

- ไฮไลต์ syntax ได้มากกว่า 100 ภาษา (สามารถนำเข้าไฟล์ TextMate/Sublime Text.tmLanguage ได้)
- มี theme ให้เลือกมากกว่า 20 แบบ (สามารถนำเข้าไฟล์ TextMate/Sublime Text.tmTheme ได้)
- จัดระเบียบข้อความได้อัตโนมัติ
- สามารถเลือก command line ได้
- รองรับเอกสารขนาดใหญ่ (มีขีดจำกัดอยู่ที่ประมาณ 4 ล้านบรรทัด)
- key binding สามารถปรับแต่งได้อย่างเต็มที่ (รวมถึงมีโหมด vim และ Emacs)
- สามารถค้นหาและแทนที่ด้วยนิพจน์ทั่วไปได้
- ไฮไลต์วงเล็บที่ตรงกัน
- สามารถสลับไปมาระหว่าง soft tab และ real tab ได้
- แสดงอักขระที่ซ่อนอยู่ได้

- คลุมและวางข้อความโดยใช้เมาส์
- จัดเก็บแต่ละบรรทัดได้
- จัดเก็บโค้ดได้
- เคอร์เซอร์หลายตัวสามารถทำงานพร้อมกัน และเลือกพร้อมกันได้
- สามารถเช็ค syntax ได้ตลอดเวลา (ปัจจุบันมี JavaScript, CoffeeScript, CSS, XQuery)
- ฟังก์ชัน cut, copy และ paste

2.6 Node.js

Node.js คือสภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ที่ทำงานด้วย V8 engine นั้นหมายความว่าเราสามารถใช้ Node.js ในการพัฒนาแอปพลิเคชันแบบ Command line แอปพลิเคชัน Desktop หรือแม้แต่เว็บเซิร์ฟเวอร์ได้ โดยที่ Node.js จะมี APIs ที่เราสามารถใช้สำหรับทำงานกับระบบปฏิบัติการ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ก เป็นต้น รูปที่ 2.4 แสดงหน้าเว็บไซต์ Node.js



รูปที่ 2.4: หน้าเว็บไซต์ Node.js

2.7 Socket.io

Socket.io เป็น JavaScript frameworks ที่เอาไว้เรียกใช้งาน Websocket เพื่อคอยรับส่งข้อมูลจาก client-browser และ server และทำงานในแบบ Real-time ตัวอย่างของการใช้งาน Websocket เช่น Chat room, Document collaboration ซึ่งจะแตกต่างจากรูปแบบการทำงานของเว็บทั่วไปหรือ RESTful HTTP ซึ่งจะมีลักษณะการทำงานแบบ Request-Respond คือ เมื่อ Client ทำการส่ง Request หน้าเว็บไป ตัว Server ก็จะตอบกลับมาด้วย Respond และจะทำงานในลักษณะนี้ไปเรื่อยๆ แต่ในการทำงานของ Websocket นั้นจะมีลักษณะ event-based communication คือ จะทำงานตามที่เราเขียน code ไว้ เช่น ถ้าเกิดเหตุการณ์ A ก็ให้ส่งข้อมูล B ไป เป็นต้น

บทที่ 3

การดำเนินงาน/วิธีวิจัย

3.1 ศึกษาการใช้งานเครื่องมือที่จำเป็น

3.1.1 Repl.it

ใช้เป็นตัวอย่าง web-based IDE

3.1.2 Xterm.js

ใช้กำหนดหน้าต่างของ terminal

3.1.3 Ace

ใช้เมธอดต่างๆสำหรับการทำงานของ IDE เช่น `setTheme` สำหรับการตั้งค่าธีมของหน้าต่างเขียนโค้ด, ใช้เมธอด `setMode` ในการเลือกโหมดการทำงานของ IDE ตามแต่ละภาษา, ใช้เมธอด `getValue` สำหรับการรับค่าโค้ดเข้ามา เป็นต้น

3.1.4 Node.js

ใช้ในการรันโปรเจกต์ ทั้งตัว client และ server

3.1.5 Socket.io

เรียกใช้งาน Websocket เพื่อคอยรับส่งข้อมูลจาก client และ server

3.2 ดาวน์โหลดและติดตั้งเครื่องมือที่จำเป็น

3.2.1 โปรแกรมที่ใช้

1. Visual Studio Code
2. Node.js
3. Java
4. Python

3.2.2 Package ที่ใช้

1. Ace

3.3 สร้าง Web-based Terminal

3.3.1 สร้างส่วน Server

1. สร้างโฟลเดอร์ server และตั้งเป็นโปรเจก node.js สำหรับเก็บส่วนประกอบทั้งหมดของ server หลังจากนั้นไว้ภายในโฟลเดอร์
2. สร้างไฟล์ PTYService.js ตั้งค่าการทำงานของ terminal โดยใช้ node-pty ในการสร้างการทำงานแบบ pseudo-terminal การรับ input จะรับผ่าน node-pty และส่ง output ไปยัง socket.io client
3. สร้างไฟล์ SocketService.js กำหนดการทำงานของ event ที่เกี่ยวกับ socket มีการรับตัว server และ PTY เพื่อทำการสร้างช่องทางการติดต่อกันระหว่าง client และ server และสั่งเริ่มการทำงานใน PTYService.js โดยมีตัว socket เป็น input
4. สร้างไฟล์ index.js เพื่อทำการสร้าง server กำหนด port ของ server และสั่งเริ่มการทำงานใน SocketService.js โดยส่งตัว server ที่สร้างมานี้เข้าเป็น input

3.3.2 สร้างส่วน Client

1. สร้างโฟลเดอร์ client และตั้งเป็นโปรเจก node.js สำหรับเก็บส่วนประกอบทั้งหมดของ client หลังจากนั้นไว้ภายในโฟลเดอร์
2. สร้างไฟล์ TerminalUI.js ใช้ xterm.js ในการสร้างตัว front-end และตกแต่งหน้าต่างของ terminal มีการ print input ที่ได้รับมาจาก PTY บน terminal และมีการส่ง input ไปยัง PTY process ใน server
3. สร้างไฟล์ Terminal.js เพื่อสร้าง socket.io client ในการรับ/ส่ง event จาก PTY process บน server มีการเชื่อมต่อ socket ไปยัง port ของ server และมีการสั่งเริ่มต้นการทำงานของ terminal
4. สร้างไฟล์ index.html เพื่อแสดงผล Web-based Terminal ที่รับมาจาก Terminal.js

3.4 สร้าง Web-based IDE

3.4.1 เพิ่มส่วน Server

1. สร้างโฟลเดอร์ storage สำหรับเก็บไฟล์โค้ด

2. เพิ่มโค้ดในไฟล์ index.js ให้ทุกครั้งที่ทำการรัน server จะทำการลบไฟล์ทั้งหมดในโฟลเดอร์ storage ให้กลายเป็นโฟลเดอร์เปล่า หากมีการรับ post ที่ระบุค่าโค้ด, ค่าชื่อไฟล์ และค่าภาษา มาจากฝั่ง client จะทำการเขียนไฟล์โค้ด โดยตัวโค้ดภายในไฟล์, ชื่อไฟล์ และนามสกุลไฟล์ จะขึ้นอยู่กับค่าโค้ด, ค่าชื่อไฟล์ และค่าภาษาที่ได้รับมา หากชื่อไฟล์และนามสกุลไฟล์ซ้ำ จะทำการเขียนทับไฟล์เดิม หากมีการรับ post ที่ระบุค่าโจทย์โปรแกรม markdown มาจากฝั่ง client จะส่งค่าโจทย์นั้นไปยัง API ของกลุ่มโครงงานที่มีอาจารย์ที่ปรึกษาร่วมกันเพื่อตรวจโจทย์ จากนั้นจะทำการรับผลลัพธ์การตรวจ และส่งผลลัพธ์นั้นกลับไปยังฝั่ง client

3.4.2 เพิ่มส่วน Client

1. เพิ่มโค้ดในไฟล์ index.html ให้มีส่วนประกอบอื่นๆของ IDE เพิ่มเข้ามา ประกอบด้วย ปุ่มเลือกภาษาระหว่างภาษา Python และ Java, หน้าต่างเขียนโค้ด, ช่องตั้งชื่อไฟล์โค้ด, ปุ่มบันทึกโค้ด, หน้าต่างแสดงผลการตรวจโจทย์โปรแกรม markdown และปุ่มตรวจโจทย์ และนำเข้าการทำงานของไฟล์ Editor.js

2. สร้างไฟล์ Editor.js สำหรับเขียนฟังก์ชันต่างๆที่เกี่ยวข้องกับการทำงานของส่วนประกอบของ IDE ข้างต้น ประกอบด้วย

1) ฟังก์ชันเริ่มต้นการทำงานเว็บไซต์ โดยเมื่อเริ่มต้นการทำงานเว็บไซต์ จะแสดงหน้าต่างเขียนโค้ด แสดงผลรวมไปกับ terminal และกำหนดธีมและโหมดเริ่มต้นของหน้าต่างเขียนโค้ดด้วย Ace

2) ฟังก์ชันเปลี่ยนภาษา สำหรับเปลี่ยนโหมดของหน้าต่างเขียนโค้ดให้มีการรูปแบบการไฮไลต์โค้ดตามภาษาที่เลือกด้วย Ace

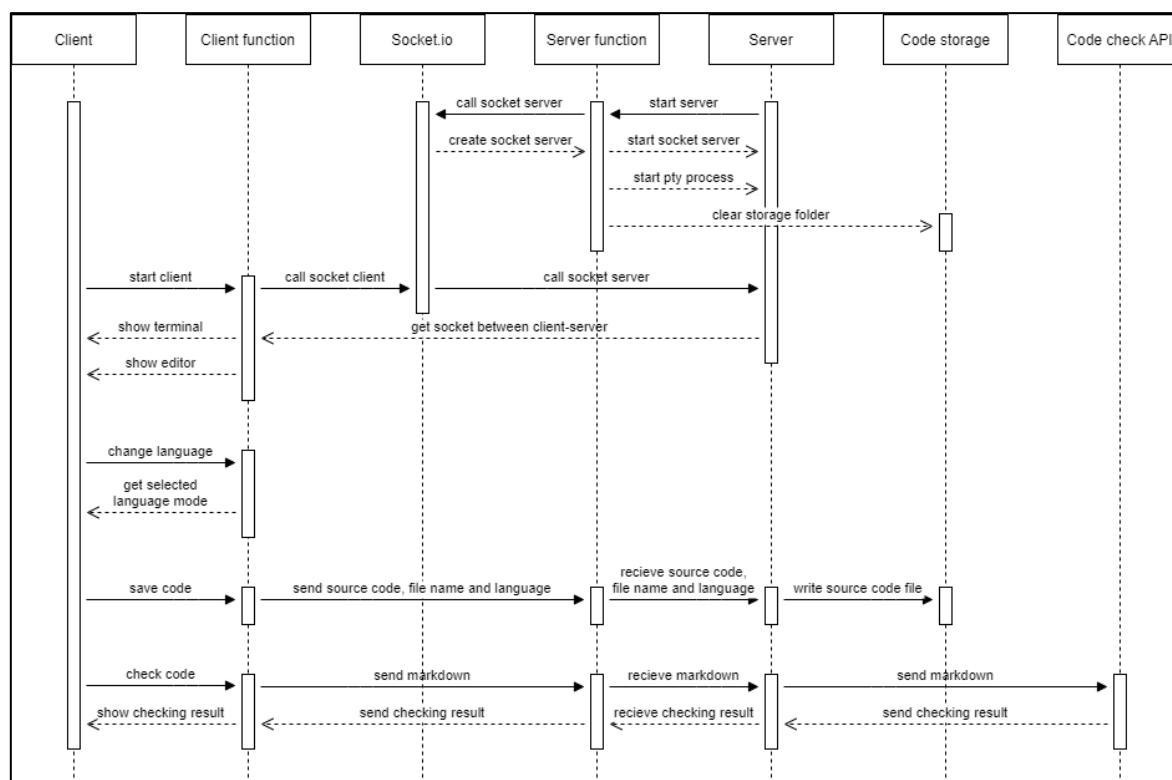
3) ฟังก์ชันบันทึกโค้ดเป็นไฟล์ โดยเมื่อกดปุ่มบันทึกโค้ด จะทำการส่ง post ที่ระบุค่าโค้ด, ค่าชื่อไฟล์ และค่าภาษา ไปยังฝั่ง server เพื่อให้ฝั่ง server ทำการเขียนไฟล์ตามค่าที่ระบุไปอีกที่

4) ฟังก์ชันตรวจโจทย์ โดยเมื่อกดปุ่มตรวจโจทย์ จะทำการส่ง post ที่ระบุค่าโจทย์โปรแกรม markdown จากหน้าต่างเขียนโค้ดไปยังฝั่ง server เพื่อให้ฝั่ง server ส่งค่าโจทย์นั้นไปยัง API ของกลุ่มโครงงานที่มีอาจารย์ที่ปรึกษาร่วมกันอีกที่เพื่อตรวจโจทย์ หากฝั่ง server ส่งผลลัพธ์การตรวจกลับมาแล้ว จะนำผลลัพธ์นั้นไปแสดงยังหน้าต่างแสดงผลการตรวจโจทย์บนหน้าเว็บไซต์

บทที่ 4

ผลการดำเนินงานและอภิปรายผล

4.1 หลักการทำงานของ Web-based IDE



รูปที่ 4.1: Sequence Diagram แสดงหลักการทำงานของ Web-based IDE

4.1.1 เริ่มต้นการทำงานของ Server

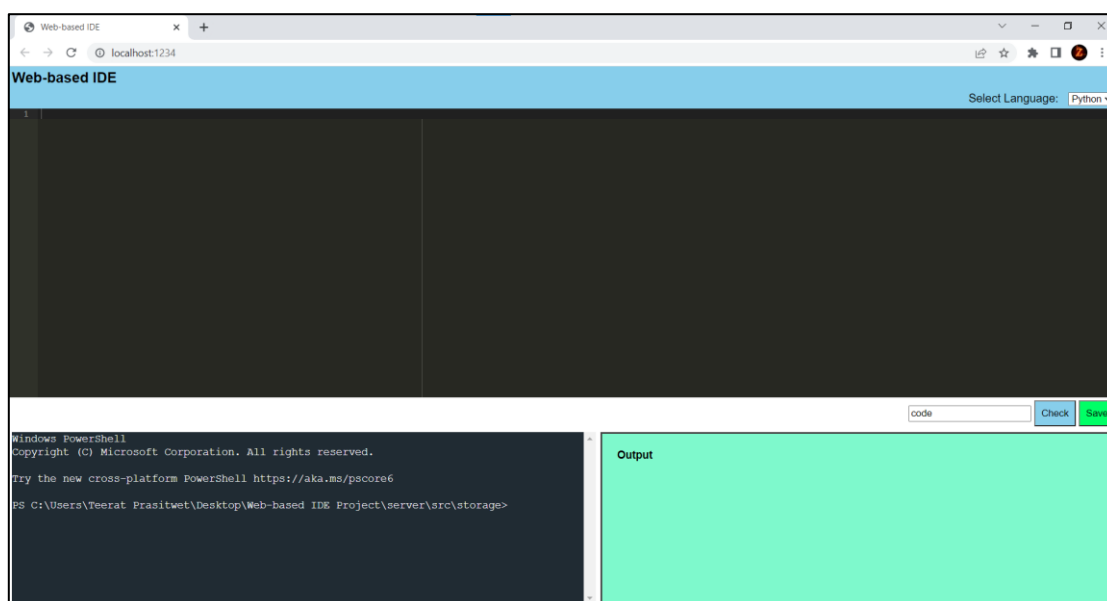
1. server จะสั่งเริ่มการทำงานที่ฟังก์ชันของ server แล้วฟังก์ชันของ server จะทำการเรียก socket server มาจาก Socket.io

2. Socket.io จะสร้าง socket server แล้วส่งกลับมายังฟังก์ชันของ server จากนั้นฟังก์ชันของ server จะทำให้ทางฝั่ง server มีการทำงานของ socket server และ pseudo-terminal และลบข้อมูลทั้งหมดในโฟลเดอร์เก็บให้กลายเป็นโฟลเดอร์เปล่า ดังรูปที่ 4.1

4.1.2 เริ่มต้นการทำงาน Client

1. client จะส่งเริ่มการทำงานที่ฟังก์ชันของ client แล้วฟังก์ชันของ client จะทำการเรียก socket client มาจาก Socket.io ทำให้ Socket.io ไปเรียก socket server มาจากฝั่ง server

2. ฝั่ง server จะได้ socket ที่เป็นการติดต่อกันระหว่าง client กับ server แล้วส่งกลับมาให้ฟังก์ชันของ client จากนั้นฟังก์ชันของ client จะทำให้ทางฝั่ง server มีการแสดงผลทั้ง terminal และ editor ออกมา ดังรูปที่ 4.1 โดยจากนี้ terminal จะสามารถรับ-ส่งข้อมูล input และ output ผ่าน socket ได้ ดังรูปที่ 4.2



รูปที่ 4.2: หน้าเว็บไซต์เริ่มต้นของ Web-based IDE

4.1.3 เลือกเปลี่ยนภาษา

1. client จะส่งเปลี่ยนภาษาไปยังฟังก์ชันของ client

2. ฟังก์ชันของ client จะได้โหมดของหน้าต่างเขียนโค้ดเป็นภาษาตาม que เลือก แล้วส่งกลับมาที่ฝั่ง client ดังรูปที่ 4.1 ทำให้ในตอนนี หน้าตาเขียนโค้ดจะมีการไฮไลต์โค้ดในรูปแบบของภาษาตาม que เลือก

4.1.4 กดปุ่ม save code

1. client จะส่งบันทึกโค้ดไปยังฟังก์ชัน client แล้วฟังก์ชันของ client จึงส่งค่าโค้ด, ค่าชื่อไฟล์ และค่าภาษาของโค้ดนั้นไปยังฟังก์ชันฝั่ง server

2. หลังจากฝั่ง server ได้รับค่าโค้ด, ค่าชื่อไฟล์ และค่าภาษาของโค้ดแล้ว จะทำการเขียนไฟล์โค้ดซึ่งเป็นไฟล์ตามภาษานั้นลงบนไฟล์เดสก์ทอป โดยจะมีเนื้อหาภายในไฟล์ตามค่าโค้ด และมีชื่อไฟล์ตามค่าชื่อไฟล์ที่ได้รับมา หากไฟล์โค้ดนั้นมีชื่อไฟล์ซ้ำกับที่มีอยู่ในไฟล์เดสก์ทอปอยู่แล้ว จะทำการเขียนทับไฟล์โค้ดเดิมที่เคยมีอยู่ทันที ดังรูปที่ 4.1 ทำให้ในตอนนี้ ผู้ใช้สามารถส่งรันไฟล์โค้ดที่ได้บันทึกไปล่าสุดผ่านการรันใน terminal บนหน้าเว็บไซต์ได้

4.1.5 กดปุ่ม check code

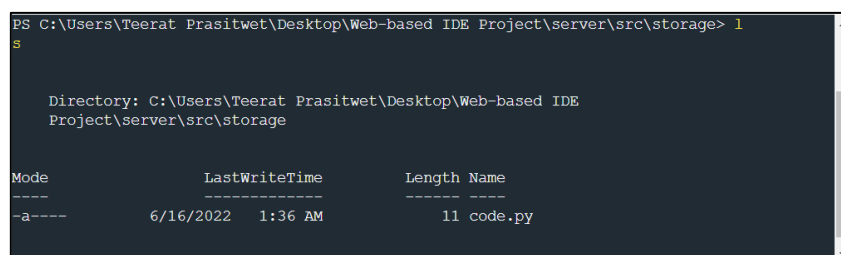
1. client จะส่งตรวจเช็คโปรแกรม markdown ไปยังฟังก์ชัน client แล้วฟังก์ชันของ client จึงส่งค่าไจทน์ไปยังฟังก์ชันฝั่ง server

2. หลังจากฝั่ง server ได้รับค่าไจทน์แล้ว จะทำการส่งค่าไจทน์ไปยัง API ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาร่วมกันเพื่อตรวจเช็ค API จะส่งผลการตรวจกลับมายังฝั่ง server จากนั้น server จะส่งผลลัพธ์นั้นกลับไปยังฟังก์ชันฝั่ง client อีกที ดังรูปที่ 4.1 เพื่อให้ client แสดงผลลัพธ์ให้ผู้ใช้งานได้เห็นผ่านหน้าต่างแสดงผลการตรวจไจทน์บนหน้าเว็บไซต์

4.2 ผลการสร้างตัว Web-based Terminal

4.2.1 การทำงานของ terminal

1. terminal สามารถทำงานได้ในรูปแบบเดียวกับ powershell โดยมี path และการทำงานอยู่ในตัวเครื่อง server ที่ทำการรัน terminal ดังรูปที่ 4.3



```
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> ls
Directory: C:\Users\Teerat Prasitwet\Desktop\Web-based IDE
Project\server\src\storage

Mode                LastWriteTime         Length Name
----                -
-a-----         6/16/2022  1:36 AM             11 code.py
```

รูปที่ 4.3: แสดงตัวอย่าง path และการใช้งาน command ใน terminal

2. สามารถแสดง path ของ terminal ในปัจจุบัน
 3. สามารถเข้าถึงทุก path ของเครื่อง server ได้เหมือนกับ powershell ของเครื่อง server เอง ดังรูปที่ 4.4

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> cd ..
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src>
```

รูปที่ 4.4: แสดงตัวอย่างการเข้าถึง path อื่นๆในเครื่อง server

4. สามารถรันไฟล์ แสดงผล output และรับ input จาก Web-based Terminal ได้ ทั้ง ภาษา Python และ Java ดังรูปที่ 4.5, 4.6, 4.7, 4.8

```
server > src > storage > code.py > ...
1  x = input("Enter your name: ")
2  print("Hello, " + x)
3
```

รูปที่ 4.5: แสดงตัวอย่างโค้ดรับ input ภาษา Python ในไฟล์ code.py ผ่าน Visual Studio Code

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> javac code.java
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> java code
Enter your name:
TestName
Hello, TestName
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage>
```

รูปที่ 4.6: แสดงตัวอย่างการรันไฟล์ การรับ input และแสดงผล output ไฟล์ภาษา Python ผ่าน terminal

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> python code.py
Enter your name: TestName
Hello, TestName
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage>
```

รูปที่ 4.7: แสดงตัวอย่างโค้ดรับ input ภาษา Java ในไฟล์ code.java ผ่าน Visual Studio Code

```

server > src > storage > code.java > ...
1  import java.util.Scanner;
2
3  public class code {
    Run | Debug
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("Enter your name: ");
7
8          String x = sc.nextLine();
9          System.out.println("Hello, " + x);
10     }
11 }

```

รูปที่ 4.8: แสดงตัวอย่างการรันไฟล์ การรับ input และแสดงผล output ไฟล์ภาษา Java ผ่าน terminal

5. ใช้ environment ของเครื่อง server เองเป็น environment ให้ terminal ทำให้หากเปิด client หลาย client environment ในทุก terminal จะเป็น environment เดียวกัน

6. มีบัคการแสดงผลบางกรณี เช่น บางครั้งการ input ใน terminal จะทำให้ terminal แสดงผลตัวอักษรตัวแรกของ input นั้นซ้ำหลายครั้ง, หากใช้ command "ls" จะทำให้ทุกการ input ใน terminal หลังจากนั้น terminal จะแสดงผลตัวอักษรตัวแรกของ input นั้นซ้ำหลายครั้ง และบางครั้ง command "ls" แสดงไฟล์ใน directory ได้ไม่ครบหากเป็นการ input ครั้งแรก โดยบัคทั้งหมดที่เกิดขึ้นนี้เป็นเพียงบัคการแสดงผลใน terminal แต่ไม่ส่งผลต่อค่าจริง ๆ ที่ทำการ input เข้าไป และไม่ส่งผลต่อไฟล์ในเครื่อง server ดังรูปที่ 4.9

Mode	LastWriteTime	Length	Name
-a----	6/16/2022 2:11 AM	750	code.class
-a----	6/16/2022 2:11 AM	261	code.java

```

PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> p
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> p
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> p
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> p
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> p
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> p
PS C:\Users\Teerat Prasitwet\Desktop\Web-based IDE Project\server\src\storage> p
y code

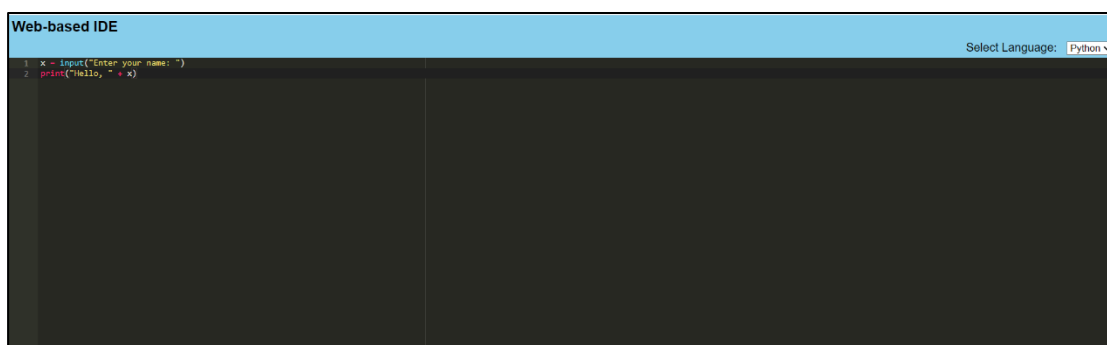
```

รูปที่ 4.9: แสดงตัวอย่างบัคการแสดงผลตัวอักษรตัวแรกของ input ซ้ำหลายครั้ง

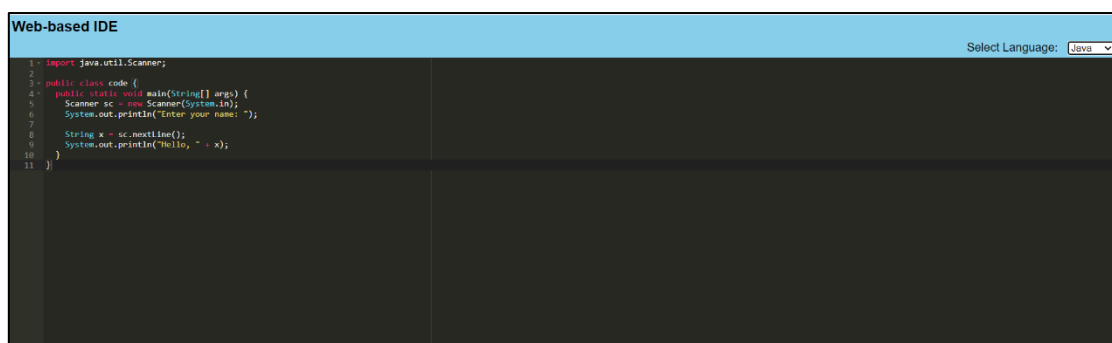
4.3 ผลการสร้างตัว Web-based IDE

4.3.1 การทำงานของ IDE

1. สามารถแสดงผลหน้าต่างเขียนโค้ดตามธีมที่กำหนดด้วย Ace ได้
2. สามารถเลือกเปลี่ยนโหมดของหน้าต่างเขียนโค้ดตามภาษาที่ต้องการระหว่าง Python และ Java ได้ หน้าต่างเขียนโค้ดจะทำการไฮไลต์โค้ดในรูปแบบของภาษาที่เลือก ดังรูปที่ 4.10, 4.11

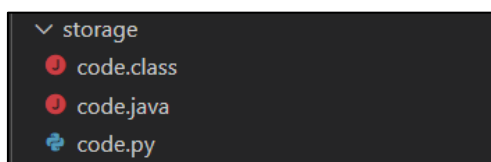


รูปที่ 4.10: แสดงตัวอย่างการไฮไลต์โค้ดภาษา Python ในหน้าต่างเขียนโค้ด



รูปที่ 4.11: แสดงตัวอย่างการไฮไลต์โค้ดภาษา Java ในหน้าต่างเขียนโค้ด

3. สามารถบันทึกโค้ดที่เขียนในหน้าต่างเขียนโค้ดลงบนเครื่องฝั่ง server ได้ โดยจะบันทึกเป็นไฟล์ที่มีเนื้อหาภายในและมีนามสกุลตามโค้ดและโหมดภาษาของหน้าต่างเขียนโค้ด และมีชื่อไฟล์ตามที่กำหนดไว้ในช่องตั้งชื่อไฟล์ ดังรูปที่ 4.12



รูปที่ 4.12: แสดงไฟล์ภาษา Python และ Java ที่ถูกบันทึกลงในโฟลเดอร์ storage

4. สามารถส่งโจทย์โปรแกรม markdown ที่เขียนในหน้าต่างเขียนโค้ดไปตรวจที่ API ของกลุ่มโครงการที่มีอาจารย์ที่ปรึกษาพร้อมกันได้ และนำผลลัพธ์มาแสดงผลที่หน้าต่างแสดงผลลัพธ์การตรวจโจทย์ได้ ดังรูปที่ 4.13, 4.14, 4.15

```

1 [run option]:#time:10 mem:102400 filename:input
2 [language]:py
3 ::startcode:
4 input = input()
5 print("hello, " + input)
6 ::endcode:
7 [check condition]:exact-mach//ignore space/sensitive//insensitive
8 ::start input:
9 ::end input:
10 ::start output:
11 hello, world
12 ::end output:
13 [check element]:#if for
  
```

รูปที่ 4.13: แสดงตัวอย่างการไฮไลต์โจทย์โปรแกรม markdown ภาษา Python ในหน้าต่างเขียนโค้ด

```

1 [run option]:#time:10 mem:102400 filename:HelloInput
2 [language]:#java
3 ::startcode:
4 import java.io.BufferedReader;
5 import java.io.InputStreamReader;
6 import java.io.IOException;
7 class HelloInput {
8     public static void main(String[] args) throws IOException{
9         BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
10         String name = reader.readLine();
11         System.out.println("hello, " + name);
12     }
13 }
14 ::endcode:
15 [check condition]:#input/exact-mach//ignore space//sensitive//insensitive
16 ::start input:
17 world
18 ::end input:
19 ::start output:
20 hello, world
21 ::end output:
22 [check element]:#if for
  
```

รูปที่ 4.14: แสดงตัวอย่างการไฮไลต์โจทย์โปรแกรม markdown ภาษา Java ในหน้าต่างเขียนโค้ด

Output

```

1: {"Condition": "exact-mach", "Result": false, "Percent-Correct": 72, "Expect output": "hello, world\n", "Your output": "hello, None\n"}
2: {"Condition": "ignore-space", "Result": false, "Percent-Correct": 66.67, "Expect output": "hello, world", "Your output": "hello, None"}
3: {"Condition": "insensitive-case", "Result": false, "Percent-Correct": 72, "Expect output": "hello, world\n", "Your output": "hello, none\n"}
4: {"Condition": "sensitive-case", "Result": false, "Percent-Correct": 72, "Expect output": "hello, world\n", "Your output": "hello, None\n"}
5: {"Condition": "check_element", "Percent-Correct": 0, "ElementIsFound": {"if": false, "for": false}}
  
```

รูปที่ 4.15: แสดงตัวอย่างผลลัพธ์ในหน้าต่างแสดงผลลัพธ์การตรวจโจทย์โปรแกรม markdown

บทที่ 5

สรุปผลการดำเนินงาน อุปสรรค และการพัฒนาในอนาคต

5.1 สรุปผลการดำเนินงาน

1. ตัวเว็บไซต์มีหน้าแรกเพียงหน้าเดียว ทำให้ไม่เกิดความซับซ้อนในการเข้าถึงการใช้งานฟังก์ชันต่างๆ ของเว็บไซต์
2. terminal สามารถทำงานได้บนเว็บไซต์
3. หน้าต่างเขียนโค้ดสามารถทำงานได้ทั้ง 2 ภาษา คือ Python และ Java
3. สามารถส่งโจทย์โปรแกรม markdown ไปตรวจที่ API ของกลุ่มโครงงานที่มีอาจารย์ที่ปรึกษา ร่วมกัน แล้วนำผลลัพธ์ของการตรวจสอบมาแสดงผลได้
4. สามารถนำ Web-based IDE มาใช้เพื่อการศึกษา พัฒนาและทดสอบการใช้งานซอฟต์แวร์บนเว็บไซต์ได้

5.2 อุปสรรค

1. การทำงานของเครื่องมือที่เลือกใช้บางครั้งไม่ครอบคลุมถึงฟังก์ชันการทำงานใหม่ๆ ที่ถูกเขียนขึ้น ในภายหลัง ทำให้ต้องมีการศึกษาและเปลี่ยนไปใช้เครื่องมืออื่นแทนระหว่างการทำงานหลายครั้ง
2. ใช้เครื่องคอมพิวเตอร์ส่วนตัวที่ใช้อยู่เป็นประจำในการเป็นทั้งเครื่อง client และ server ทำให้ไม่สามารถจัดการเรื่องการจำกัดการเข้าถึง path อื่นๆ ใน Web-based Terminal เพื่อรักษาความปลอดภัยได้ เนื่องจากต้องมีการปรับแต่ง powershell ของตัวเครื่องคอมพิวเตอร์ ซึ่งอาจส่งผลกระทบให้เกิดความเสียหาย ต่อการทำงานของตัวเครื่องคอมพิวเตอร์ได้
3. ไม่สามารถหาสาเหตุและแก้ไขบัคที่เกิดขึ้นกับการแสดงผลบางอย่างของตัว Web-based Terminal ได้

5.3 การพัฒนาในอนาคต

1. จัดการเรื่องจำกัดการเข้าถึง path อื่นๆ ของผู้ใช้ ผ่าน Web-based Terminal นอกจาก directory ที่กำหนดของเครื่อง server เพื่อความปลอดภัยของตัวเครื่อง server
2. แก้ไขบัคที่เกิดขึ้นกับการแสดงผลบางอย่างของตัว Web-based Terminal
3. แยก environment ของ terminal จากต่าง client เพื่อให้ client สามารถใช้งาน Web-based Terminal ได้หลาย client พร้อมกัน
4. ปรับปรุงหน้าเว็บไซต์ให้มีความสวยงามและน่าใช้งานมากขึ้น

บรรณานุกรม

สื่ออิเล็กทรอนิกส์

1. pongkung. (ก.พ. 2560). "IDE คืออะไร" [Online]. Available: <http://computer2know.blogspot.com/2017/02/ide.html>. [Accessed: 11 กันยายน 2564].
2. Admin ITGenius. (ต.ค. 2557). "Web-based Application คืออะไร" [Online]. Available: <https://www.itgenius.co.th/article/Web-based%20Application%20%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html>. [Accessed: 11 กันยายน 2564].
3. "xterm.js" [Online]. Available: <https://github.com/xtermjs/xterm.js/>. [Accessed: Sep. 13, 2021].
4. "Built for Code" [Online]. Available: <https://ace.c9.io/>. [Accessed: Sep. 13, 2021].
5. Sai Sandeep Vaddi. (Oct. 2019) "How to create web-based terminals" [Online]. Available: <https://dev.to/saisandeepvaddi/how-to-create-web-based-terminals-38d?fbclid=IwAR38GY1Pm9KKeLnbbW8vgK-PJkbZZRRUiObhPdQkcHYKqV6WUNlgOY4Atn0>. [Accessed: Jan. 15, 2022].

ภาคผนวก

ลิงค์ Github ของโครงการ

<https://github.com/6110613178/Web-based-IDE.git>