

**โครงงานวิศวกรรมคอมพิวเตอร์**

พัฒนาระบบ “เว็ปรวมปฏิทิน”

**โดย**

นายอนุสรณ์ ศรีพรหม

อาจารย์ที่ปรึกษา

อ.ดร.คมศัลล์ ศรีวิสุทธิ์

รายงานนี้เป็นส่วนหนึ่งของรายวิชา 523491 สหกิจศึกษาของสาขาวิศวกรรมคอมพิวเตอร์

ตามหลักสูตรปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 1/2566

**กิตติกรรมประกาศ**

โครงงานเรื่อง “เว็ปบันทึกการพบประอาจารย์ที่ปรึกษาโครงงาน”สำเร็จได้ด้วยดีเนื่องจากได้รับความกรุณาอย่างสูงจาก อ. ดร.คมศัลล์ ศรีวิสุทธิ์ ที่ให้คำแนะนำและชี้แนะมาโดยตลอดตั้งแต่ต้นจนสำเร็จลุล่วงได้โดนดีผู้จัดทำโครงงานขอขอบพระคุณด้วยความเคารพอย่างสูงไวเ ณ โอกาสนี้

และขอขอบคุณพระคุณมหาวิทยาลัยโทคโนโลยีสุรนารี ที่ทำให้ได้มาเรียนในสาจาคอมวิศวกรรมคอมพิวเตอร์แห่งนี้ ทำให้ได้เจอ เพื่อ พี่น้อง และอาจารย์ผู้สอนทุกท่านที่คอยให้คำแนะนำ และชี้นำทางด้านการเรียนและการใช้ชีวิตต่างๆ

สุดท้ายนี้ ขอขอบคุณอาจารย์ผู้สอนทุกท่านที่ให้ความรู้ทางด้านวิชาการ และขอขอบพระคุณบิดา มารดา ที่ให้กำเนิด ความรัก กำลังใจ และส่งเสริมทางด้านการศึกษาอย่างดีโดยตลอด จนทำให้ประสบความสำเร็จในชีวิตเรื่อยมา

อนุสรณ์ ศรีพรหม

สารบัญ

หน้า

กิตติกรรมประกาศ ก

สารบัญ ข

สารบัญ ค

บทที่ 1 บทนำ 1

* 1. ความเป็นมาและความสำคัญของโครงงาน 1  
     1.1.1. ทำไมถึงควรทำ Web Application 2  
     1.1.2. ความเหมือนที่แตกต่างของเว็ปไซต์และแอพพลิเคชั่น 2  
     1.1.3 UX/UI คืออะไรสำคัญต่อการออกแบบเว็ปแอพพลิเคชั่นอย่างไร 3  
     1.1.4 UX/UI คืออะไร 3  
     1.1.5 ความสำคัญของ UX/UI 4
  2. วัตถุประสงค์ 5
  3. ขอบเขตของโครงงาน 6
  4. เครื่องมือที่ใช้ในการทำโครงงาน 6
  5. ประโยชน์ที่คาดว่าจะได้รับ 6

บทที่ 2 เอกสารและโครงงานที่เกี่ยวข้อง 7

2.1. Web Application 3 เว็ปแอพพลิเคชั่นตัวอย่าง 7-9

2.2. NestJS 10

2.3. MongoDB Compass 11

2.4. React 12

บทที่ 3 วิธีการดำเนินงาน 13

3.1. ศึกษาข้อมูลที่เกี่ยวข้อง 13

3.2. วางแผนการทำงาน 13

3.3. ขั้นตอนการดำเนินโครงงาน 14

3.3.2. ออกแบบ User Interface 15

3.3.3. เตรียมฐานข้อมูล 20

3.3.4. ทดสอบการเชื่อมต่อระบบ 21

บทที่ 4 ปัญหาและข้อเสนอแนะ 40

4.1. ปัญหาการทำโครงงาน 40

4.2. ข้อเสนอแนะ 40

บรรณานุกรม 41

สารบัญรูปภาพ

รูปที่ หน้า

2.1. หน้าเว็ป Keenthemes.com 7

2.2. หน้าเส็ปไซต์ในส่วน Calendar 8

2.3. หน้าเว็ปของ Google.com 9

2.4. NestJS Logo 10

2.5. MongoDB Logo 11

2.6. ตัวอย่าง MongoDB Data Base 11

2.7. React Logo 12

2.8. Concept ของ React 13

2.9. Vite Logo 13

3.1. หน้าเข้าสู่ระบบ 16

3.2. หน้าสมัครสมาชิก 16

3.3. หน้า Send Agenda(User) 17

3.4. หน้า History 17

3.5. หน้า User Profile , Lecturer 18

3.6. หน้า Contact 18

3.7. หน้า Reset Password 19

3.8. หน้า My Student 19

3.9. หน้า Approve Agenda 20

3.10. API เชื่อม Data Base 21

3.11. Data Base MongoDB 21

3.12. Backend เชื่อม API 22

บทที่ 1

บทนำ

โครงงาน “บันทึกการพบประอาจารย์ที่ปรึกษาโครงงาน”

ผู้รับผิดชอบโครงงาน นายอนุสรณ์ ศรีพรหม รหัสนักศึกษา B6130381 สาขาวิชาวิศวกรรมคอมพิวเตอร์สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ที่ปรึกษาโครงงาน

ดร.คมศัลล์ ศรีสุทธิ์

ความเป็นมาและความสำคัญของโครงงาน

เนื่องจากนักศึกษาได้เห็นว่าการที่เรารวมปฏิทินนี้มันจะอำนวยความสะดวกให้แก่ผู้ใช้งานที่เข้ามาใช้งานไม่ว่าจะเป็นนักเรียนหรือนักศึกษาที่วางแผนไม่ว่าจะไปเที่ยวหรือว่าทำกิจกรรมต่างๆในวันหยุดอย่างเช่นพวกกิจกรรมเข้าค่ายหรือว่ากิจกรรมจิตอาสา ทั้งนี้ยังมีประกาศข่าวสารจากมหาลัยก็จะ เป็นการง่ายที่นักศึกษาเพียงแค่คลิกเข้าเว็บเพื่อเช็คข่าวสาร

Web Application คือซอฟต์แวร์ชนิดหนึ่งที่อยู่ในรูปแบบของเว็บไซต์โดยตัวแอพพลิเคชั่นถูกจัดเก็บไว้ในเซิร์ฟเวอร์อินเทอร์เน็ตในการใช้งานบนอุปกรณ์ต่างๆเช่นคอมพิวเตอร์มือถือว่าจะเป็นเว็บไซต์นานๆจะถูกใช้งานเพื่อจุดประสงค์ใดแต่ถ้าเว็บไซต์ใช้งานได้มากกว่าการอ่านเนื้อหาทั่วไปเช่นพิมพ์เอกสารแต่งและเซฟเป็นไฟล์ได้ก็ถูกจัดว่าเป็น web app

โดย web Application ทั้งหลายจะถูกเชื่อมต่อกับเครือข่ายที่ใช้งานได้มีอินเตอร์เฟสหรือหน้าต่างของใช้ที่ถูกออกแบบให้ใช้งานโดยเฉพาะและสามารถเข้าถึงได้ด้วย UI ผ่าน Web Browser ซึ่งแบ่งตามการใช้งานหลายประเภทมากบางคนอาจไม่รู้ด้วยซ้ำว่ากำลังใช้เว็บอยู่

**1.1.1. ทำไมถึงควรทำ Web Application**

เว็บแอพพลิเคชั่นคือ app ที่ถูกเขียนขึ้นทำให้สามารถเปิดใช้ในเว็บบราวเซอร์ได้โดยตรงไม่ต้องโหลด Application แบบเต็มๆลงเครื่องทำให้รวมแล้วกินทรัพยากรค่อนข้างต่ำและสามารถเปิดใช้งานได้ไว้  
 ภายในตัว Web Application มันถูก Optimus ให้มีการทำงานรวดเร็วกว่าเปิด Application แบบปกติซึ่งมีหน้าตาเป็นมิตรและใช้งานค่อนข้างง่ายซึ่งปัจจุบันมีเว็บแอปพลิเคชันที่หลากหลายทั้งในรูปแบบใช้งานระดับโลกระดับมหาวิทยาลัยไปจนถึงระดับบริษัทก็มีเว็บแอพพลิเคชั่นเป็นของตัวเอง

**1.1.2. ความเหมือนที่แตกต่างของเว็บไซต์และเว็บแอพพลิเคชั่น**

ทำไมคนจึงเข้าใจผิดว่าเว็บไซต์ละ web Application นั้นเหมือนกันสิ่งที่สังเกตได้ง่ายที่สุดคือทั้งสองอย่างล้วนเกิดจากที่เดียวกันนั่นก็คือเว็บบราวเซอร์แต่หารู้ไม่ว่าเว็บไซต์ในตอนนี้ล้วนถูกปรับเปลี่ยนให้เป็น web Application กันเกือบหมดแล้วลองมาดูความแตกต่างของเว็บไซต์และแอพพลิเคชั่นมีอะไรบ้าง

1. การใช้งาน  
   WebSite คือหน้าเพจที่ถูกจัดทำขึ้นเพื่อนำเสนอข้อมูล  สารสนเทศ และมีการเชื่อมโยงข้อมูลไปยังเว็บเพจย่อยต่างๆตามรูปแบบของเว็บไซต์นั้นได้กำหนดและตั้งค่าไว้ โดยเว็บไซต์นี้เน้นให้ผู้คนเข้ามาดูเป็นหลัก  
   web Application ทำหน้าที่คล้ายเว็บไซต์สามารถปิด Application ได้ด้วยคือเน้นให้ผู้คนเข้ามาใช้งานมากกว่าดูเช่น web app สำหรับคิดเลข web app สำหรับจับเวลาแบบสำหรับแปลภาษาโดยส่วนมากแล้วจะมีความสะดวก รวดเร็ว และสบายตากว่าเว็บไซต์ปกติเนื่องจากเป็นการใช้งานในเรื่องหนึ่งเป็นหลัก
2. หน้าตาภายนอก  
   เว็บไซต์ มักเน้นความสวยงามดึงดูดบางเว็บก็มีความซับซ้อนบ้างมีลิงค์ต่างๆเพื่อให้คนเข้าสามารถเปิดเว็บเพศภายในเว็บไซต์ได้ web Application  มุ่งไปที่ความเรียบง่าย สะอาด บางครั้งหน้าเว็บแอพพลิเคชั่นก็แสดงโปรแกรมและฟังก์ชันต่างๆขึ้นมาตรงๆแบบไม่ตกแต่งอะไรมากเนื่องจากต้องการให้ผู้ใช้งานสามารถใช้งานได้อย่างรวดเร็วที่สุด
3. การทํางานเบื้องหลัง   
   Website โดยรวมแล้วหน้าเว็บไซต์แต่ละหน้ามักมีความซับซ้อนค่อนข้างต่ำ จะหนักไปในเชิงการตกแต่งและจัดรูปแบบให้เหมาะสมกับความชอบของผู้เข้าชม   
    Web Application มีความซับซ้อนและยุ่งยาก อาจต้องใช้ผู้มีเชี่ยวชาญเฉพาะทาง เพื่อจัดการและออกแบบแอปพลิเคชัน รวมถึงฟังก์ชั่นภายในให้ใช้งานได้ง่าย จะออกแบบอย่างไรให้สิ่งที่ซับซ้อน กลายเป็นสิ่งที่คนไม่เคยใช้ก็ใช้งานได้ จากที่กล่าวมาข้างต้นเป็นการอธิบายสัดส่วนหลักสำหรับการทำงานของ Web Application เท่านั้น และส่วนประกอบในการทำงานทั้งหมดอาจมีการเปลี่ยนแปลงเชิงลึกได้ตามความถนัด ความชำนาญ ไปจนถึงต้องการรายละเอียดในแบบไหน เพราะการทำ Web Application คือ การ สร้างสิ่งที่ต้องการ ซึ่งสามารถเปิดผ่าน Web Browser ได้ ดังนั้นสิ่งที่จะถูกรังสรรค์ขึ้นมาจะเป็นเกม เว็บแอปสุขภาพ ไปจนถึงติดตามข่าวซุบซิบดารา ก็เป็นไปได้ทั้งสิ้นและแทนที่คุณจะเสียเวลาทำ เว็บไซต์แล้วเสียเวลาค่าแอพพลิเคชั่นเพิ่มต่างหากจะดีกว่าหากคุณสร้าง Web Application ทีเดียว ให้เป็นได้ทั้ง App และเว็บไซต์ไปด้วยในตัวตอบโจทย์ความสะดวกและประหยัดได้อย่างครบวงจร  
    การออกแบบ Web Application ที่ดีและเป็นระบบก็เป็นอีกหนึ่งปัจจัยที่ทำให้ผู้ใช้งานนั้นให้ ความสนใจกับเว็บแอพพลิเคชั่นมากขึ้น เพราะเว็บแอพพลิเคชั่นมีความสวยงามและใช้งานง่าย จึงทำ UV/UX มาใช้ ให้ผู้ใช้งานมีประสบการณ์ในการใช้งานที่ดีเหตุนี้การนำความรู้และความเข้าใจทางด้าน จึงเป็นเรื่องที่สำคัญเช่นกัน

**1.1.3 UX/UI คืออะไร สําคัญต่อการออกแบบเว็บแอพพลิเคชั่นอย่างไร**

ในการออกแบบเว็บแอพพลิเคชั่น UX คือ User Experience และ UI หรือ User Interface เป็นเรื่องที่สำคัญมาก เพราะไม่ได้คำนึงถึงแค่ความสวยงามของเว็บแอพพลิเคชั่น เพียงอย่างเดียว แต่ยังคำนึงถึงการใช้งานอีกด้วย ซึ่งทำให้หน้าเว็บแอพพลิเคชั่นตอบสนอง ความต้องการของผู้ใช้งานในรอบด้าน

**1.1.4 UX/UI คืออะไร?**

UX ย่อมาจากคําว่า User Experience การออกแบบด้านนี้ จะเกี่ยวข้องกับกระบวนการ ที่สินค้าหรือบริการมีปฏิสัมพันธ์กับผู้ใช้ โดยเน้นการเก็บข้อมูลเพื่อเอากลับมาพัฒนาตัวสินค้าหรือ บริการต่อไป เช่น การทำ Usability Testing หรือการให้ผู้ใช้ได้ทดลองใช้ผลิตภัณฑ์หรือบริการ แล้วนำข้อมูลที่ได้มาวิเคราะห์ (ยกตัวอย่างเช่น Google Analytics) UX ถือเป็นหัวใจของการ ออกแบบทุกประเภท เพราะเป็นการออกแบบเพื่อให้ผู้ใช้หรือลูกค้าขององค์กรรู้สึกดีและพึงพอใจ ต่อผลิตภัณฑ์สินค้าหรือบริการ

UI นั้น ย่อมาจากคำว่า User Interface เน้นไปที่เรื่องการออกแบบหน้าจอหรือ แพลตฟอร์มต่างๆ เช่น เว็บไซต์ แอพพลิเคชั่น เกมส์ รวมไปถึงอุปกรณ์ที่จับต้องได้ เช่น จอทัชสกรีน เป็นต้น ให้มีความสวยงาม น่าใช้ เป็นมิตร เข้าใจง่าย (User-friendly) เป็นต้น การ ออกแบบหน้าตาให้สวยงาม โดดเด่น สร้าง first impression ที่ดีให้กับผู้ใช้งานไม่ว่าจะเป็นเรื่อง ของสี ฟ้อนต์ การจัดวางองค์ประกอบต่างๆ คำนึงถึงภาพลักษณ์และความสวยงามเป็นสำคัญครับ

**1.1.5 ความสำคัญของ UX/UI**

ความรู้ ความเข้าใจเรื่อง UX/UI ช่วยให้รู้จักผู้ใช้ แล้วนำมาต่อยอดในการทำการตลาด เพื่อทำให้ผู้ใช้เข้าถึงสินค้าได้อย่างประทับใจคือ ประทับใจแต่แรกเห็น ในแง่ของโลกออนไลน์ที่มี การสื่อสารผ่านหน้าจอ ทั้ง UX และ UI จึงสำคัญมากประมวลได้ คือ

1. ช่วยเรื่องการทําเว็บไซต์ ซึ่งต้องมีทั้งศาสตร์และศิลป์ ออกแบบดีคือ ต้องให้ง่ายต่อการเข้าถึง ไม่ซับซ้อน เกิดความประทับใจตั้งแต่เริ่มต้นการเข้าชมเว็บไซต์

2. หน้าตาของเว็บไซต์ สวยงาม จัดวางตำแหน่งของสิ่งที่จำเป็นได้ดี เช่น ตำแหน่งค้นหาข้อมูลที่ต้องการต้อง เห็นชัด ตัวหนังสืออ่านง่าย เป็นต้น

3. ตอบโจทย์ที่ผู้ใช้ต้องการ

4. สร้างความน่าเชื่อถือ ชื่นชอบตั้งแต่การทำเว็บไซต์ ก็จะพลอยยินดีต่อการนำเสนอสินค้าไปด้วย ความสำคัญของทั้ง UX/UI สำคัญต่อการทำการตลาด เพราะนอกจากจะเข้าถึงตัว ก็จะรู้วิธี เข้าถึงใจ จนถึงขั้นตัดสินใจเป็นลูกค้าได้ ซึ่งถ้าเจ้าของกิจการเข้าใจสองเรื่องนี้เป็นอย่างดี จะ หานักออกแบบ นักวิจัย ทำการวิจัยผลิตภัณฑ์เพื่อให้ตอบโจทย์ที่จะมีผลสัมฤทธิ์ตามที่ต้องการ ในที่สุดในการออกแบบเว็บไซต์ก็ทำให้เว็บแอพพลิเคชั่นของเรามีความน่าเชื่อถือมากขึ้นด้วย

ที่มา:

<https://www.dmit.co.th/th/%E0%B8%82%E0%B9%88%E0%B8%B2%E0%B8%A7%E0%B8%AA%E0%B8%B2%E0%B8%A3/ux-vs-ui/>

ตามที่กล่าวมาข้างต้น อีกทั้งการออกแบบหรือพัฒนาเว็บไซต์ต่างๆ เหล่านี้ทางผู้พัฒนาสามารถ ออกแบบเว็บแอพพลิเคชั่นออกมาในแบบที่ผู้ใช้งานต้องการ และเป็นมิตรต่อการใช้งานของผู้ใช้งาน

1.2. วัตถุประสงค์

พัฒนาเว็บไซต์แอพพลิเคชั่น TotalCalendar เพื่อให้ผู้คนที่ต้องการติดตามการประกาศหรือข้อมูลข่าวสารต่างๆสามารถเข้ามาหาข้อมูลเกี่ยวกับมหาลัยได้ง่ายยิ่งขึ้นและผู้คนทั่วไปสามารถดูหรือสอดส่องวันหยุดต่างๆรวมถึงวันพระได้อีกด้วยทั้งนี้วัตถุประสงค์ก็เพื่ออำนวยความสะดวกในเรื่องข้อมูลข่าวสารให้แก่ผู้ใช้งาน

โดยออกแบบระบบให้มีวัตถุประสงค์ในการใช้งานดังนี้

1. ผู้ใช้ที่ยังไม่สมัครเป็นสมาชิกสามารถเข้ามาดูข้อมูลข่าวสารได้ได้แก่ ปฏิทิน, ประกาศของมาหาลัย, ประกาศเกี่ยวกับค่าย, ประกาศเกี่ยวกับกิจกรรม,คอมเม้นท์
2. ผู้ใช้สามารถสมัครเข้ามาในระบบได้
3. ผู้ใช้ในระบบ(User)สามารถที่จะส่งคำขอเพื่อเป็น User Admin ของมหาลัยนั้นๆได้
4. ผู้ใช้ในระบบ(User)สามารถที่จะแสดงความคิดเห็นในประกาศต่างๆได้เช่นประกาศของมหาลัย ค่าย หรือกิจกรรมต่างๆ
5. ผู้ใช่ในระบบ(User)สามารถที่จะกดถูกใจให้กับความเห็นของ User อื่นได้และสามารถกดถูกใจให้กับประกาศหรือค่ายหรือกิจกรรมต่างๆได้
6. ผู้ใช้ในระบบ(User) สามารถที่จะโพสสร้างกิจกรรมต่างๆได้
7. ผู้ใช้ในระบบ(User) สามารถที่จะแก้ไขโพสกิจกรรมได้
8. ผู้ใช้ในระบบ(User) สามารถแก้ไขข้อมูลส่วนตัวได้
9. ผู้ใช้ในระบบ(User) สามารถเปลี่ยนรหัสผ่านได้
10. ผู้ใช้ในระบบ(User) สามารถให้คำแนะนำในการพัฒนาเว็ปไซต์เพิ่มเติมได้
11. ผู้ใช้ในระบบ(User) สามารถรับรหัสผ่านใหม่ได้เมื่อมีการลืมรหัสผ่าน
12. ผู้ใช้ในระบบ(User Admin) สามารถอัพโหลดข้อมูลเกี่ยวกับวันต่างๆในปฏิทินได้
13. ผู้ใช้ในระบบ(User Admin) สามารรับรหัสผ่านใหม่ได้เมื่อมีการลืมรหัสผ่าน
14. ผู้ใช้ในระบบ(User Admin) สามารถสร้างโพส แก้ไข ลบ เกี่ยวกับประกาศ ค่าย กิจกรรมของมหาลัยตนเองได้
15. Admin สามารถแก้ไขบทบาทให้แก่ User, User Admin ได้
16. Admin สามารถที่จะแก้ข้อมูลส่วนตัวของตัวเองได้
17. Admin สามารถอัพโหลดข้อมูลเกี่ยวกับวันต่างๆในปฏิทินได้
18. Admin สามารถรับรหัสผ่านเมื่อลืมรหัสผ่านได้

**1.3. ขอบเขตโครงงาน**

พัฒนาเว็บไซต์ “เว็ปรวมปฏิทิน” สำหรับเป็นที่กระจายข่าวสารให้แก่ผผู้ใช้โดยให้มีผู้ใช้งานคือ Admin UserAdmin User และผู้ใช้ ทุกบทบาทสามารถดูข้อมูลข่าวสารได้ บทบาท Admin สามารถเปลี่ยนให้แก่ User ได้และ UserAdmin สามารถโพส แก้ไข ลบประกาศต่างๆรวมถึงค่ายและกิจกรรมได้รวมถึงสามารถ Comment และกด Like ให้กับกิจกรรมต่างๆรวมถึงประกาศค่ายได้

User สามารถสร้างโพสในกิจกรรมต่างๆได้รวมถึงกด like และ Comment ได้ด้วย

**1.4. เครื่องมือที่ใช้ทำโครงงาน**

1.4.1 Window 10

1.4 (Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.19 GHz)  
 1.4.2. Visual Studio Code

1.4.4. Google Chrome, Microsoft Edge, Postman

1.4.5. Draw.io

1.4.6. Command, PowerShell

1.4.7. MongoDB Compass

1.4.8. Front Awsome

1.4.9. CSS

1.4.10. React

1.4.11. NestJS

1.4.12. Figma

1.4.13. Github

1.4.14. Icon Editor

**1.5. ประโยชน์ที่คาดว่าจะได้รับ**

1. ผู้คนที่ใช้งานจะมีความสะดวกในเรื่องการหาข้อมูลข่าวสารได้ดียิ่งขึ้น

2.  Application มีความเข้าใจง่ายต่อการใช้งานเพิ่มความสะดวกสบายสบายตาให้ผู้ชายยิ่งขึ้น

3. ทั้งนี้ยังสามารถใช้งานในเว็บบราวเซอร์ทั่วไปได้ลดขั้นตอนในการดาวน์โหลดและติดตั้ง

4. การเพิ่มศักยภาพของนักศึกษาหลังจากทำโปรเจคนี้สำเร็จ

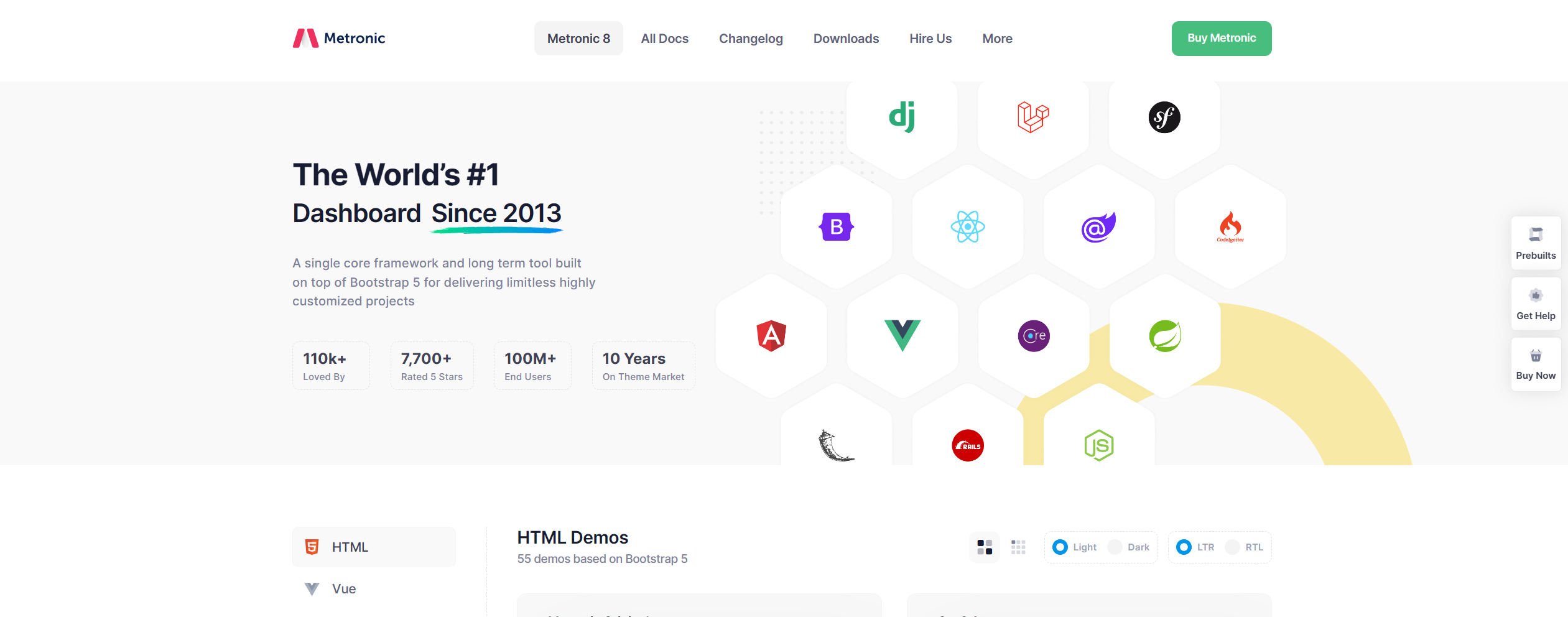
# บทที่ 2

**เอกสารและโครงงานที่เกี่ยวข้อง**

ในการทำ Web Application TotalCalendar ครั้งนี้ ผู้จัดทำได้ศึกษาข้อมูลเกี่ยวกับเว็บไซต์ต่างๆ 3 เว็บแอพพลิเคชั่นในอินเตอร์เน็ต 3 เว็ป เพื่อเป็นแนวทางในการออกแบบ web Application และ ในครั้งนี้

นำเสนอตามหัวข้อดังนี้

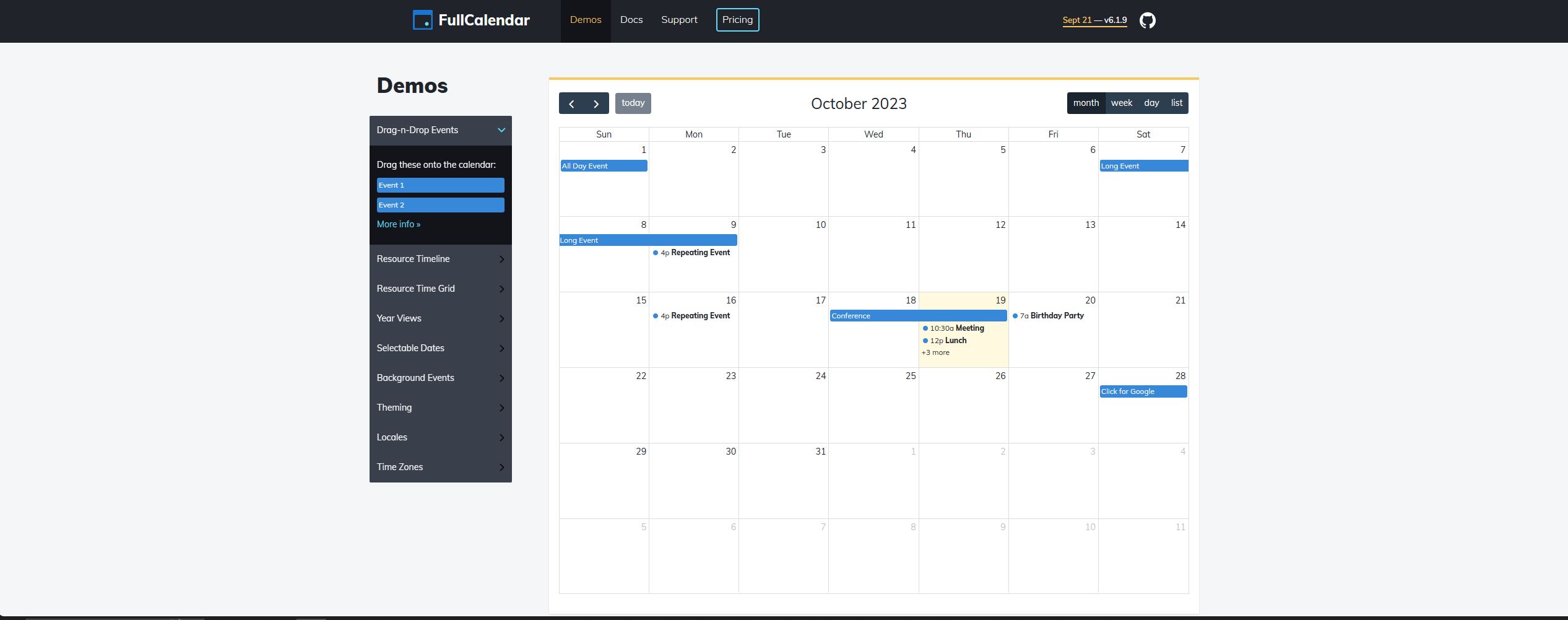
**2.1. Web Application 3 เว็บแอพพลิเคชั่นตัวอย่าง**

1. https://www.keenthemes.com /  
   

รูปที่ 2.1 หน้าเว็ป keenthemes.com

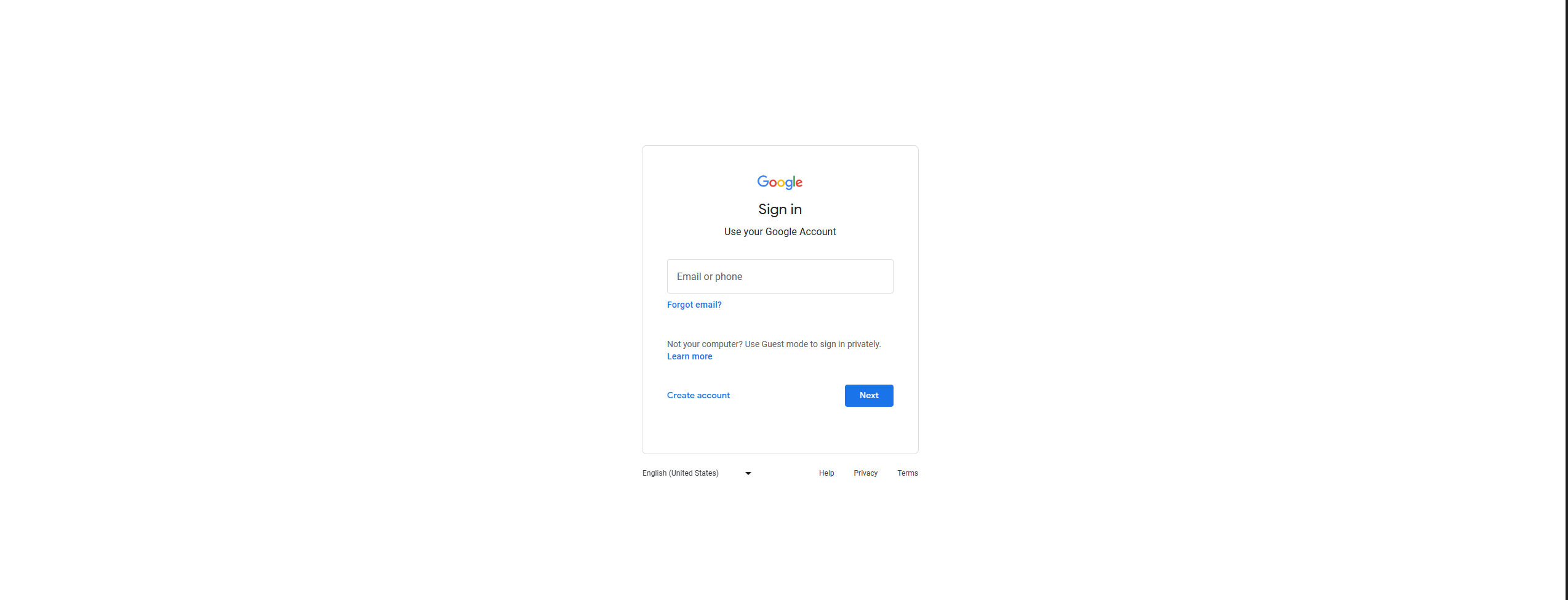
การดีไซ์ที่ดูเรียบง่าย ของเว็ปไซต์

2. เว็บไซต์สำหรับ ตัวอย่าง Fullcalendar (https://fullcalendar.io)



รูปที่ 2.2. หน้าเว็ปไซต์ในส่วน Calendar

ได้ Fullcalendar เป็นต้นแบบของการอกแบบปฏิทิน

3. เว็ป Google.com (https://www.Google.com/)  


รูปที่ 2.3 หน้าเว็ปของ Google.com  
ได้ Google Login เป็นต้นแบบของการ Login

**2.2. NestJS**



**รูปที่ 2.4. NestJS Logo**

NestJS (opens new window)เป็น Node.js Framework สำหรับเขียน Web Backend เขียนโดยใช้ภาษา TypeScript, JavaScript ความสามารถของ NestJS ก็คือ มันมาพร้อมกับโครงสร้างที่เหมาะสมกับการ Scale แถมยังพร้อมทำ DI (Dependency Injection) ได้อย่างง่ายดายด้วย และจะดีขึ้นไปอีกถ้าคุณเคยเขียน Angular มาก่อน เพราะโครงสร้างรูปแบบการทำงานจะเหมือนๆกันเลย ต่างกันที่ Angular เราเขียนเพื่อใช้ใน Frontend   
ตัว NestJS ถูกสร้างโดยมีพื้นฐานของ Express (opens new window)ทำงานอยู่เบื้องหลัง หรือจะเปลี่ยนให้ NestJS ไปใช้ Fastify (opens new window)แทนก็ยังได้

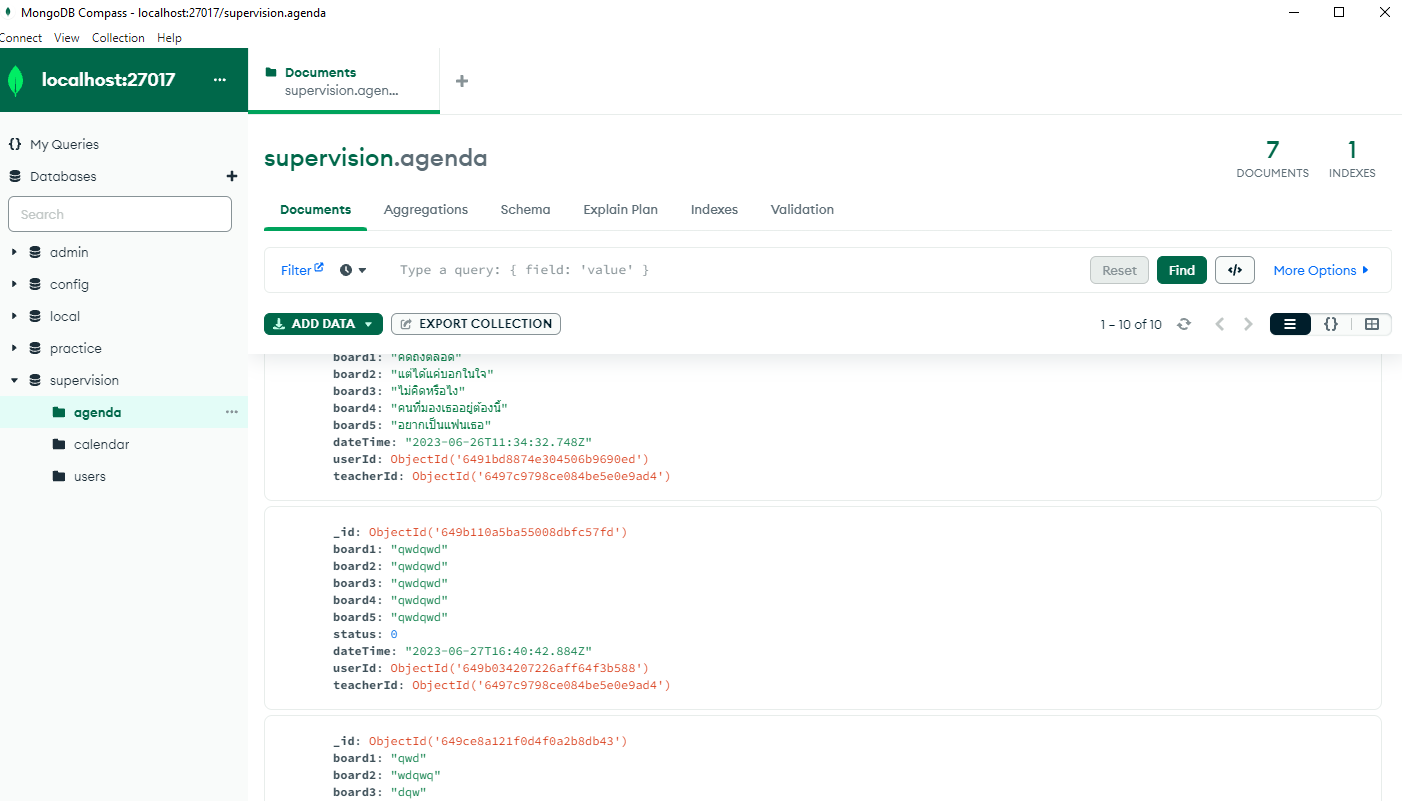
องค์ประกอบของ NestJS จะมีคือ Module , Controller , Service   
**Module** : { จะเป็นหน้าที่เชื่อมต่อกับฐานข้อมูลเพื่อเรียกใช้งานในส่วนต่างๆไม่ว่าจะเป็น Controllers Services หรือพวก Injectable }  
Controllers : {จะทำหน้าที่กำหนด Path และเรียกใช้งาน Service}  
Services : คือส่วนที่ค้าหาหรือ Map ข้อมูลที่ตัว DataBase แล้วใช้เงื่อนไขต่างๆเพื่อสกัดข้อมูลออกมา

**2.3. MongoDB Compass**



รูป 2.5. MongoDB Logo

**MongoDB** เป็นฐานข้อมูลแบบ NoSQL ที่ถูกพัฒนาขึ้นโดยบริษัท MongoDB Inc. ซึ่งมีลักษณะการจัดเก็บข้อมูลแบบ Document-Oriented คือใช้เอกสาร JSON ในการจัดเก็บข้อมูล โดย MongoDB จะไม่มีโครงสร้างแบบตาราง (Table) อย่าง MySQL เพื่อให้การเพิ่มเติมข้อมูล แก้ไข และลบข้อมูลใน MongoDB สามารถทำได้ง่ายและรวดเร็ว

ตัวอย่างการใช้งาน:   


รูป 2.6. ตัวอย่าง MongoDB Data Base

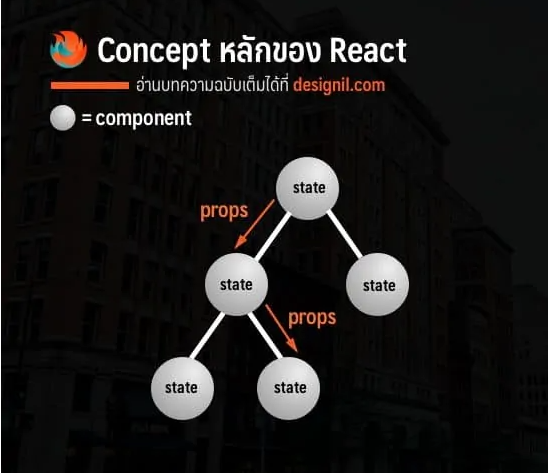
2.4. React

  
รูป 2.7. React Logo

React คือ JavaScript Library ที่ทีม Facebook เป็นคนพัฒนาขึ้นมา และเปิดให้คนทั่วไปนำมาใช้ฟรี ซึ่งเว็บไซต์ในปัจจุบันของ Facebook.com ก็ใช้ React อยู่เช่นกันครับ

สรุปคือ คอนเซปต์ที่เราต้องรู้เพื่อเขียน React หลัก ๆ มีแค่ 3 Concept เท่านั้นเอง

1. **Component** – ส่วนต่าง ๆ ในเว็บเราจะมองเป็น Component
2. **State** – ข้อมูลที่อยู่ใน Component แต่ละชิ้น เราเรียกว่า State
3. **Props** – ข้อมูลที่ถูกส่งต่อจาก Component ชั้นบนลงไปชั้นล่าง เราเรียกว่า Props (Properties)



รูป 2.8. Concept ของ React  
สำหรับการเขียน Component นั้น ก็เหมือนกับการเขียน HTML เลยครับ React ใช้สิ่งที่เรียกว่า **JSX** ในการแสดงผลเว็บไซต์ หน้าตาจะเหมือน HTML มาก แตกต่างตรงเราเขียนเข้าไปในไฟล์ JavaScript แทนไฟล์ HTML ทำให้เราสามารถเล่นอะไรกับมันได้มากกว่า

2.5 Vite (https://vitejs.dev/)



รูป 2.9. Vite Logo

Vite คือ Frontend Tooling ที่ช่วยอำนวยความสะดวกในการขึ้นโปรเจ็กต์อย่างรวดเร็ว โดยที่เราเลือก option แค่ไม่กี่ข้อ ก็จะได้ project bundle มาพร้อมใช้งานทันท

บทที่ 3

**วิธีการดำเนินงาน**

ในวิธีการดำเนินการนี้ เพื่ออธิบายการวางแผนการดำเนินงานของผู้จัดทำ โดยได้ทำการศึกษาข้อมูลที่เกี่ยว ของที่นำมาเป็นตัวอย่างข้อมูลในการทำงาน และอธิบาย Features และ Function ต่างๆ แบบพอสังเขป ว่ามีการ ทำงานอย่างไรและผู้ใช้งานสามารถใช้งานได้อย่างไร และได้อธิบายวิธีการสร้างตารางข้อมูลในรูปแบบ NoSQL โดยใช้ MongoDB และมีรูปภาพตัวอย่างการใช้โปรแกรมต่างๆ โดยมีวิธีการดำเนินการดังนี้

**3.1. ศึกษาข้อมูลที่เกี่ยวข้อง**

1. เว็บไซต์ตัวอย่างที่เป็นแนวทางในการออกแบบ 3 เว็บไซต์บนอินเทอร์เน็ต ได้แก่

1.1. keenthemes

1.2. Sussex

1.3.  CodeCademy

1. Framework ที่นำมาใช้คือ

2.1. Bootstrap 5

2.2. React

2.3. MongoDB

2.4. NestJS

1. โปรแกรมจำลอง Server ที่นำมาใช้คือ  
   3.1. Express.js

3.2. **วางแผนการทํางาน**

1. คิดหัวข้อที่จะทำ

2. ระบุขอบเขตของระบบที่จะทำ

3. ออกแบบฟังก์ชันและฟีเจอร์

4. ออกแบบฐานข้อมูล (Database และสร้างฐานข้อมูลใน MongoDB

5. ออกแบบ User Interface และเริ่ม Coding ใน VS Code

6. ทดลองโปรแกรมและทดลองฟังก์ชันต่างๆ ภายในระบบ

7. แก้ไขโปรแกรมและฟังก์ชันที่เกิดข้อผิดพลาด

**3.3 ขั้นตอนการดำเนินโครงงาน**

3.3.1. กำหนดขอบเขตการทำงาน

ระบุ Feature และ Function ของระบบเว็บแอพพลิเคชั่น TotalCalendarกำหนดขอบเขตของการทำงานโดย ระบบเว็บแอพพลิเคชั่น TotalCalendarมีความสามารถ ( Feature) ทั้งหมด 22 Feature

**Actor ที่ 1 User**

* สมัครสมาชิก
* เข้าสู่ระบบ
* ขอบทบาท User Admin
* ดูประกาศ, กิจกรรม, ค่าย
* คอมเม้นท์ใน ประกาศ,กิจกรรม,ค่าย
* แก้ไขข้อมูลโปรไฟล์
* Reset Password
* คอมเม้นท์
* โพสกิจกรรม
* ดู Event ต่างๆในปฏิทิน

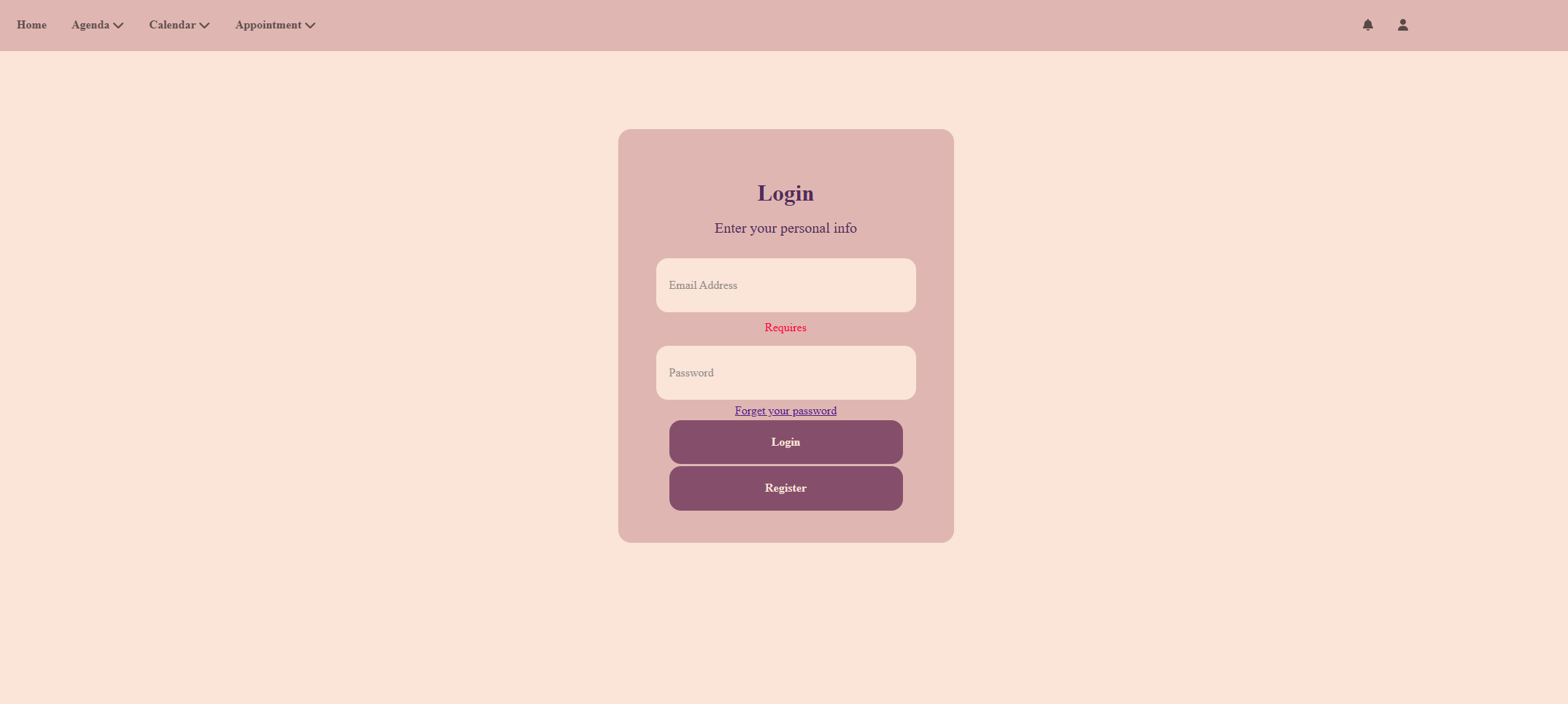
**Actor ที่ 2 User Admin**

* เข้าสู่ระบบด้วยบทบาทแอดมินของหาลัย
* แก้ไขข้อมูลโปรไฟล์
* ดูประกาศ, กิจกรรม, ค่าย
* คอมเม้นท์ใน ประกาศ,กิจกรรม,ค่าย
* Reset Password
* แก้ไขข้อมูล Calendar ของระบบ

**Actor ที่ 3** Admin

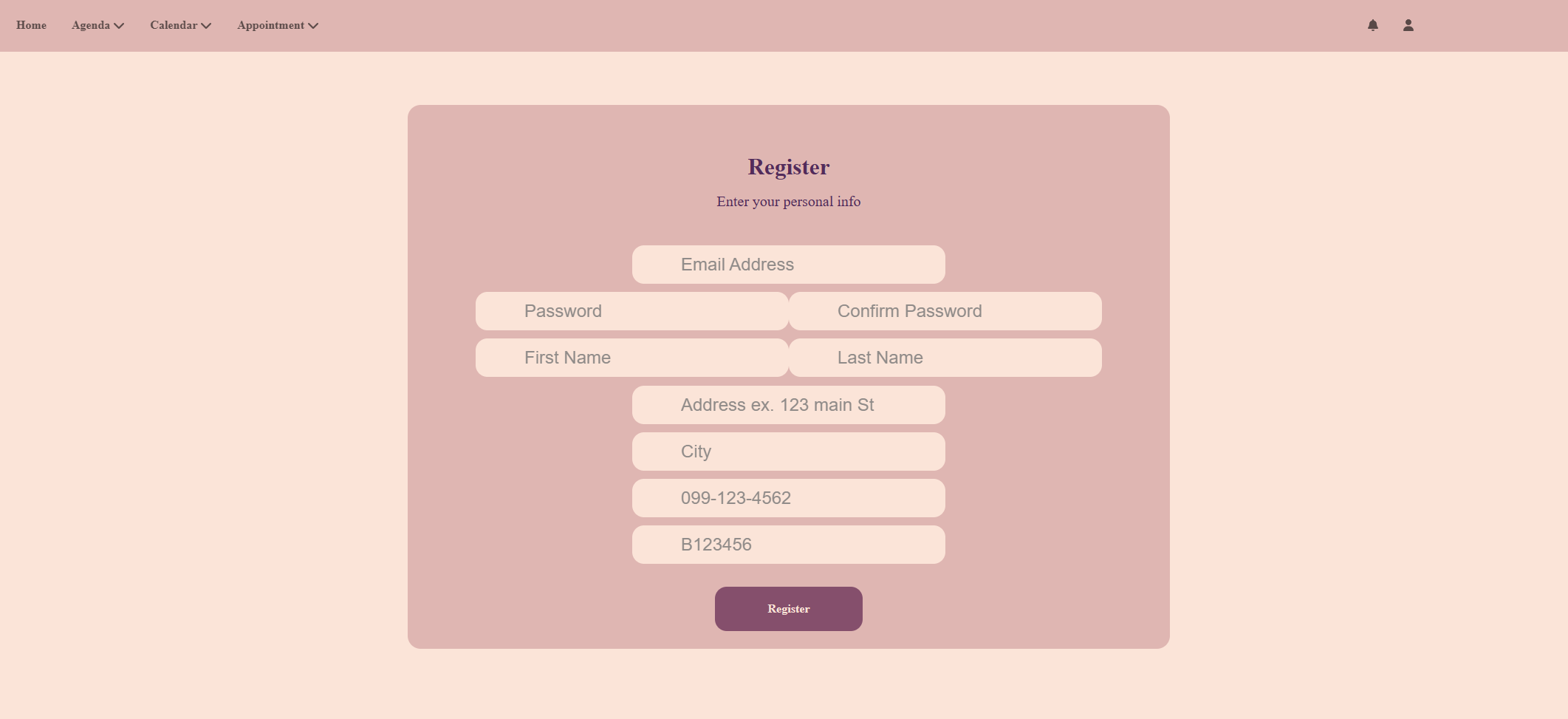
* เข้าสู่ระบบด้วยบทบาทแอดมิน
* แก้ไขบทบาทให้ User, User Admin
* ดูประกาศ, กิจกรรม, ค่าย
* แก้ไขข้อมูลโปรไฟล์
* คอมเม้นท์ใน ประกาศ,กิจกรรม,ค่าย
* Reset Password

3.3.2 ออกแบบ User Interface โดยใช้ React

- หน้าเข้าสู่ระบบ (Admin, Lecturer, User)  


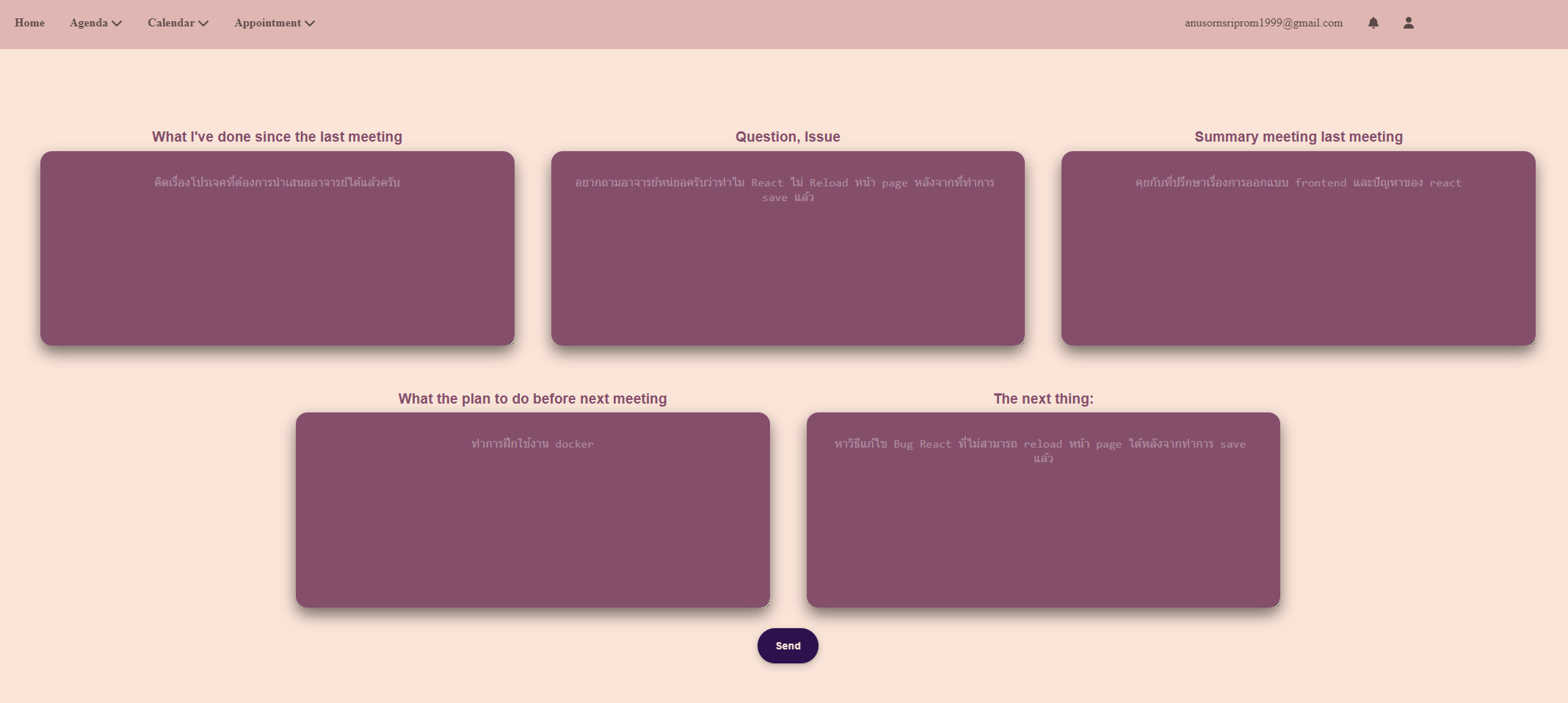
รูปที่ 3.1. หน้าเข้าสู่ระบบ

- หน้าสมัครสมาชิก (Lecturer, Users)



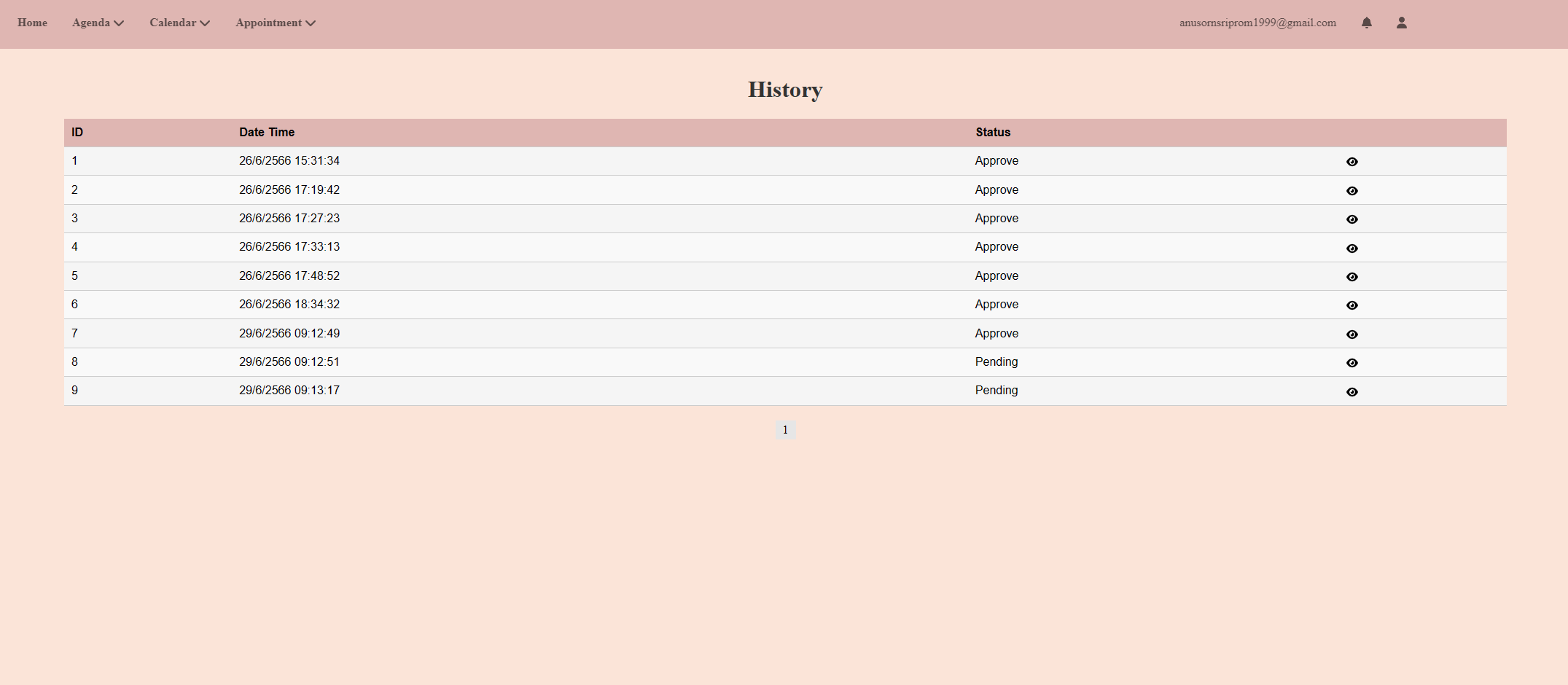
รูปที่ 3.2. หน้าสมัครสมาชิก

- หน้า Send Agenda (User)



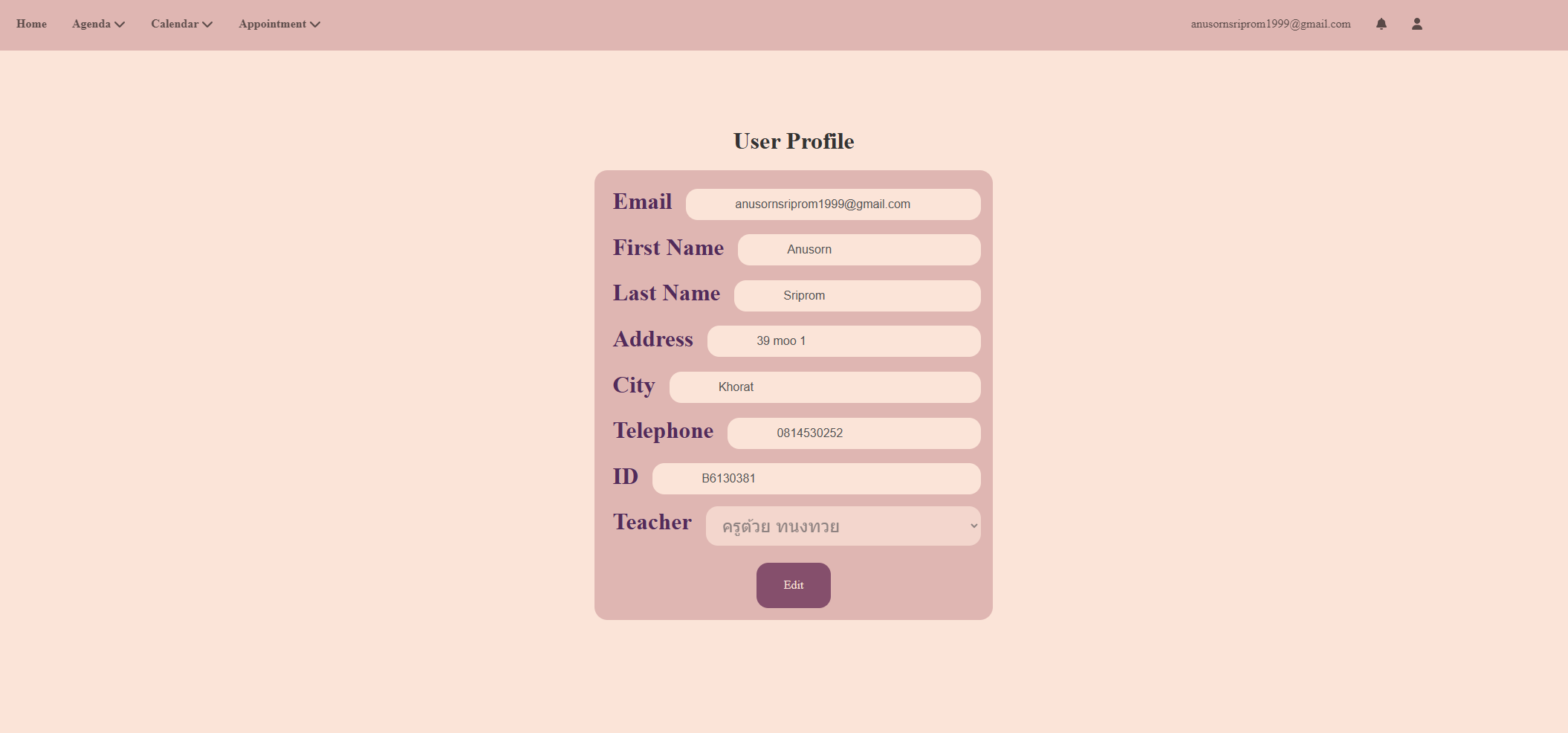
รูปที่ 3.3. หน้า Send Agenda

- หน้า History



รูปที่ 3.4. หน้า History

- หน้า User Profile



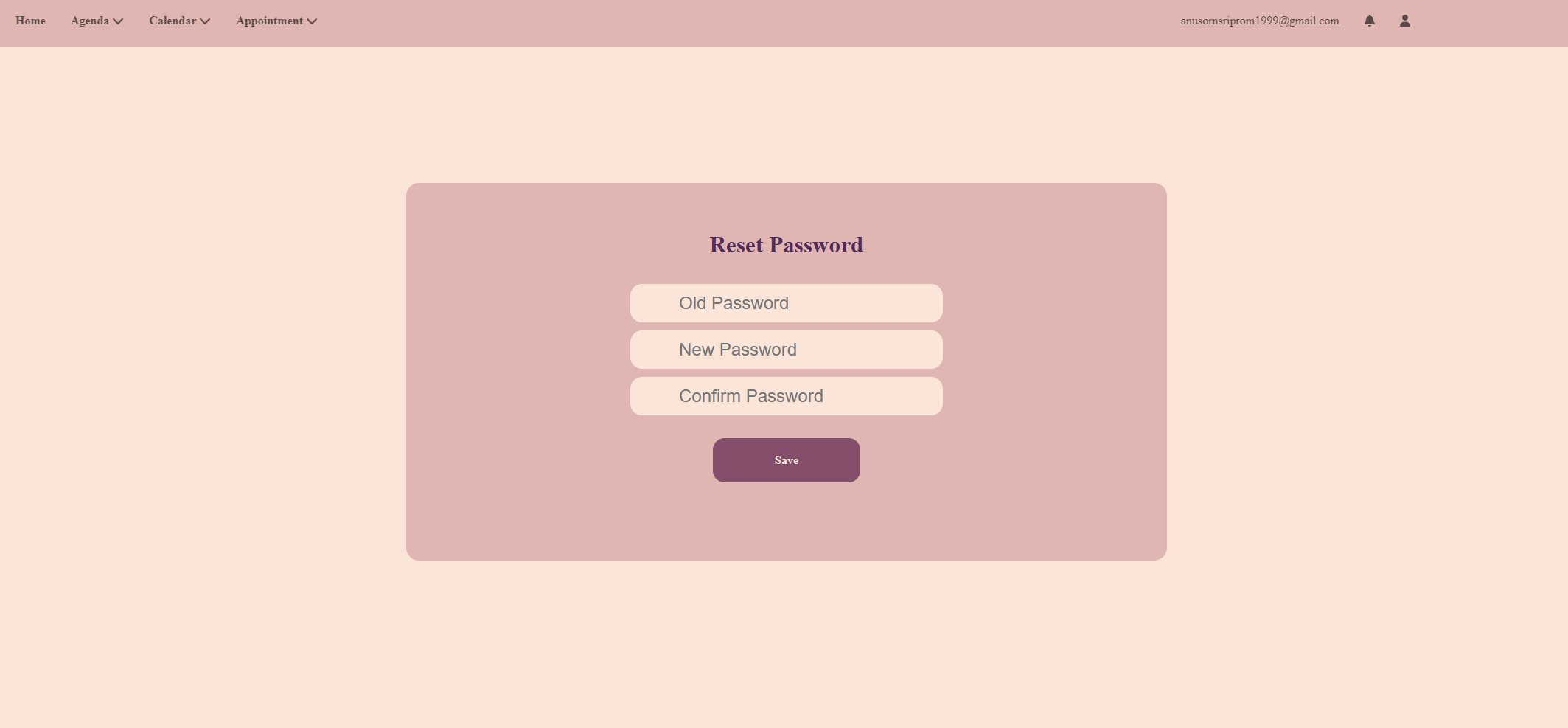
รูปที่ 3.5. หน้า User Profile, Lecturer

- หน้า Contact

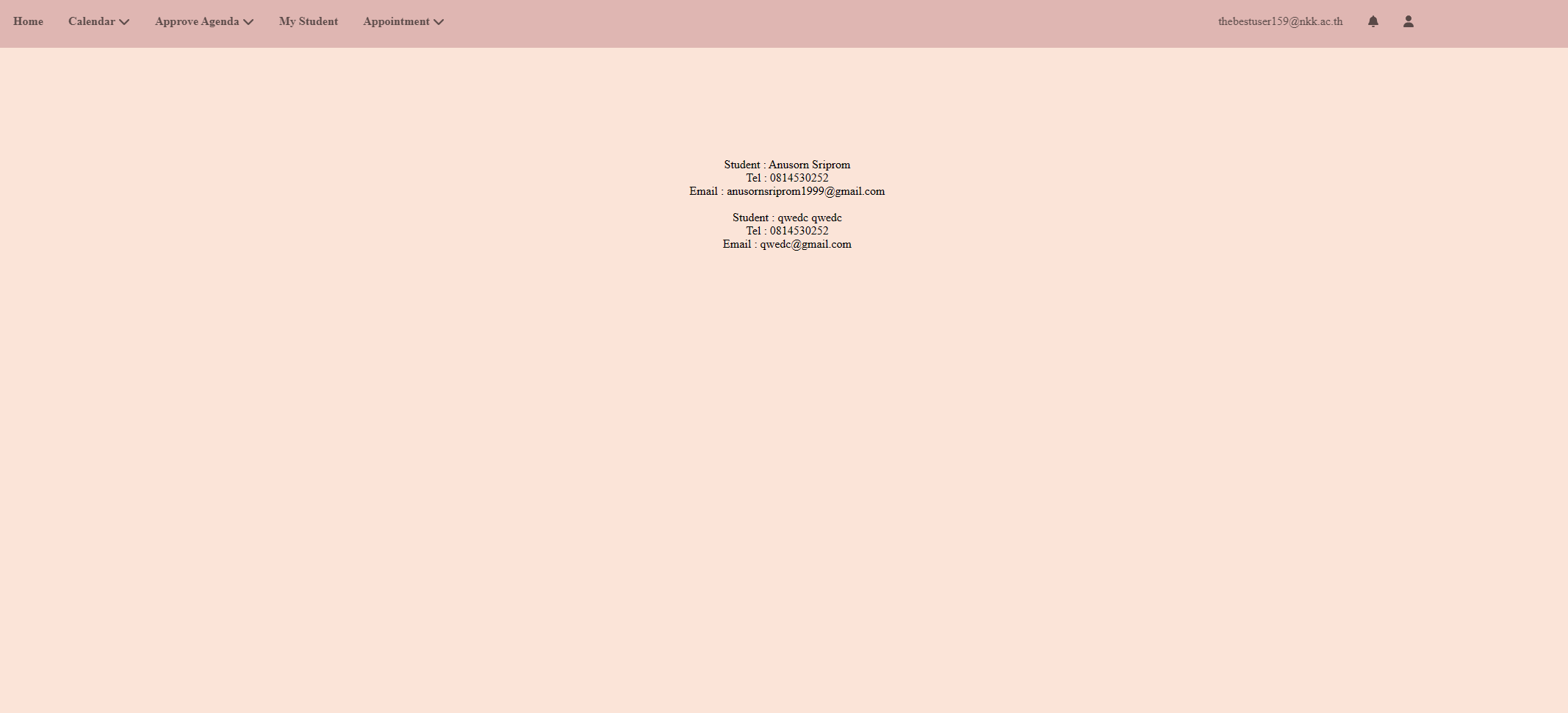


รูปที่ 3.6. หน้า Contact

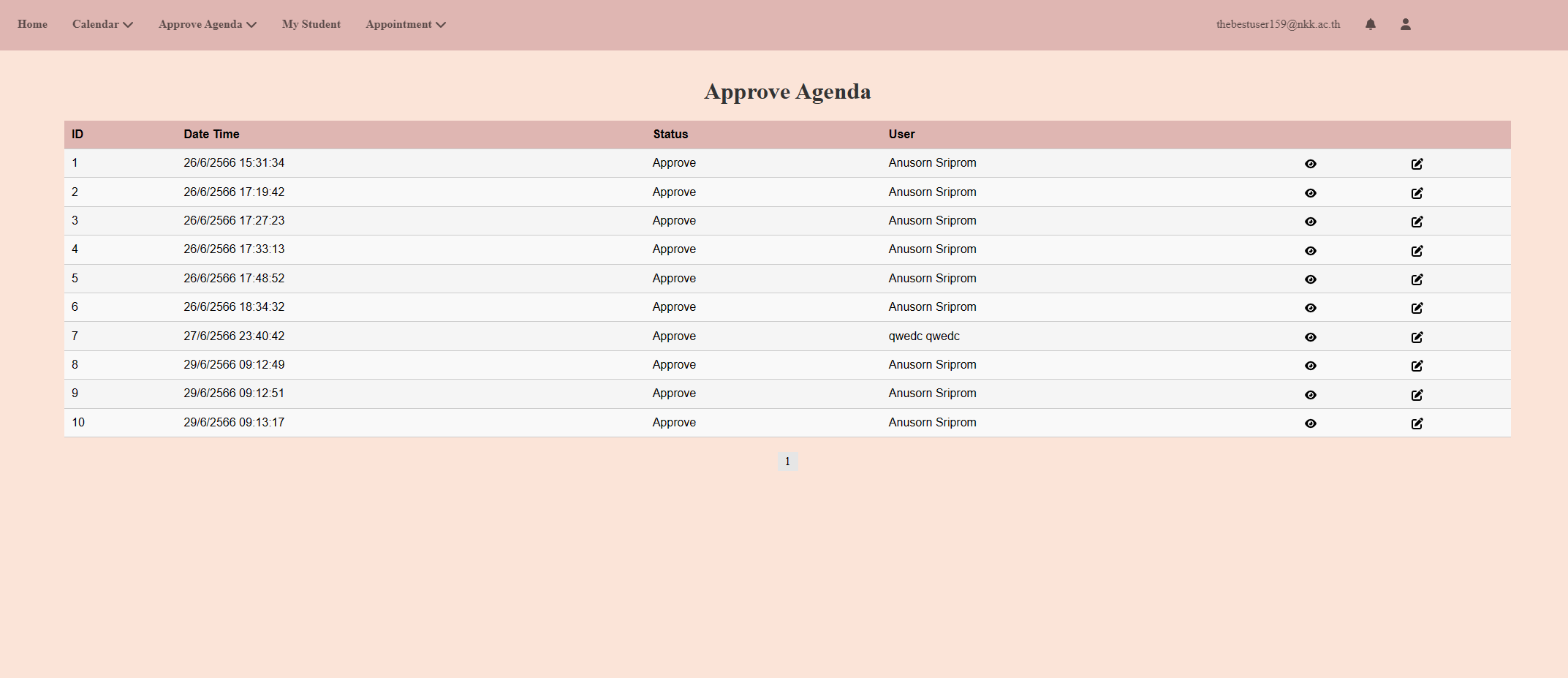
* หน้า Reset Password

  
รูปที่ 3.7. หน้า Reset Password

- หน้า My Student

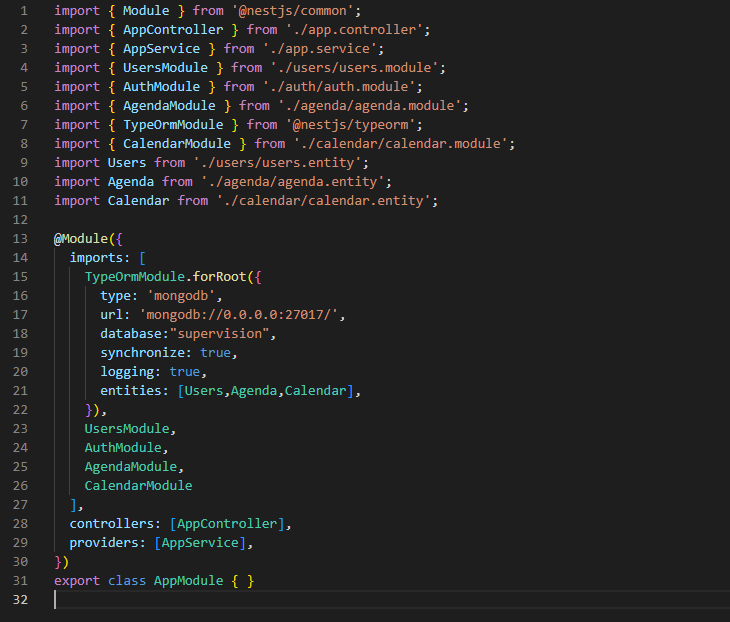
  
รูปที่ 3.8. หน้า My Student

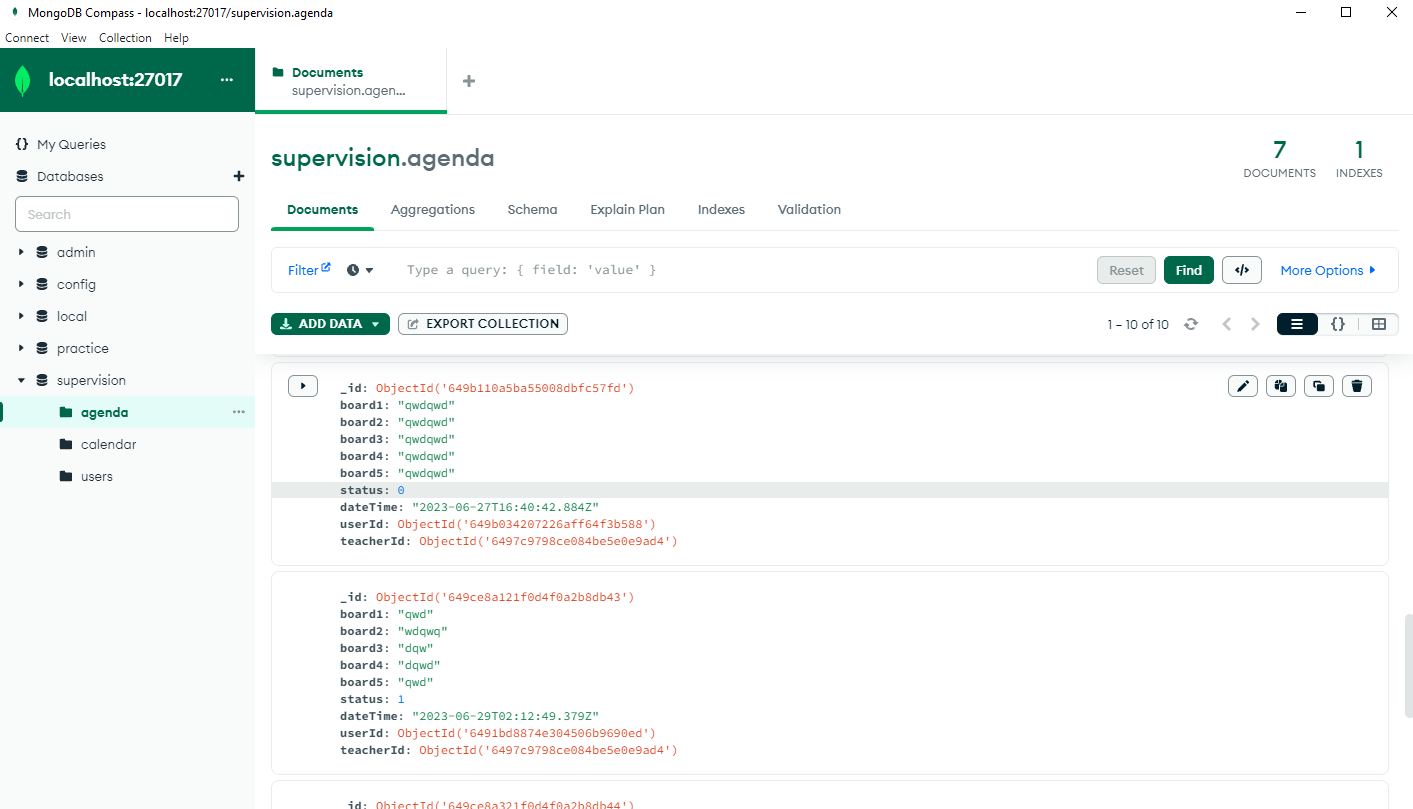
* หน้า Approve Agenda



รูปที่ 3.9. หน้า Approve Agenda

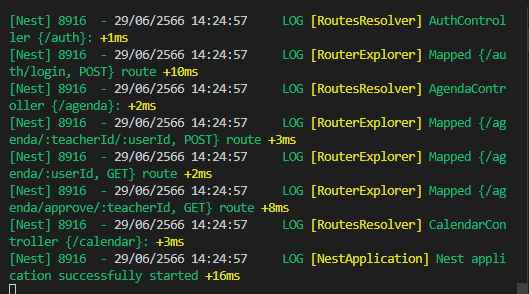
* + 1. เตรียมฐานข้อมูล  
       โดยใช้ NestJS เป็นตัวเชื่อมต่อกับฐานข้อมูลด้วย TypeORM เพื่อที่จะไป Map ข้อมูลต่างๆจากตัว DataBase โดยใช้ตัว Repository เป็นตัว Map เมื่อใช้ TypeORM ก็เลยต้องสร้าง Entyty, DTO เพื่อที่จะได้ Create, Get , Update, Delete ได้



รูป 3.10 API เชื่อม Data Base  


รูป 3.11 Data Base MongoDB

3.3.4 ทดสอบการเชื่อมต่อระบบ

โดยทำการเรียก API เชื่อมกับ DataBase ด้วยคำสั่ง npm run start  


รูป 3.12 Backend เชื่อม API

**บทที่ 4**

**ปัญหาและข้อเสนอแนะ**

จากการดำเนินศึกษาข้อมูลเกี่ยวกับการทำงานของ NoSQL, Angular, nestJS เพื่อทำการจัดเก็บข้อมูลแล้วสร้างเว็บไซต์ การศึกษาเกี่ยวกับการทำงาน Bootstrap และ CSS เพื่อใช้ในการสร้าง Style ต่างๆเข้าเว็บไซต์ให้สวยงามมากินขึ้น ดูในการศึกษาข้อมูลต่างๆ เพื่อดำเนินการทำโปรแกรมในครั้งนี้ได้มีปัญหาต่างๆ และมีข้อเสนอแนะในการพัฒนาโปรแกรมสำหรับพัฒนาโปรแกรมในอนาคตเพื่อใช้งานจริงดังนี้

**4.1. ปัญหาในการทำโครงงาน**

* เนื่องจากต้องการความรู้ที่จะเขียนให้ Project ดียิ่งขึ้นจะมีการศึกษาเพิ่มเติมเพื่อใช้ในแบบต่างๆจึงทำให้เสียเวลาในการทำโปรเจค
* การเชื่อมต่อกับ Google Calendar  ซึ่งต้องมีการศึกษา API ต่างๆเพิ่มเติมและยังมีปัญหาในเรื่องการสร้าง credenrial ซึ่งทำให้เสียเวลาในการเชื่อมต่อ API และต้องศึกษาเพิ่มเติมไม่ว่าจะเกี่ยวกับ Google developer เพื่อเปิดใช้งานให้กับ Project ที่สร้างขึ้นจึงทำให้เสียเวลาในการศึกษาเพิ่มเติมเกี่ยวกับ Google Calendar หน้า

**4.2. ข้อเสนอแนะ**

จากที่กล่าวไว้ข้างต้น feature และ funtion ต่างๆสามารถที่จะพัฒนาต่อได้ผู้จัดทำจึงขอเสนอแนะแนวทางพัฒนาต่อโดยสังเขปดังนี้

* การสมัครควรจะสามารถสมัครโดยเชื่อมกับ Google Account ได้
* ในอนาคตอาจจะมีฐานข้อมูลใหม่ในเรื่องการเก็บรูปภาพรูปประจำตัวรูปโปรไฟล์ของ User
* ในการปรับปรุงหน้า calendar ควรจะยกทั้งหน้า calendar มาใส่ไว้ใน App เพื่อที่จะไม่ต้องเปิดต้องเว็บเช่นเว็บ TotalCalendarกับเว็บ Google calendar พร้อมกันควรจะเปิดแค่ TotalCalendarแล้วสามารถเล่นได้เหมือน Google calendar ด้วย

**บรรณานุกรม**

React framework (v.14.0.5) [ออนไลน์]. สืบค้นเมื่อวันที่ 5 พฤษภาคม 2566

แหล่งที่มา : https://angular.io/

NestJS framework [ออนไลน์]. สืบค้นเมื่อวันที่ 3 มิถุนายน 2566

แหล่งที่มา : <https://nestjs.com/>

Google Calendar for Developer [ออนไลน์]. สืบค้นเมื่อวันที่ 15 มิถุนายน 2566

แหล่งที่มา : https://developers.google.com/calendar/api

Google Cloud [ออนไลน์]. สืบค้นเมื่อวันที่ 15 มิถุนายน

แหล่งที่มา : <https://console.cloud.google.com/>

W3schools. (2022). CSS. [ออนไลน์]. สืบค้นเมื่อวันที่ 5 พฤษภาคม 2566

แหล่งที่มา : <https://www.w3schools.com/css/>

Medium. (2022). CRUD Express.js MongoDB. [ออนไลน์]. สืบค้นเมื่อวันที่ 5 พฤษภาคม 2566

แหล่งที่มา : https://karnyong.medium.com/