



## โครงการ

Mini project

## จัดทำโดย

6504062630278 นายวิศรุต ศรีนวลปาน

## เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2565

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

## เกี่ยวกับโครงการ

ชื่อโปรเจค : numerical website

นำเสนอโดย : นายวิศรุต ศรีนวลปาน

อาจารย์ผู้สอน : ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

### บทที่ 1 ที่มาและความสำคัญ

โครงการนี้จัดทำขึ้นเพื่อใช้ความรู้จากวิชา Object Oriented Programming มาประยุกต์ใช้ในการสร้างเว็บไซต์ด้วย React Frame work ควบคู่ไปกับวิชา Numerical Method

### ประเภทของโครงการ

โปรแกรมเว็บแอปพลิเคชัน

### ประโยชน์

1. เมื่อความสะดวกสบายของผู้ใช้
2. เพื่อเรียนรู้การใช้ React

### ตารางแผนการทำงาน

ลำดับ	รายการ	13 ก.ย.-19 ก.ย.	20 ก.ย.-20 ต.ค.	21 ต.ค.-31 ต.ค.
1	ศึกษาการเขียน React			
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง			
3	ลงมือเขียนโปรแกรม			
4	จัดทำเอกสาร			
5	ตรวจสอบและแก้ไขข้อผิดพลาด			

## บทที่ 2 ส่วนการพัฒนา

### 2.1 รูปแบบการพัฒนา

- 2.1.1 สร้าง function ในการคำนวณ
- 2.1.2 สร้าง API สำหรับการนำค่าไปคำนวณ
- 2.1.3 สร้าง Route แต่ละหน้าของ เว็บ
- 2.1.4 สร้างหน้า Ui
- 2.1.5 ทำกราฟแสดงข้อมูลที่คำนวณ
- 2.1.6 สร้าง Database เพื่อเก็บข้อมูลโจทย์

### 2.2 อธิบายส่วนของโปรแกรมเกี่ยวกับหลักการ OOP

```
class Bisection extends Component<Nullable<unknown>, MyState> {
  GraphY: number[] = [];
  constructor(props: Nullable<unknown>) {
    super(props);

    this.state = {
      xl: 0,
      xr: 0,
      question: "",
      questionNumber: 1,
      resData: null,
      err: [],
      errY: [],
    };
  }
}
```

ตัวอย่างการใช้ class component ในการสร้างหน้า UI ของ component Bisection

component Bisection มีการ Inherit Component มา.

มีการเรียกใช้ **Constructor** ในการสร้าง Hook useState()

และภายในของ component Bisection ก็จะมี Method เช่น

```
getData = async (question: string, xl: number, xr: number): Promise<void> => {
  if (!question || isNaN(xl) || isNaN(xr)) {
    alert("Please enter valid values.");
    return;
  }

  try {
    const response = await axios.post<BisectionRes>(`http://localhost:8080/bisection`, {
      question,
      xl,
      xr,
    });
    const newErr = [];
    for (let i = 0; i < response.data.iterationData.length; i++) {
      newErr.push(response.data.iterationData[i].error);
    }
    console.log(newErr);
    const step = Math.abs(xl - xr) / response.data.iterationData.length;
    const newErrY = [];
    for (let i = 0; i < response.data.iterationData.length; i++) {
      newErrY.push(i);
    }
    console.log(newErrY);
    this.setState({ err: newErr });
    this.setState({ errY: newErrY });
    this.setState({ resData: response.data });
    console.log(response.data);
  } catch (error) {
    console.error("Error:", error);
    alert("An error occurred while fetching data.");
  }
};
```

getData ทำหน้าที่ในการตรวจสอบความผิดพลาดของ input และนำค่าที่ได้ส่งไปให้ API endpoint /bisection ที่ทำเอาไว้ด้วย express ด้วยการ POST และมี Input เป็น Body หลังจากนั้นจะนำค่าที่ได้จาก API มา set State.

```
function App() {
  const navigate = useNavigate();
  return (
    <>
      <Box
        sx={{
          backgroundColor: (theme) =>
            theme.palette.mode === "light"
              ? theme.palette.grey[100]
              : theme.palette.grey[900],
        }}
      >
        <Navbar navigate={navigate} />
        <Box padding={3} margin="auto" maxWidth={70} >
          <Paper sx={{ padding: 3, minHeight: "100vh" }} elevation={1}>
            <Routes>
              <Route path="/" element={<Index navigate={navigate} /> } />
              <Route path="/Test" element={<Test /> } />
              <Route
                path="/root"
                element={
                  <Outlet /> // Render nested routes
                }
              >
                <Route path="bisection" element={<Bisection /> } />
                <Route path="false-position" element={<FalsePosition /> } />
                <Route path="graphical" element={<Graphical /> } />
                <Route path="one-point" element={<OnePoint /> } />
                <Route path="newton-raphson" element={<NewtonRaphson /> } />
                <Route path="secant" element={<Secant /> } />
              </Route>
            </Routes>
          </Paper>
        </Box>
      </>
    </>
  );
}
```

ใน component App มีการเรียกใช้ Navbar มาเป็นส่วนประกอบของตอนเหมือนกันกับการ Composition

## บทที่ 3 ส่วนการพัฒนา

### ปัญหาที่พบระหว่างการพัฒนา

1. การเขียน React ด้วย class component เป็นสิ่งที่ React ไม่ได้สนับสนุนเพิ่มเติมแล้ว ทำให้การเขียนแบบ function component เป็นตัวเลือกที่ดีกว่า
2. การเขียน class component มี Document ที่ล้าสมัยทำให้เรียนรู้ได้ยาก
3. React ใช้หลักการ OOP ได้ไม่ครบเนื่องจาก Javascript ไม่เหมาะกับการทำ OOP เท่า Java

### จุดเด่นของโปรแกรม

1. มีการดึงข้อมูลจาก database ทำให้ผู้ใช้สะดวกสบายยิ่งขึ้น
2. มีการแบ่งเป็นส่วน Client และ ส่วน Server ทำให้สามารถแยกเครื่องการทำงานได้

### คำแนะนำ

ใช้ Frame work อื่นที่รองรับ การเขียนแบบ OOP มากกว่านี้