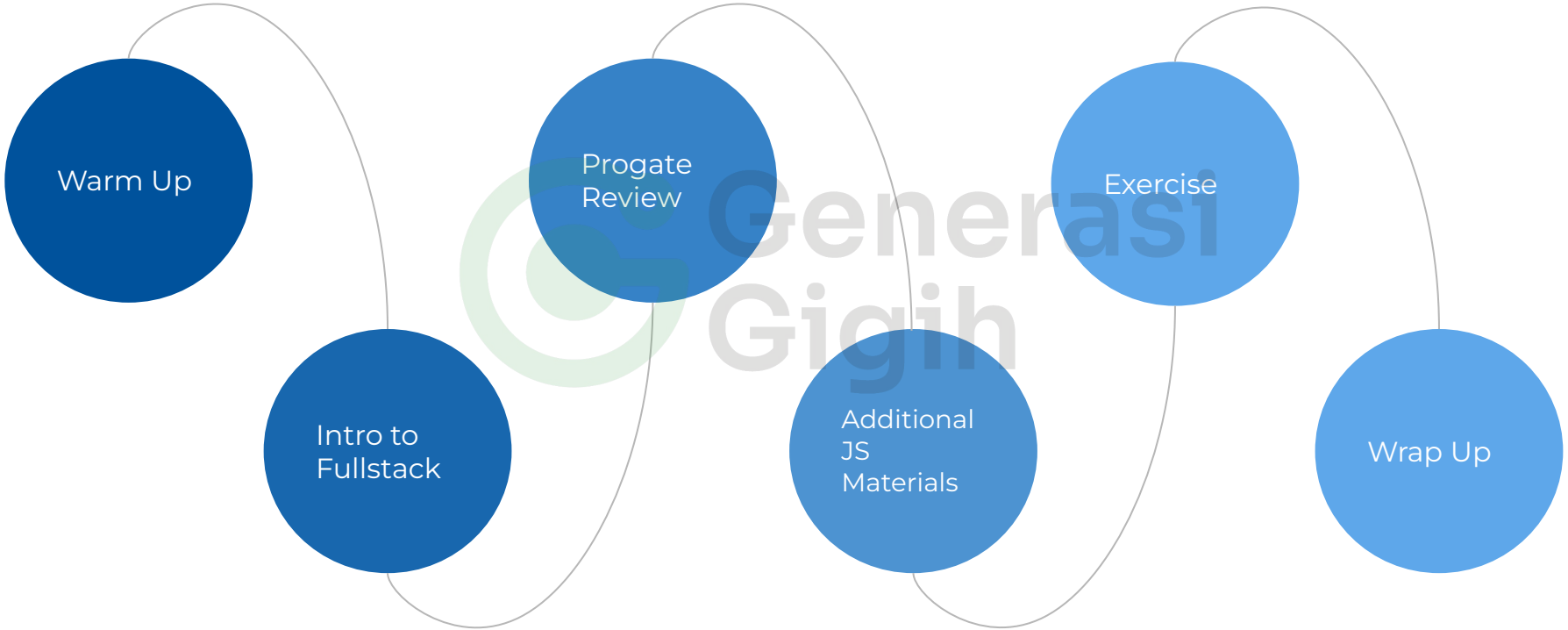
Full Stack Engineer

Module 1.1: Introduction to Fullstack Track

Our Agenda


Warm Up
Intro to Fullstack
Progate Review
Additional JS Materials
Exercise
Wrap Up

Generasi Gigih

Let’s Warm Up!

Let’s Discuss

What is fullstack

Javascript

Generasi Gigih

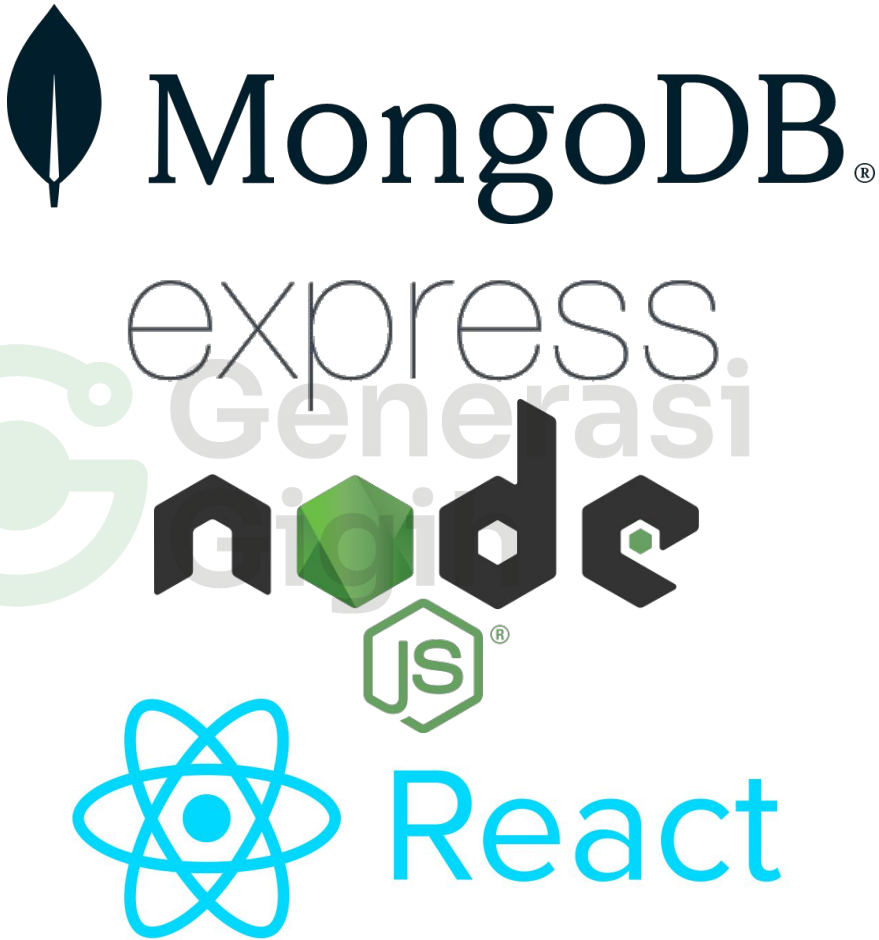
Let’s Talk About The Materials

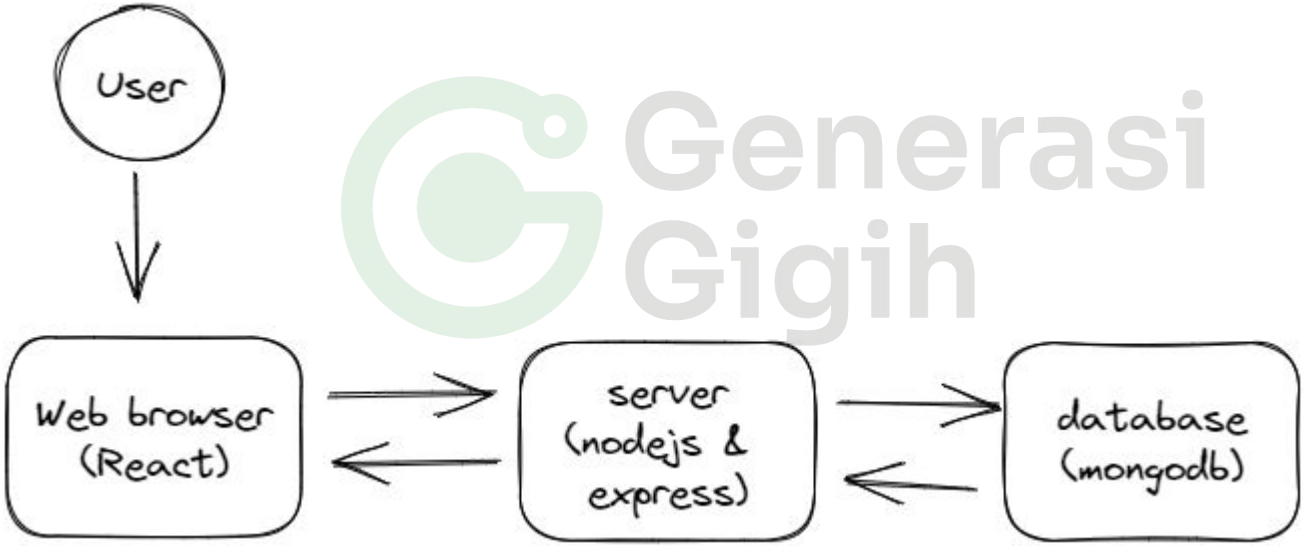
Fullstack Engineering

- involves working on both the frontend and backend of an app
- Frontend: refers to the user interface that users interact with
- Backend: refers to the server-side development that supports the frontend
- Fullstack engineers are responsible for managing the entire application development process from start to finish.
- In this project, we will use 1 language, Javascript, that supports both Backend and Frontend application

Generasi Gigih

MERN stack





Review: Javascript

- Before we dive further into the course, let’s take some quiz on Javascript
- There will be some basic questions related to Javascript.
- Try to answer it by yourself
- Good luck :)

Review: Javascript

```
let number = 1
number += 2

console.log(number)
```

Guess the output

a. 1
b. 2
c. 3
d. 4

Review: Javascript

```
let number = 1
number += 2

console.log(number)
```

Guess the output

a. 1
b. 2
c. 3
d. 4

Review: Javascript

```
const A = true
const B = false
```

What is the output of (A || B)

a. true
b. false

Review: Javascript

```
const A = true
const B = false
```

What is the output of (A || B)

a. true
b. false

Review: Javascript

```
for (let i = 0; i < 5; i++) {
  console.log(i)
}
```

Guess the output

a. 0 1 2 3 4
b. 0 1 2 3 4 5
c. 1 2 3 4
d. 1 2 3 4 5

Review: Javascript

```
for (let i = 0; i < 5; i++) {
  console.log(i)
}
```

Guess the output

a. 0 1 2 3 4
b. 0 1 2 3 4 5
c. 1 2 3 4
d. 1 2 3 4 5

Further reading:
https://en.wikipedia.org/wiki/Off-by-one_error

Review: Javascript

```
const fruits = [
  “apple”,
  “banana”,
  “orange”
]
```

What is the value of fruits[1]

a. apple
b. banana
c. orange

Review: Javascript

```
const fruits = [
  “apple”,
  “banana”,
  “orange”
]
```

What is the value of fruits[1]

a. apple
b. banana
c. orange

Remember that array in Javascript start from 0

Review: Javascript

```
const student = {
  name: “Alice”,
  age: 15
}
```

What is the value of student.name

a. Alice
b. 15
c. {name: “Alice”, age: 15}
d. undefined

Review: Javascript

```
const student = {
  name: “Alice”,
  age: 15
}
```

What is the value of student.name

a. Alice
b. 15
c. {name: “Alice”, age: 15}
d. undefined

Explanation

- (b) is when we call student.age
- (c) is when we call student
- (d) is when we call key that does not exist in student

Review: Javascript

```
func someFunc(num1, num2) {
    return num1 + num2
}
```

What is the output of (console.log(2,3))

a. 2
b. 3
c. 5

Review: Javascript

```
func someFunc(num1, num2) {
    return num1 + num2
}
```

What is the output of (console.log(2,3))

a. 2
b. 3
c. 5

The function given is a function to add 2 values

Review: Javascript

```
// animal.js

const animal = “dog”
export default animal
```

How can we import animal in our code

a. Import { animal } from animal.js
b. Import animal from animal.js
c. None of the above
d. both

Review: Javascript

```
// animal.js

const animal = “dog”
export default animal
```

How can we import animal in our code

a. Import { animal } from animal.js
b. Import animal from animal.js
c. None of the above
d. both

Since we’re doing export default, then we import it without { }. If we’re doing

export const animal = “dog”

Then we import it with (a)

Generasi Gigih

Additional JS materials

Optional Chaining

```
const students = [{ name: "Alice", age: 20 }];
students.forEach(student => {
  console.log("Age", student?.age);
});

console.log("Second Student Age",
students[1]?.age);
```

- ?. operator is to safeguard us from calling a function or accessing a property inside an undefined object.
- If we call it without ?. operator, then it will throw TypeError, because JS cannot execute `undefined.age`

Object Destructuring

```
function studentsAgeReducer2(acc, { age }) {
  return acc + age;
}

function averageAge2(students) {
  const totalAge = students.reduce(studentsAgeReducer2,
0);
  return totalAge / students.length;
}

const students = [
  { name: "Alice", age: 20 },
  { name: "Bob", age: 21 },
  { name: "Jane", age: 20 }
];
console.log(averageAge2(students));

const obj = { name: "Alice", age: 20 };
const { name, age } = obj;
console.log(name, age);
```

Shorthand Property Names

```
const students = [
 { name: "Alice", age: 20 },
 { name: "Bob", age: 21 },
 { name: "Jane", age: 20 }
]

const name = "John"
const age = 20
const newStudent = { name, age }

students.push(newStudent)
students.forEach((student) => {
 console.log(student.name)
})
```

Rest & Spread

```
const students = [
  { name: "Alice", age: 20 },
  { name: "Bob", age: 21 },
  { name: "Jane", age: 20 }
];

const name = "John";
const age = 20;
const newStudent = { name, age };

const newStudents = [...students, newStudent];
console.log(newStudents);
```

Generasi Gigih

Hands On

Exercise time!

Create a basic HTML page with 2 textboxes, email and password. After that. create a validation function that accepts email and password. The basic validation rules are:

1. email must be in email format
2. password cannot be less than 6 characters
3. Both field cannot be empty
4. ... add some more validation that you think necessary

(Optional) connect the 2 textboxes to the validation function

Generasi Gigih

Showcase Time!

Generasi Gigih

Q&A!

Generasi Gigih

Finally, Let’s Wrap Up!