.

Lesson 02 Demo 14

Implementing Stack and Queue Operations Using Deque

Objective: To demonstrate the implementation of both stack and queue operations using a deque (double-ended queue) in JavaScript, which showcases its versatility in supporting multiple linear data operations and broadens your understanding of versatile data structures

Tools required: Visual Studio Code

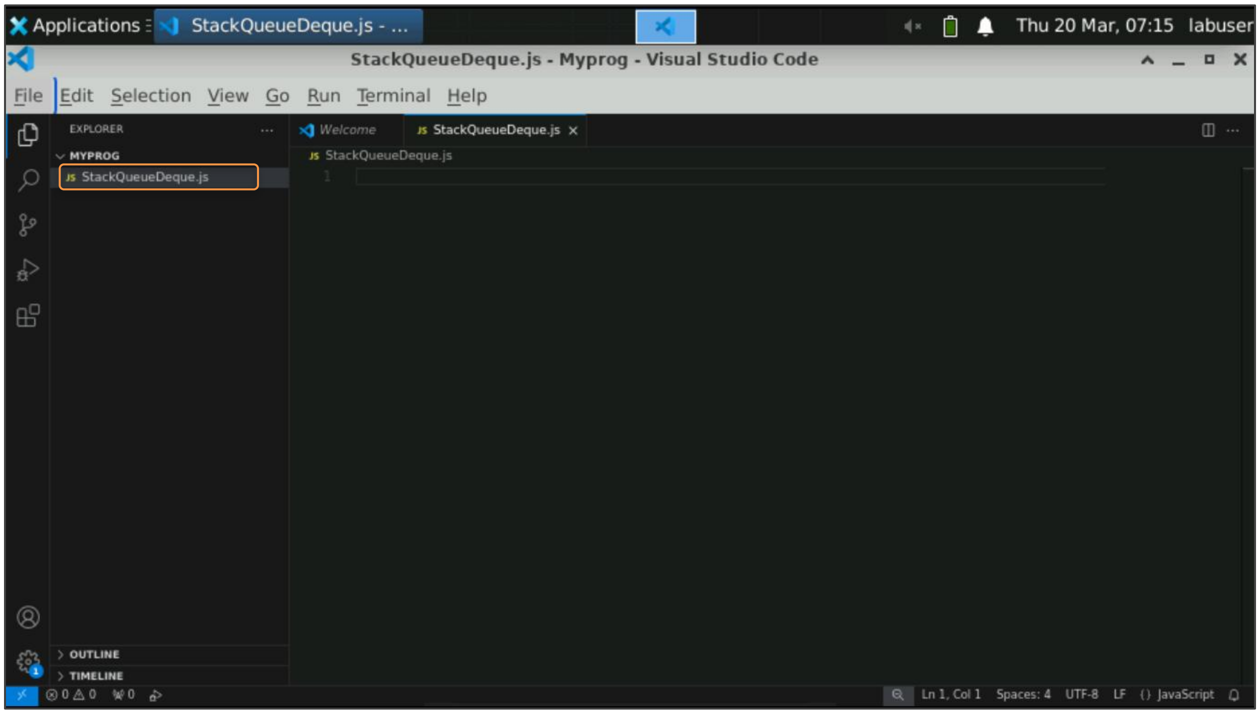
Prerequisites: A basic understanding of stacks, queues, deques, and JavaScript

Steps to be followed:

1. Create a JavaScript file and execute it

Step 1: Create a JavaScript file and execute it

1.1 Open the Visual Studio Code editor and create a JavaScript file named **StackQueueDeque.js**


Applications
StackQueueDeque.js - ...
Thu 20 Mar, 07:15 labuser
StackQueueDeque.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
StackQueueDeque.js
Welcome
StackQueueDeque.js
OUTLINE
TIMELINE
Ln 1, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

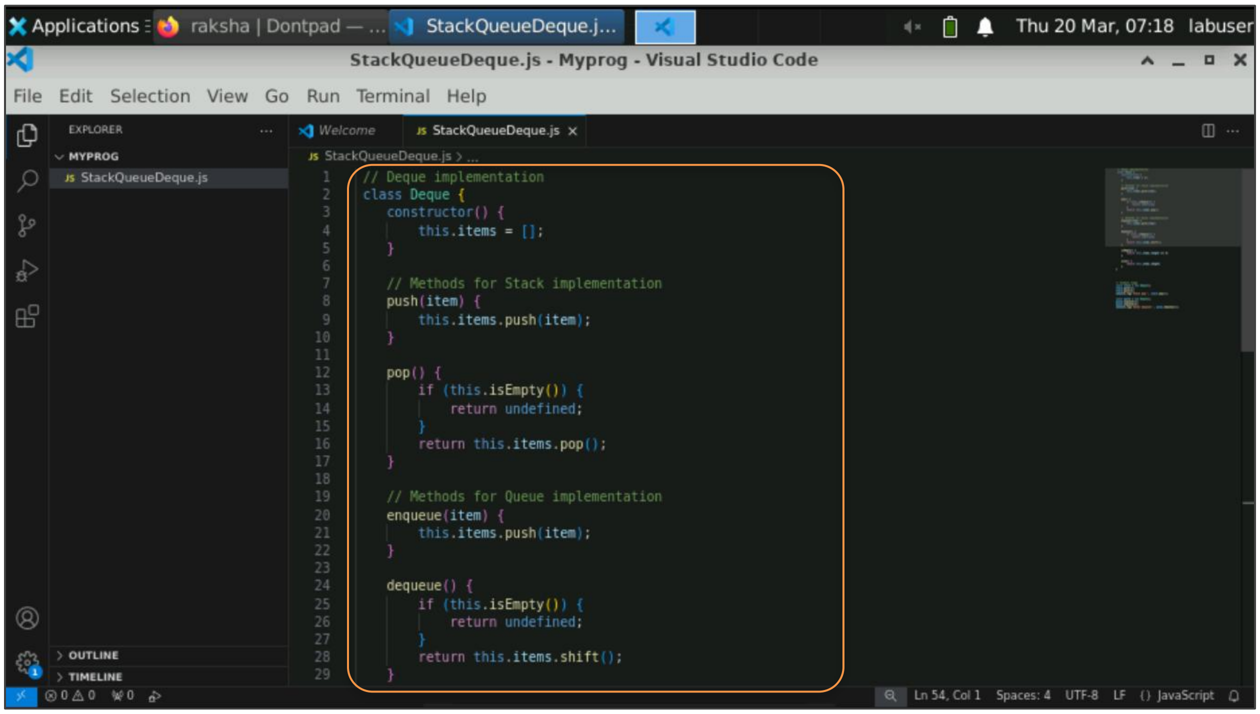
.

1.2 Add the following code to the file:

```
// Deque implementation
class Deque {
  constructor() {
    this.items = [];
  }

  // Methods for Stack implementation
  push(item) {
    this.items.push(item);
  }

  pop() {
    if (this.isEmpty()) {
      return undefined;
    }
    return this.items.pop();
  }

  // Methods for Queue implementation
  enqueue(item) {
    this.items.push(item);
  }

  dequeue() {
    if (this.isEmpty()) {
      return undefined;
    }
    return this.items.shift();
  }

  isEmpty() {
    return this.items.length === 0;
  }

  size() {
    return this.items.length;
  }
}
```

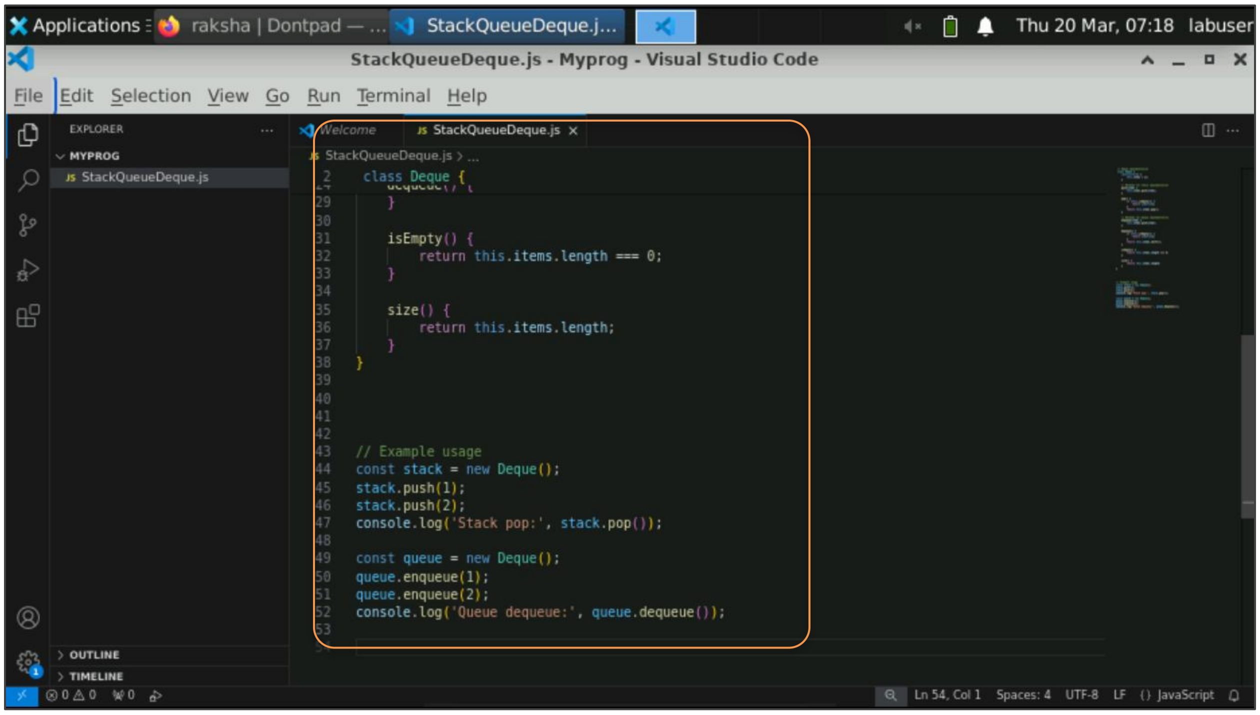
.

```
// Example usage
const stack = new Deque();
stack.push(1);
stack.push(2);
console.log('Stack pop:', stack.pop());

const queue = new Deque();
queue.enqueue(1);
queue.enqueue(2);
console.log('Queue dequeue:', queue.dequeue());
```


Applications
raksha | Dontpad — ...
StackQueueDeque.j...
Thu 20 Mar, 07:18 labuser
StackQueueDeque.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
StackQueueDeque.js
Welcome
StackQueueDeque.js
OUTLINE
TIMELINE
Ln 54, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

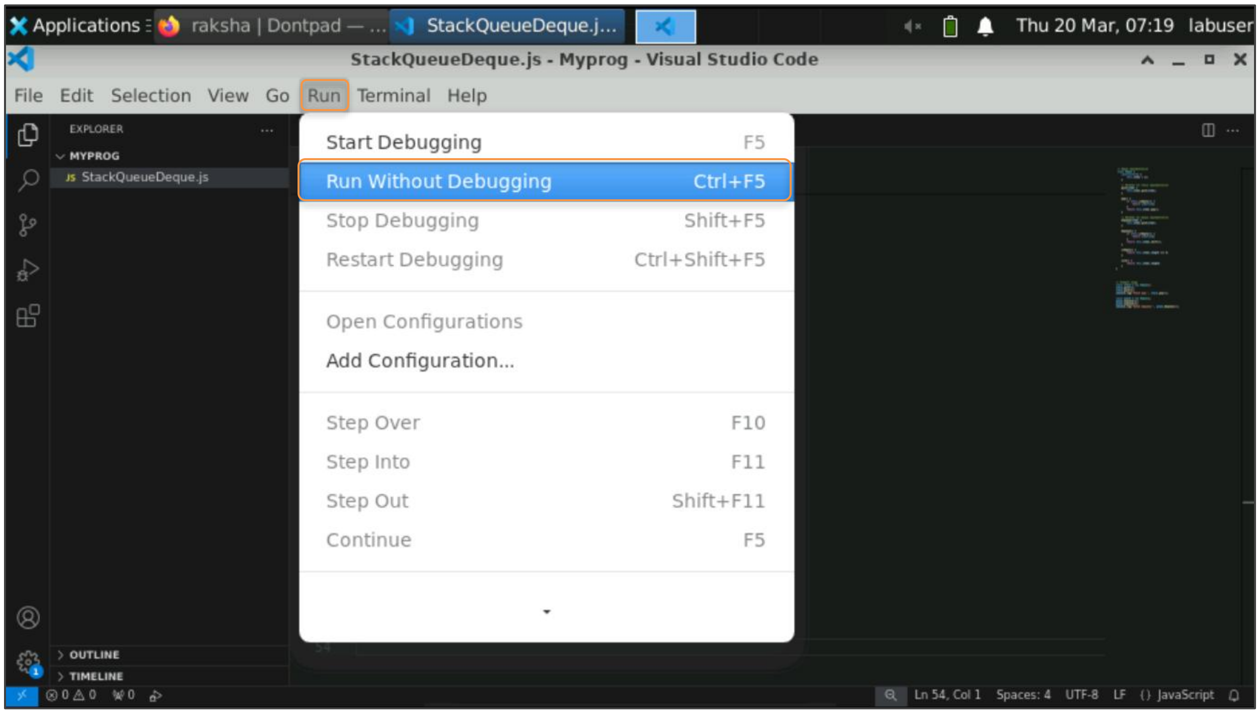
```
// Deque implementation
class Deque {
    constructor() {
        this.items = [];
    }

    // Methods for Stack implementation
    push(item) {
        this.items.push(item);
    }

    pop() {
        if (this.isEmpty()) {
            return undefined;
        }
        return this.items.pop();
    }

    // Methods for Queue implementation
    enqueue(item) {
        this.items.push(item);
    }

    dequeue() {
        if (this.isEmpty()) {
            return undefined;
        }
        return this.items.shift();
    }
```

.

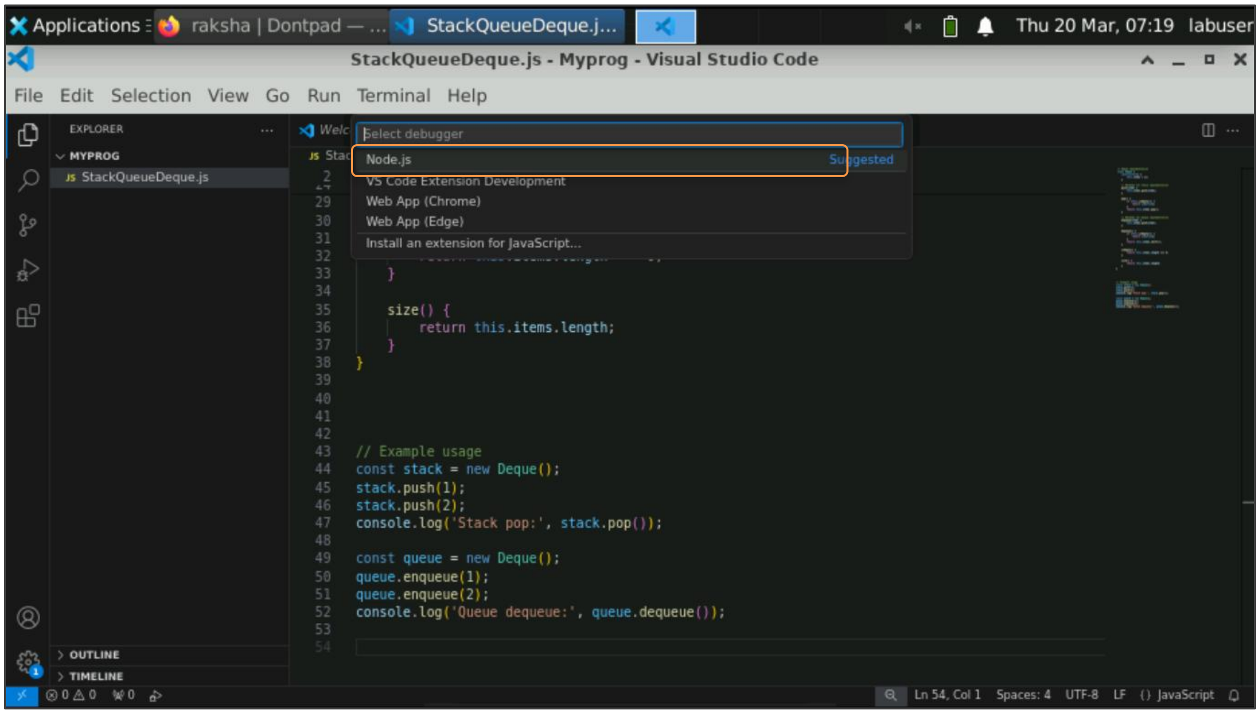

Applications
raksha | Dontpad — ...
StackQueueDeque.j...
Thu 20 Mar, 07:18 labuser
StackQueueDeque.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
StackQueueDeque.js
Welcome
StackQueueDeque.js
OUTLINE
TIMELINE
Ln 54, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

```
class Deque {
    }

    isEmpty() {
        return this.items.length === 0;
    }

    size() {
        return this.items.length;
    }
}



// Example usage
const stack = new Deque();
stack.push(1);
stack.push(2);
console.log('Stack pop:', stack.pop());

const queue = new Deque();
queue.enqueue(1);
queue.enqueue(2);
console.log('Queue dequeue:', queue.dequeue());
```

1.3 Click **Run** and then **Run Without** Debugging. Select **Node.js** to check the output in the DEBUG CONSOLE.

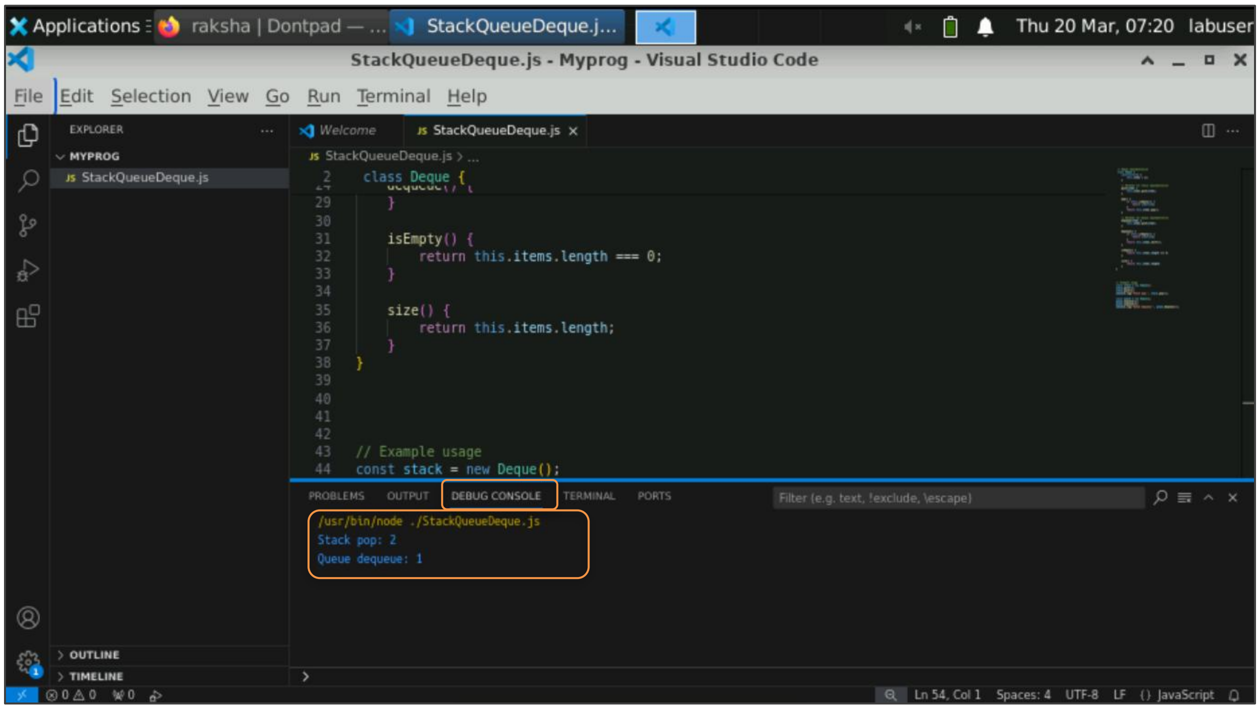


.


Select debugger
Node.js
Suggested
VS Code Extension Development
Web App (Chrome)
Web App (Edge)
Install an extension for JavaScript...

```
    }

    size() {
        return this.items.length;
    }
}



// Example usage
const stack = new Deque();
stack.push(1);
stack.push(2);
console.log('Stack pop:', stack.pop());

const queue = new Deque();
queue.enqueue(1);
queue.enqueue(2);
console.log('Queue dequeue:', queue.dequeue());
```

1.4 View the output in the **DEBUG CONSOLE** as shown below:



```
class Deque {
    }

    isEmpty() {
        return this.items.length === 0;
    }

    size() {
        return this.items.length;
    }
}



// Example usage
const stack = new Deque();
```

```
/usr/bin/node ./StackQueueDeque.js
Stack pop: 2
Queue dequeue: 1
```

By following these steps, you have successfully implemented stack and queue operations using a deque in JavaScript, broadening your understanding of versatile data structures and their applications in programming.