

WRITTEN BY MR. HUYNH NAM

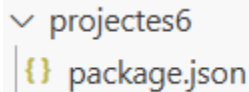
Youtube: <https://www.youtube.com/@nammedia>

Topic: ES6

Watch movies on @nammedia Youtube channel: lesson 01 and lesson 02

Build a simple CRUD console application to manage product list using Node.js and ES6 features like classes, modules, arrow functions, promises or async/await, template literals, and destructuring.

1. Create the project named **projectes6**



2. Prompt: ***In the project, create class Product has four properties: code, name, price and quantity in product.js file***

Then using the hint of code in AI to generate code in product.js file

```
class Product {
  constructor(code, name, price, quantity) {
    this.code = code;
    this.name = name;
    this.price = price;
    this.quantity = quantity;
  }

  getTotalPrice() {
    return this.price * this.quantity;
  }

  getDetails() {
    return `Code: ${this.code}, Name: ${this.name}, Price: ${this.price}, Quantity: ${this.quantity}`;
  }

  setOutOfStock() {
    this.quantity = 0;
  }
}

module.exports = Product;
```

3. Prompt: ***Create class ProductManager which implements CRUD operations for managing product list is in productmanager.js***

Check code in productmanager.js

WRITTEN BY MR. HUYNH NAM

Youtube: <https://www.youtube.com/@nammedia>

```
const Product = require('./product');

class ProductManager {
  constructor() {
    this.products = [];
  }

  // Create
  addProduct(product) {
    this.products.push(product);
  }

  // Read
  getAllProducts() {
    return this.products;
  }

  getProductByCode(code) {
    return this.products.find(p => p.code === code);
  }

  // Update
  updateProduct(code, updatedFields) {
    const product = this.getProductByCode(code);
    if (product) {
      Object.assign(product, updatedFields);
      return true;
    }
    return false;
  }

  // Delete
  deleteProduct(code) {
    const index = this.products.findIndex(p => p.code === code);
    if (index !== -1) {
      this.products.splice(index, 1);
      return true;
    }
    return false;
  }
}

module.exports = ProductManager;
```

4. Prompt: ***The main console program which shows the menu with five options in CRUD is in app.js file***

Simple app.js by AI here

WRITTEN BY MR. HUYNH NAM

Youtube: <https://www.youtube.com/@nammedia>

```
const readline = require('readline');
const Product = require('./product');
const ProductManager = require('./productmanager');

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

const manager = new ProductManager();

function showMenu() {
  console.log('\n--- Product Manager ---');
  console.log('1. Add Product');
  console.log('2. View All Products');
  console.log('3. Update Product');
  console.log('4. Delete Product');
  console.log('5. Exit');
}

function promptMenu() {
  showMenu();
  rl.question('Choose an option (1-5): ', (answer) => {
    switch (answer.trim()) {
      case '1':
        addProduct();
        break;
      case '2':
        viewProducts();
        break;
      case '3':
        updateProduct();
        break;
      case '4':
        deleteProduct();
        break;
      case '5':
        rl.close();
        break;
      default:
        console.log('Invalid option. Please try again.');
```

```
        promptMenu();
    }
  });
}
```

Repost copy is not allowed

WRITTEN BY MR. HUYNH NAM

Youtube: <https://www.youtube.com/@nammedia>

```
function addProduct() {
  rl.question('Enter product code: ', (code) => {
    rl.question('Enter product name: ', (name) => {
      rl.question('Enter product price: ', (price) => {
        rl.question('Enter product quantity: ', (quantity) => {
          const product = new Product(code, name, parseFloat(price), parseInt(quantity));
          manager.addProduct(product);
          console.log('Product added successfully.');
```



```
          promptMenu();
        });
      });
    });
  });
}

function viewProducts() {
  const products = manager.getAllProducts();
  if (products.length === 0) {
    console.log('No products found.');
```

```
  } else {
    console.log('\nProduct List:');
    products.forEach(p => {
      console.log(`Code: ${p.code}, Name: ${p.name}, Price: ${p.price}, Quantity: ${p.quantity}`);
    });
  }
  promptMenu();
}

function updateProduct() {
  rl.question('Enter product code to update: ', (code) => {
    const product = manager.getProductByCode(code);
    if (!product) {
      console.log('Product not found.');
```

```
      promptMenu();
      return;
    }
    rl.question('Enter new name (leave blank to keep current): ', (name) => {
      rl.question('Enter new price (leave blank to keep current): ', (price) => {
        rl.question('Enter new quantity (leave blank to keep current): ', (quantity) => {
          const updatedFields = {};
          if (name) updatedFields.name = name;
          if (price) updatedFields.price = parseFloat(price);
          if (quantity) updatedFields.quantity = parseInt(quantity);
          manager.updateProduct(code, updatedFields);
          console.log('Product updated successfully.');
```

```
          promptMenu();
        });
      });
    });
  });
}

function deleteProduct() {
  rl.question('Enter product code to delete: ', (code) => {
    if (manager.deleteProduct(code)) {
      console.log('Product deleted successfully.');
```

```
    } else {
      console.log('Product not found.');
```

```
    }
    promptMenu();
  });
}

promptMenu();
```

Repost copy is not allowed

WRITTEN BY MR. HUYNH NAM

Youtube: <https://www.youtube.com/@nammedia>

5. Upgrade level of code in app.js by prompt: **using promise, await and async**

```
const readline = require('readline');
const Product = require('./product');
const ProductManager = require('./productmanager');

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

const manager = new ProductManager();


function askQuestion(query) {
  return new Promise(resolve => rl.question(query, resolve));
}

function showMenu() {
  console.log('\n--- Product Manager ---');
  console.log('1. Add Product');
  console.log('2. View All Products');
  console.log('3. Update Product');
  console.log('4. Delete Product');
  console.log('5. Exit');
}
```

WRITTEN BY MR. HUYNH NAM

Youtube: <https://www.youtube.com/@nammedia>

```
async function promptMenu() {
  showMenu();
  const answer = await askQuestion('Choose an option (1-5): ');
  switch (answer.trim()) {
    case '1':
      await addProduct();
      break;
    case '2':
      await viewProducts();
      break;
    case '3':
      await updateProduct();
      break;
    case '4':
      await deleteProduct();
      break;
    case '5':
      rl.close();
      return;
    default:
      console.log('Invalid option. Please try again.');
```




```
  }
  await promptMenu();
}

async function addProduct() {
  const code = await askQuestion('Enter product code: ');
  const name = await askQuestion('Enter product name: ');
  const price = await askQuestion('Enter product price: ');
  const quantity = await askQuestion('Enter product quantity: ');
  const product = new Product(code, name, parseFloat(price), parseInt(quantity));
  manager.addProduct(product);
  console.log('Product added successfully.');
```

```

}

async function viewProducts() {
  const products = manager.getAllProducts();
  if (products.length === 0) {
    console.log('No products found.');
```



```
  } else {
    console.log('\nProduct List:');
    products.forEach(p => {
      console.log(`Code: ${p.code}, Name: ${p.name}, Price: ${p.price}, Quantity: ${p.quantity}`);
    });
  }
}
```

Repost copy is not allowed

WRITTEN BY MR. HUYNH NAM

Youtube: <https://www.youtube.com/@nammedia>

```
async function updateProduct() {
  const code = await askQuestion('Enter product code to update: ');
  const product = manager.getProductByCode(code);
  if (!product) {
    console.log('Product not found. ');
    return;
  }
  const name = await askQuestion('Enter new name (leave blank to keep current): ');
  const price = await askQuestion('Enter new price (leave blank to keep current): ');
  const quantity = await askQuestion('Enter new quantity (leave blank to keep current): ');
  const updatedFields = {};
  if (name) updatedFields.name = name;
  if (price) updatedFields.price = parseFloat(price);
  if (quantity) updatedFields.quantity = parseInt(quantity);
  manager.updateProduct(code, updatedFields);
  console.log('Product updated successfully. ');
}

async function deleteProduct() {
  const code = await askQuestion('Enter product code to delete: ');
  if (manager.deleteProduct(code)) {
    console.log('Product deleted successfully. ');
  } else {
    console.log('Product not found. ');
  }
}

promptMenu();
```

6. Run and test project
7. Prompt with AI tool to update project with more async functions as following
 - a. Option 9 – Exit
 - b. Option 5 – Show name, total price of product by code as keyword
 - c. Option 6 – Find and show all products by name as keyword
 - d. Option 7 – Set out of stock for product by code as keyword
 - e. Option 8 – List all product which is out of stock