



## MODULE 1: FOUNDATIONS (GETTING STARTED)

---

### LESSON 2: Setting Up Your Environment

#### Learning Objective

By the end of this lesson, you will have Node.js installed and your first Next.js application running on your computer.

---

#### 2.1 What You Need to Install

Before writing code, we need to set up your computer. Think of this like preparing your kitchen before cooking.

Required Software:

1. Node.js – This lets you run JavaScript on your computer (not just in browsers)
2. VS Code – A code editor (like Microsoft Word for code)

### 3. Git – Optional but helpful for saving your work

---

## 2.2 Installing Node.js

Node.js is the foundation. Here's how to install it:

Step 1: Go to [nodejs.org](https://nodejs.org)

Step 2: Download the LTS version (LTS means Long Term Support – it's the stable version)

Step 3: Run the installer

- Click "Next" on all screens (default settings are fine)
- Let it install

Step 4: Verify installation

Open your terminal (Command Prompt on Windows, Terminal on Mac) and type:

```
bash
```

```
node --version
```

You should see something like: `v20.11.0` (the version number may differ)

Also check npm (Node Package Manager) which comes with Node:

```
bash
```

```
npm --version
```

You should see: `10.2.4` or similar

✅ Success! Node.js is now installed.

---

## 2.3 Installing VS Code

VS Code is where you'll write your code.

Step 1: Go to [code.visualstudio.com](https://code.visualstudio.com)

Step 2: Download for your operating system (Windows, Mac, or Linux)

Step 3: Install with default settings

Step 4: Open VS Code

Step 5: Install helpful extensions (optional but recommended):

- Click the extensions icon (or press Ctrl+Shift+X)
- Search and install:
  - ES7+ React/Redux/React-Native snippets - gives you code shortcuts
  - Prettier - automatically formats your code nicely

---

## 2.4 Creating Your First Next.js App

Now for the exciting part! Let's create a real Next.js application.

Step 1: Open your terminal

Step 2: Navigate to where you want your project

```
bash
```

```
# On Windows - go to your Documents folder
```

```
cd Documents
```

*# On Mac - go to your Documents folder*

```
cd ~/Documents
```

### Step 3: Create a new Next.js app

```
bash
```

```
npx create-next-app@latest ceylonmart
```

You'll see some questions. Answer like this:

```
text
```

- ✓ Would you like to use TypeScript? ... No (we'll use JavaScript for this course)
- ✓ Would you like to use ESLint? ... Yes (helps catch errors)
- ✓ Would you like to use Tailwind CSS? ... No (we'll use regular CSS)
- ✓ Would you like to use `src/` directory? ... No
- ✓ Would you like to use App Router? ... Yes (this is the modern way)
  
- ✓ Would you like to customize the default import alias? ... No

This will take 1-2 minutes to install.

### Step 4: Open your project in VS Code

```
bash
```

```
cd ceylonmart
```

```
code .
```

### Step 5: Run your app

In the VS Code terminal, type:

```
bash
```

```
npm run dev
```

Step 6: See your app!

Open your browser and go to: <http://localhost:3000>

You should see the default Next.js starter page!

✅ Congratulations! Your first Next.js app is running.

---

## 2.5 Understanding the Folder Structure

Let's explore what was created. In VS Code, look at the left sidebar. Here's what each folder does:

text

ceylonmart/

```
|— app/           # Your main code lives here
| |— favicon.ico  # The small icon in browser tab
| |— globals.css  # Styles that apply everywhere
| |— layout.js    # The main layout (header, footer)
| |— page.js      # The homepage
| |— page.module.css # Styles just for homepage
|— public/        # Images, fonts, files
| |— next.svg
| |— vercel.svg
|— next.config.mjs # Next.js settings
|— package.json   # List of dependencies

└─ README.md     # Project description
```

Important Files Explained:

File	Purpose
------	---------

---

---

<code>app/page.js</code>	This is your homepage. Whatever you write here appears at <a href="#">yoursite.com</a>
<code>app/layout.js</code>	Wraps every page. Put header and footer here
<code>app/globals.css</code>	Styles that affect everything
<code>package.json</code>	Lists all packages your app uses

---

---

## 2.6 Clean Up: Remove Default Code

Let's clean up the default code to start fresh.

Step 1: Open `app/page.js`

Delete everything inside and replace with:

```
javascript
export default function Home() {
  return (
    <div>
      <h1>CeylonMart</h1>
      <p>Welcome to our Sri Lankan online store!</p>
    </div>
  );
}
```

Step 2: Save the file (Ctrl+S)

Look at your browser at <http://localhost:3000> - it should update automatically and show your new text!

Step 3: Open `app/globals.css`

Delete everything (we'll add our own styles later)

---

## Lesson 2 Summary

- Node.js lets you run JavaScript on your computer
  - VS Code is where you'll write your code
  - `create-next-app` sets up everything automatically
  - Your app runs at <http://localhost:3000>
  - `app/page.js` controls your homepage
- 

## Practice Exercise

1. Create another Next.js app called `my-test-app` (don't worry if you make mistakes, you can always create a new one)
2. Run it and make sure you see the default page
3. Change the text in `page.js` to say "Hello [Your Name]"
4. Challenge: Try changing the text color using inline styles (hint: `<h1 style={{ color: 'red' }}>`)