



## Stepping Into Programming

Author: Lewis Kiganjo

Published: May 01, 2026 23:16

Reading time: 3 min

# Understanding the Fundamentals in an AI-Driven Development World

The Internet Makes Development Look Easier Than It Is

When I first started programming, I thought software development was mostly about writing extremely difficult code and memorizing strange syntax.

Developers looked like people operating in another universe entirely, the kind you watch in movies typing aggressively on dark screens while systems magically respond.

Today, the pressure looks different...

The internet is full of videos saying you can build an app in one day using AI.

For beginners, especially in Kenya where many people enter tech through YouTube, TikTok, online tutorials, and AI tools, this creates unrealistic expectations very quickly.

You start feeling like your small projects are not enough.

Lakini the deeper I went into development, the more I realized something important:

Most stable systems are still built on fundamentals.

The tools change constantly, but the foundation remains almost the same.

- HTML structures the web
- CSS controls layout and presentation
- JavaScript handles interaction and logic

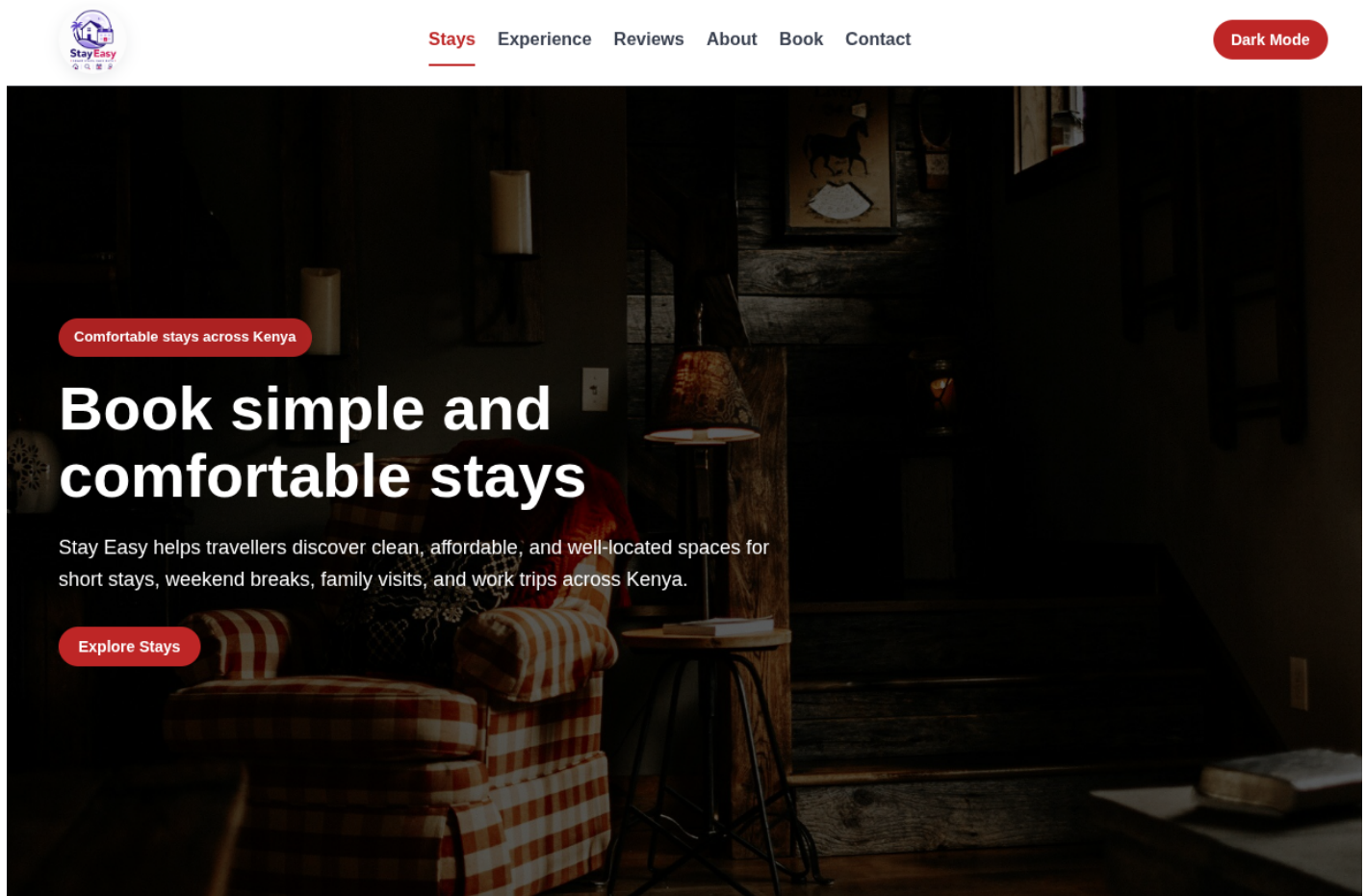
Once I understood this properly, websites stopped feeling magical and started feeling organized.



A navigation bar became structured links with spacing and alignment. Cards became layout containers managing hierarchy and responsiveness. Animations became timing and transitions working together. Development became less about memorizing code and more about understanding systems.

## Simple Projects Teach More Than People Think

One project that helped me understand this better was a simple Airbnb-style frontend called Stay Easy.



staeasyhero.png 914.09 KB

GitHub Repository:

[Stay Easy GitHub Repository](#)

Live Demo:

[Stay Easy Live Demo](#)

The project intentionally stayed simple using HTML, CSS, and JavaScript together without too much abstraction hiding what was happening underneath.

It included navigation sections, hero layouts, responsive cards, contact forms, hover interactions, and reusable UI patterns.

At the time, even those “simple” things felt difficult.

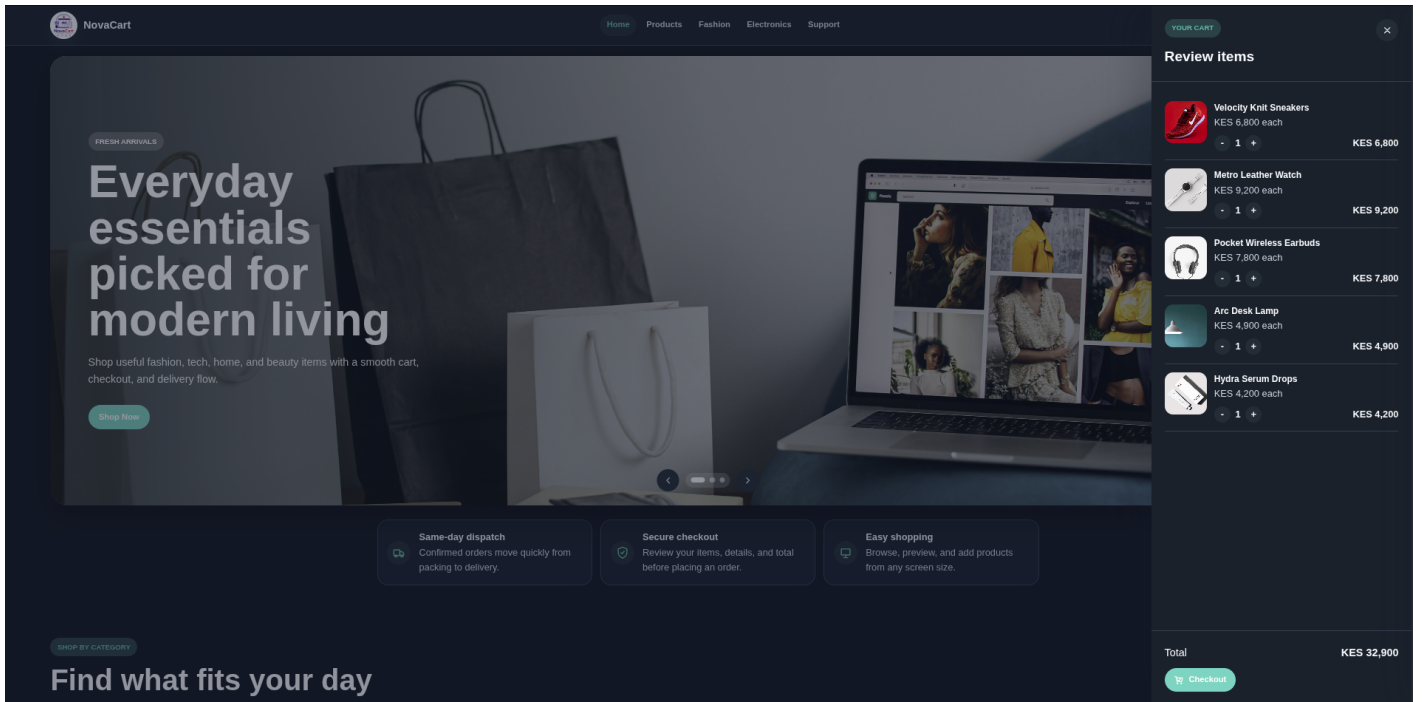
A small CSS issue could break an entire layout, mobile responsiveness introduced unexpected problems, and sometimes something worked perfectly on localhost but failed completely in production.

Pole pole, things started connecting.

You stop copying code blindly and begin understanding why something works.

Another project that pushed this understanding further was NovaCart, an ecommerce frontend project for heavily on interaction and user experience.





novacart.png 861.66 KB

GitHub Repository:

[NovaCart GitHub Repository](#)

Live Demo:

[NovaCart Live Demo](#)

NovaCart dives deeper into responsive ecommerce layouts, overlays, hover interactions, reusable sections, animations, and frontend consistency across pages.

Projects like these helped me realize that advanced systems are usually smaller ideas repeated consistently at scale.

AI tools are genuinely useful, and tools like GPT and Codex can help explain concepts, debug issues, and accelerate learning. But eventually, generated code fails, deployments break, layouts collapse on some devices, and prompts alone stop being enough.

That is where understanding matters.

## Learn the Foundations Through Real Projects





frontend\_web\_dev

One thing I noticed while learning frontend development is that many beginners jump between random tutorials without ever building complete projects properly.

That is why I created a frontend development program focused on learning through implementation instead of memorization.

The program uses the two projects above, StayEasy and NovaCart, to help beginners understand frontend development progressively through real systems instead of disconnected exercises.

You start simple, then gradually improve layouts, responsiveness, interactions, frontend organization, and application behavior over time.

The learning process mirrors real development:

Build something, break something, debug it, improve it, reorganize it, then build again better.

The program is free.

Simply register as a student, enroll, and start learning through the projects.

Most importantly, the goal is not just teaching people how to “make websites look good.”

It is helping people understand how frontend systems are structured, improved, maintained, and debugged in real-world environments.

