Project Summary:

CyberBuddy is a beginner-friendly website designed to **educate users about core cybersecurity concepts** through interactive conversations and quizzes — all without requiring internet access or storing personal data. This is built on HTML, CSS, and Javascript.

Problem Statement:
This project will solve the problem of having a reliable cybersecurity information website that is trustable with real sources and from a real cybersecurity worker compared to an unreliable website with no real explanation and with no cybersecurity worker explaining from experience.

Use Case:

The use case for my website is for adults and students 13+ that are interested in learning in cybersecurity, and/or want to pursue the cybersecurity field.

Goals and Objectives:
My goal is to make a website that talks about and explains what the field of cybersecurity is with real sources with an AI chatbot and easy and nice UI.

My goal compared to a normal cybersecurity website is to make an actual cybersecurity worker explain what cybersecurity is.

Key Features and Functions:

1. AI chatbot for any questions or assistance with the UI
2. Specific pages to learn more about cybersecurity
3. Hacking Simulation, but you need to make an account

Tech Stack and Tools:

Tech Stack for a Cybersecurity Website (Simple & Understandable)

A **tech stack** just means:

👉 *“What tools and technologies will you use to build your website?”*

Your cybersecurity website needs **3 major parts**:

1. **Front-End (what users see)**
2. **Back-End (the logic, login system, chatbot connections)**
3. **Extra Tools (security, storage, chatbot, deployment)**

Let's break each one down.

⭐ 1.

Front-End (the part the user sees)

This is what makes your website look good.

Technologies:

- **HTML** → structure (headings, text, buttons)
- **CSS** → design (colors, layout, animations)
- **JavaScript** → interactivity (buttons, quizzes, animations)

Why you need this:

This is the "visual" part of your cybersecurity website — pages like:

- Home page
- Cyber basics
- Social engineering
- Password generator
- Cyber quizzes

- Portfolio page

Tools that help you build it:

- **VS Code** → main coding app
- **Git + GitHub** → save your code, show your work
- **Bootstrap or Tailwind (optional)** → makes your website look modern without hard coding everything

---

⭐ 2.

Back-End (the "brain" of the website)

This controls login, user accounts, AI chatbot connections, saving quiz scores, etc.

You have 2 beginner-friendly options:

✨ Option A: Firebase (Easiest)

Best if you want:

- Login / signup
- Database
- Hosting
- Analytics
- Very low coding for back-end

What Firebase gives you:

- Authentication (login system)
- Firestore database (store quiz scores, user progress)
- Hosting (upload your site)
- Free tier

This is the easiest option for a beginner **but still looks professional**.

✨ Option B: Node.js + Express (More advanced, more control)

Best if you want:

- A real back-end server
- Full control
- More advanced resume project

You will need:

- **Node.js & NPM**
- **Express.js** → framework to build the backend
- **MongoDB or PostgreSQL** → database

You can later deploy it on:

- **Render (free)**
- **Railway**
- **Vercel**
- **Digital Ocean**

This is perfect if you want a “real developer” project.

⭐ 3.

Database (optional depending on your site)

If your website has login, saved progress, or user data:

▶

Easiest:

Firebase Firestore

▶

More advanced:

MongoDB (NoSQL) or

PostgreSQL (SQL)

⭐ 4.

AI Chatbot (Optional but highly recommended)

You want your CyberLearner site to have an AI assistant.

You have 3 options:

Option A — ChatGPT API (Best and easiest)

Use:

- **OpenAI API (Assistants API or Chat Completions API)**
- Connected to your **Node.js backend**

You create:

- A chatbot page
- A form where users type messages
- The backend sends the message to the AI
- The AI responds

Why it's good:

✔ Professional

✔ Modern

✔ Cheap

✔ Easy to add

✔ Adds major "wow" factor to your portfolio

Option B — HuggingFace APIs (free-ish but slower)

You can use models like:

- Llama 3
- Mistral

- Falcon

You still need a backend.

Option C — No-code chatbot (easiest beginner option)

Tools like:

- **Botpress**
- **Dialogflow**
- **Tidio AI**

You can embed it like:

Algorithm:

1. Go to the website
2. The website will display the home page
3. Read the home page
4. Go to the other pages like what is cybersecurity?
5. After you view the pages, go to the quiz
6. When you get a passing score, you pass the quiz
7. If not, you have to read the website again, and you need to get a passing score

---

⭐ 5.

Security tools to include (because it’s a CYBER site)

These are part of your *content* but can also be interactive features:

- **Password strength checker**
- **Password generator**
- **Phishing quiz**
- **Cyber tips pop-ups**
- **Two-factor explanation page**
- **Encryption demo (AES or Caesar cipher)**
- **Login Bruteforce simulator (optional)**

⭐ 6.

Deployment (putting your site online)

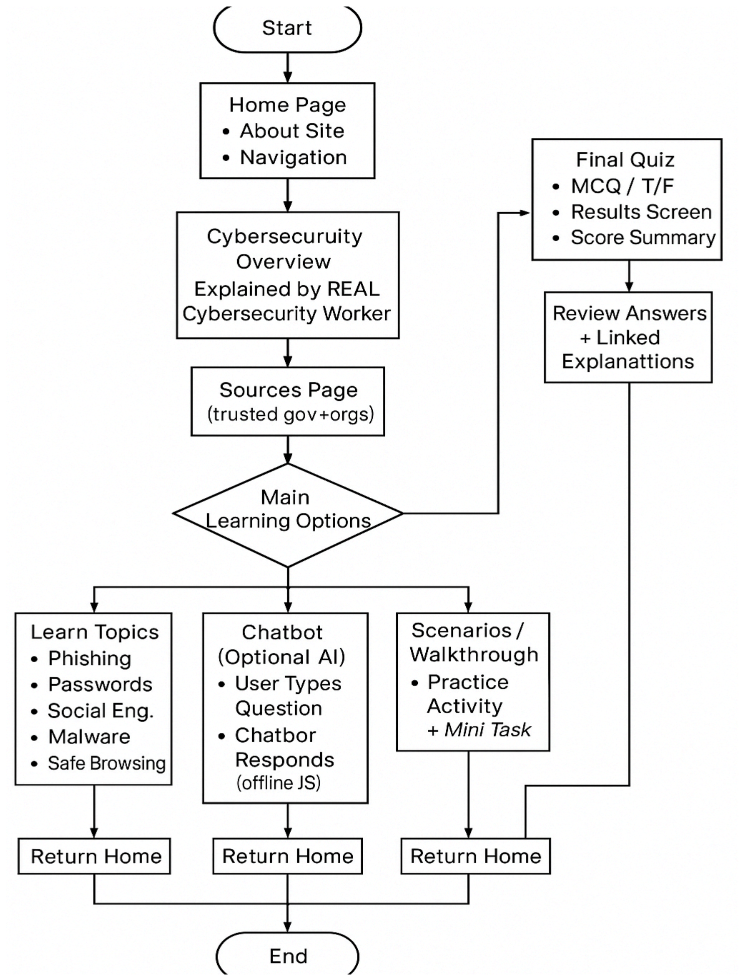
Simple (front-end only):

- **GitHub Pages**
- **Netlify**
- **Vercel**

Full-stack (backend + database + chatbot):

- **Render (free tier)**
- **Railway**
- **Vercel (with serverless functions)**

Flowchart:



Timeline:

Week of Nov 10

Mon Nov 10 — Define goals; identify features

Wed Nov 12 — UI sketches + sitemap

Week of Nov 17

Mon Nov 17 — Create project structure

Wed Nov 19 — Build navbar + footer

Week of Nov 24

Mon Nov 24 — Write “What is Cybersecurity?”

Wed Nov 26 — Basic CSS theme

Week of Dec 1

Mon Dec 1 — Create “Phishing” page

Wed Dec 3 — Create “Social Engineering” page

Week of Dec 8

Mon Dec 8 — Create page on strong passwords

Wed Dec 10 — Malware basics page

Week of Dec 15

Mon Dec 15 — Build chatbot UI placeholder

Wed Dec 17 — Expand FAQs

Week of Dec 22

Mon Dec 22 — Ensure offline styling only

Wed Dec 24 — Accessibility basics

Week of Dec 29

Mon Dec 29 — Quiz structure layout

Wed Dec 31 — Write quiz questions

Week of Jan 5

Mon Jan 5 — Quiz scoring logic

Wed Jan 7 — Results page

Week of Jan 12

Mon Jan 12 — Progress tracker

Wed Jan 14 — Flashcards

Week of Jan 19

Mon Jan 19 — Local-storage support (offline)

Wed Jan 21 — Improve visual theme

Week of Jan 26

Mon Jan 26 — Glossary page

Wed Jan 28 — PDF cheat sheet

Week of Feb 2

Mon Feb 2 — Safe browsing

Wed Feb 4 — Email safety

Week of Feb 9

Mon Feb 9 — Run offline testing

Wed Feb 11 — Improve chatbot replies

Week of Feb 16

Mon Feb 16 — Phishing examples checklist

Wed Feb 18 — Password practice exercise

Week of Feb 23

Mon Feb 23 — Animations

Wed Feb 25 — Quiz difficulty tune

Week of Mar 2

Mon Mar 2 — Finalize text

Wed Mar 4 — Improve mobile UI

Week of Mar 9

Mon Mar 9 — Write README

Wed Mar 11 — Build “How it works” page

Week of Mar 16

Mon Mar 16 — Accessibility & offline checks

Wed Mar 18 — Fix bugs

Week of Mar 23

Mon Mar 23 — Design polish

Wed Mar 25 — Backups

Week of Mar 30

Mon Mar 30 — Presentation practice

Wed Apr 1 — Final rehearsal

Week of Apr 6

Mon Apr 6 — Add IoT safety, VPN basics

Wed Apr 8 — Add infographics

Week of Apr 13

Mon Apr 13 — “Try it yourself” exercises

Wed Apr 15 — Expand quiz bank

Week of Apr 20

Mon Apr 20 — UI consistency

Wed Apr 22 — Export offline bundle

Week of Apr 27

Mon Apr 27 — Presentation outline

Wed Apr 29 — Website final review

Week of May 4

Mon May 4 — Add beginner-friendly troubleshooting tips

Wed May 6 — Add simple CSS themes (light/dark)

Week of May 11

Mon May 11 — Classroom user testing

Wed May 13 — Analyze feedback, fix minor issues

Week of May 18

Mon May 18 — Write Cyber Safety Scenarios

Wed May 20 — Build scenario-based questions

Week of May 25

Mon May 25 — Add brief parental guidance content

Wed May 27 — Visual polish on icons/images

Week of June 1

Mon Jun 1 — Add printable workbook formats

Wed Jun 3 — Add optional advanced topics

Week of June 8

Mon Jun 8 — Translate key terms (optional)

Wed Jun 10 — Retest offline reliability

Week of June 15

Mon Jun 15 — Review documentation

Wed Jun 17 — SEO improvements (local metadata only)

Week of June 22

Mon Jun 22 — Final polish + prep for release

Wed Jun 24 — FINAL ✅ Release build

Risk Mitigation:

Bad UI-

Solution- when the website is done and published, get feedback and ask people if the UI is good

Evaluation Criteria:

1. “This UI is very easy to use.”
2. “I love that you include an AI chatbot and a quiz at the end.”
3. “This info is trustable and right to the point

Future Considerations:

1. Include a cybersecurity quiz.

At the end when the user is done reading the website, ask them if they want to take an optional quiz.