**14-Day Plan (Milestone #1)**

**Repo + Skeleton**

**Do**

* Create a new GitHub repo: http-highways-site.
* Add files:  
  index.html, early-http.html, http2.html, http3.html, security-https.html, apis-cloud.html, key-concepts.html, about.html, styles.css, /assets/img/
* Paste a consistent <header><nav>…</nav></header> and <footer> on every page.
* Footer note: “This is a class project for IT 3203. Course site: https://ksuweb.github.io/IT3203/”
* Add starter HTML comments in each file: purpose, sections, TODOs.

**Proof of done**

* Local nav links work between all 8 pages.
* First commit: “feat: project skeleton, shared nav/footer, page stubs”.

**Content Map from Your Essay**

**Do**

* Create a one-pager outline for each page (bullets only):
  + Home: title, abstract (2–3 sentences), key terms, table of contents.
  + Early HTTP: 0.9 → 1.0 → 1.1 bullets, limits.
  + HTTP/2: multiplexing, HPACK, binary framing, server push, benefits/limits.
  + HTTP/3: QUIC, no transport HOL, TLS 1.3 default, mobile/wifi benefits.
  + Security/HTTPS: what TLS gives (auth, integrity, confidentiality); user vs engineer.
  + APIs/Cloud: methods, headers, status codes, content types, tokens.
  + Key Concepts: choose 7; write 1–2 bullets each.
  + About: mini-bio + project purpose. References list (URLs to add later).

**Proof of done**

* Outline saved in repo as /planning/content-map.md.
* Commit: “docs: content map for all pages”.

**Write Home + Key Concepts**

**Do**

* Write the **Home** page copy: title, abstract, key terms, TOC with links.
* Write the **Key Concepts** page (5–7 terms; 2–3 sentences each).
* Add HTML comments to explain structure.

**Proof of done**

* index.html and key-concepts.html render cleanly; internal links jump correctly.
* Commit: “content: home + key concepts”.

**Early HTTP Page + Comparison Table**

**Do**

* Draft **early-http.html** copy for 0.9, 1.0, 1.1 (short paragraphs).
* Add a **table** with columns: Version | “Vehicle” | Key features | Bottleneck | Year.
* Add an ordered list of “what changed” across 0.9→1.1.

**Proof of done**

* Page includes at least one **table** and one **list**.
* Commit: “content: early http (0.9/1.0/1.1) + comparison table”.

**HTTP/2 Page**

**Do**

* Write **http2.html**: explain multiplexing, header compression, binary framing, server push.
* Add a short “Why developers cared” bullet list (perf, fewer TCP conns).

**Proof of done**

* Page has headings (h1–h3) and a list; placeholders for an image figure.
* Commit: “content: http/2 features + benefits/limits”.

**HTTP/3 Page**

**Do**

* Write **http3.html**: QUIC over UDP, built-in TLS 1.3, no transport HOL blocking, mobile stability.
* Add a small “when it shines” list (video, streaming, gaming, spotty Wi-Fi).

**Proof of done**

* Page complete with headings/lists and image placeholder.
* Commit: “content: http/3 (quic) draft”.

**Security + APIs/Cloud + About/Refs**

**Do**

* **security-https.html**: quick TLS 1.3 benefits, plain vs HTTPS.
* **apis-cloud.html**: methods (GET/POST/PUT/PATCH/DELETE), headers (auth/content-type), status codes, statelessness at scale.
* **about.html**: mini bio + project purpose.
* **references**: gather links; put temporary list at bottom of About or a dedicated section.

**Proof of done**

* All content pages now have draft text.
* Commit: “content: security, apis-cloud, about, references (draft)”.

**CSS Base + Typography**

**Do**

* In styles.css add:
  + Base typography: body { font-family; line-height; }
  + Color tokens (3–4 max): background, headings, accents.
  + Link styles + **hover** effect (underline/opacity).
  + Spacing scale: :root { --space-1… } then use consistent margin/padding.
* Ensure **three selector types** appear: element (nav a), class (.toc-list li), id or pseudo (#site-title or a:hover).

**Proof of done**

* Styles apply site-wide from a single CSS file.
* Commit: “style: base typography, colors, link hover, spacing; selectors”.

**Layout with Flex or Grid**

**Do**

* Pick one for Milestone #1:
  + **Grid** on topic pages → 2 columns (content + sidebar/analogy box).
  + **Flex** for header/nav.
* Add a bordered callout box on each topic page (meets “borders”).
* Add consistent section spacing.

**Proof of done**

* Home + at least 2 topic pages demonstrate **grid or flex** layout clearly.
* Commit: “layout: grid/flex + bordered callouts”.

**Metaphor Images (SVG/PNG)**

**Do**

* Create simple visuals (Figma/PowerPoint/Keynote):
  + **Home**: evolution strip (pigeon → carriage → truck → multi-lane → QUIC network).
  + **Early HTTP**: 3 panels (0.9/1.0/1.1).
  + **HTTP/2**: one truck, **parallel lanes** inside (multiplexing).
  + **HTTP/3**: per-package independent paths + lock icon “TLS by default”.
  + **Security**: same vehicles with a **padlock overlay**.
  + **APIs/Cloud**: barcode/QR on boxes (headers/tokens).
* Export to /assets/img/, insert using <figure><img … alt="…"><figcaption>…</figcaption></figure>.

**Proof of done**

* Each page has **at least one image** + meaningful **alt** + caption.
* Commit: “feat: metaphor images + figures with alt/captions”.

**Accessibility + Semantics Pass**

**Do**

* Verify heading order (h1→h2→h3), unique page titles.
* Add aria-current="page" on the active nav link.
* Ensure link text is meaningful, not “click here”.
* Check color contrast (aim ≥ 4.5:1 for body text).
* Add/clarify HTML comments explaining structure and choices.

**Proof of done**

* Axe/Contrast check passes; no obvious a11y issues.
* Commit: “chore: a11y & semantic cleanup, comments”.

**Deploy to GitHub Pages + Validator**

**Do**

* GitHub → Settings → Pages → Deploy from main (root).
* Test live URL on desktop + phone.
* Run W3C HTML Validator on **each page**; fix errors.
* Fix any 404/broken nav links.

**Proof of done**

* Live site works; validator reports 0 errors (warnings ok if intentional).
* Commit: “fix: validator + link fixes; pages live”.

**Polish + References Final**

**Do**

* Tighten copy (remove redundancy, typo pass).
* Style the table (striped rows, readable padding).
* Verify every reference is a **clickable link**; add short annotations if helpful.
* Add final code comments: why grid/flex was chosen, style decisions.

**Proof of done**

* Site feels clean, professional, consistent; references page complete.
* Commit: “polish: copy, table styling, references, comments”.

**Final QA + Submission**

**Do**

* Milestone #1 **pre-flight checklist**:
  + ≥ 6 pages with consistent main menu
  + Headings, lists, **images**, **table** present
  + **One** CSS file; 3 selector types; hover; padding/margins; border; colors; fonts
  + **Grid or Flex** used for layout
  + Footer note + course link
  + **Live** on GitHub Pages; all links work
  + HTML validated; code commented
  + Zip all source files (no node\_modules, no binaries)
* Submit:
  + Paste **live URL** (clickable) into D2L comments.
  + Upload the **.zip** of source files.

**Proof of done**

* D2L shows link + zip uploaded; repo tagged v1.0-m1.
* Final commit: “release: milestone 1 submission”.

**If you get behind (buffer plan)**

* Combine Days 5–6 (HTTP/2 + HTTP/3) and Days 7–8 (Security/APIs + CSS base).
* Minimum to pass: all pages + table + one figure + grid/flex + CSS features + validator + live link.

**Tiny spiritual nudge (for morale)**

when it gets tedious, zoom out: “whatever you do, work at it with all your heart…” (Col 3:23). the craft matters. ship it clean.