

Capstone Command Center

Step 1: Create the Capstone Project Folder (now)

On your computer (and later in GitHub), make:

capstone-daily-dashboard
(or any name we finalize later)

Inside it, create these folders/files even before code:

capstone-daily-dashboard/

```
├── planning/
│   ├── capstone-plan.md
│   ├── feature-list.md
│   ├── timeline.md
│   ├── advisor-notes.md
│   └── ai-usage.md
├── design/
│   ├── wireframes/
│   ├── color-palette.png
│   └── inspiration.md
└── src/
    ├── index.html
    ├── stats.html
    └── css/
        └── js/
            ├── README.md
            └── .gitignore
```

This alone already makes you look like a real developer.

Step 2: Your Game Plan System (we'll build this together)

We'll maintain three living documents:

1) MASTER TASK BOARD (timeline.md)

Broken into:

PHASE 1 — PLANNING

- Finalize concept
- Lock MVP vs Stretch
- Pick API
- Draw wireframe
- Meet advisor

PHASE 2 — FOUNDATION

- HTML skeleton
- Routing / navigation
- Layout grid

PHASE 3 — LOGIC

- Task CRUD
- LocalStorage
- Validation
- API fetch
- Stats calculation

PHASE 4 — UI

- Styling
- Responsive
- Animations

PHASE 5 – POLISH

- README
- AI documentation
- Commit cleanup
- Presentation prep

2) FEATURE CONTRACT (feature-list.md)

Each feature gets:

- Purpose
- Requirement it satisfies
- Status (planned / building / working / polished)
- Files involved
- Whether AI-assisted

3) DAILY WORK LOG (advisor-notes.md)

So when Dan asks “what have you worked on?” you have:

- Dates
 - What you built
 - Blockers
 - Questions
-

Step 3: Git From Day One (important for commits requirement)

Once the folder exists, we will:

```
git init  
git commit -m "Initial project structure  
and planning docs"
```

Then every meaningful change = real commit. No fake "bulk upload at the end" nonsense.

Now, before we proceed, do this small concrete action:

Create the folder on your machine and tell me when it exists.

Then we'll start filling in **capstone-plan.md** together, beginning with:

- Project Name (working)
- One-sentence elevator pitch
- User
- Problem
- Solution
- MVP
- Stretch
- API
- Routes
- Risks (scope creep, time, complexity)

This is how we turn "I'm overwhelmed" into "I have a system."