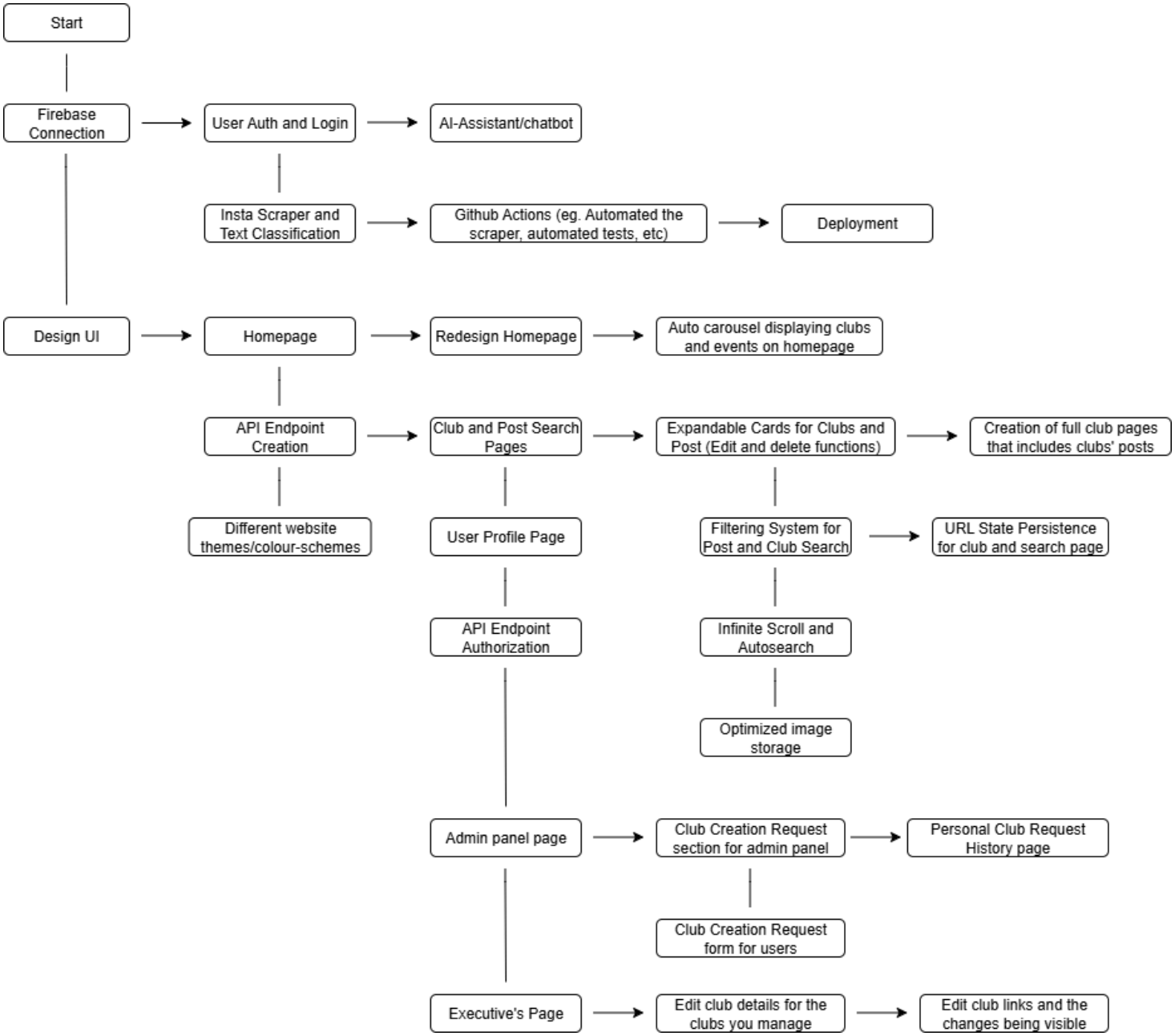
Network Diagram:

This network diagram represents all the tasks done so far and their dependencies and the tasks that were required to be finished beforehand. This diagram does NOT represent the ORDER of which we did the tasks, just the dependencies (Although sometimes the order is correct).

- " | " means that the task right above and right below this are **parallel tasks**, meaning the tasks didn't depend on one another and have the same dependency.

- "→" means that the task on the **left was required and/or completed before** the task on the right.
  (This does **not** imply relation to other parallel branches.)

- If an arrow "→" is on a different line **with nothing to the left of it**, refer to the closest **parallel branch above** to find the originating task.



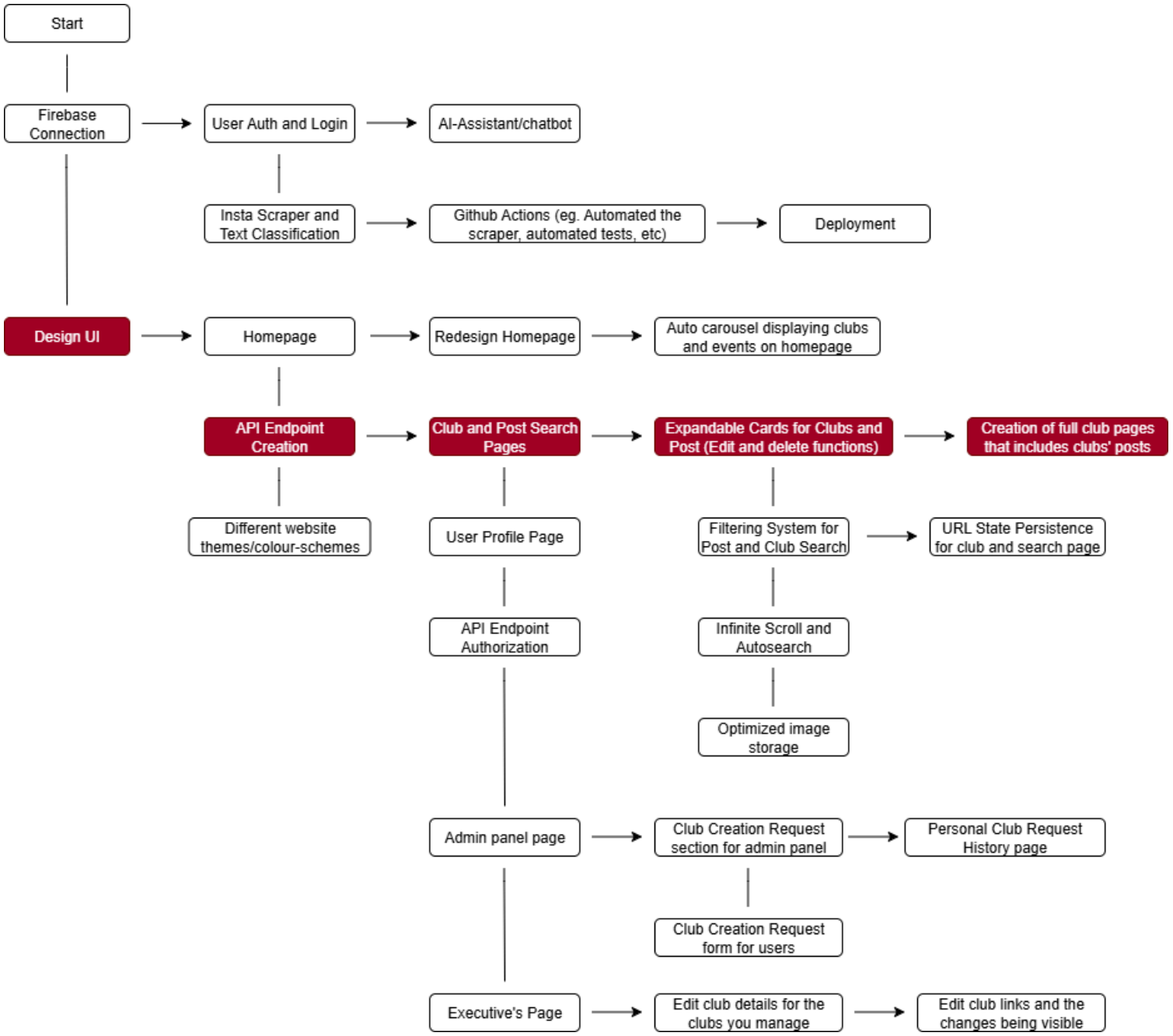
Task Dependencies:

The task dependencies talked about here are focused around the tasks completed in Sprint 3, but tasks in Sprint 3 depended on tasks finished in previous sprints which is mentioned in the “Depends On” section

Task	**Depends On**	**Description**
AI Assistant / Chatbot	Firebase Connection, User Auth	Implements a fully functioning AI chatbot to assist users
GitHub Actions CI/CD	Insta Scraper, Project Repo, Dockerfile setup	Automates testing, building, and deployment pipelines
Deployment	Dockerized Setup, GitHub Actions CI/CD	Full live deployment of site and services through container-based delivery
Personal Club Request History Page	Club Creation Request Form, Admin Panel Page	Users can view their submitted club requests and their approval status
Executives Edit Club Links	Executive’s Page, Club Management Panel	Allows executives to add/edit external links for their club profiles
Optimized Image Storage	Club Post Pages, Firebase Backend	Images stored efficiently to reduce load time and bandwidth

Critical Path:

This represents the longest path of dependent tasks being completed (up until now) based on the network diagram since the tasks need to be dependent. The path is not accurate to the timeline of this project, meaning the tasks in the path are not completely in sequence of when they were done, but are in sequence of their dependencies.



Risk Areas:

Risk Area	**What Could Go Wrong**	**Mitigation Strategy**
Misestimating Effort for New Tech	Docker, GitHub Actions, and E2E testing took longer than expected due to inexperience	Set aside dedicated time for setup and assign pairs to handle unfamiliar tools.
Disorganized DevOps Planning	CI/CD and Docker steps lacked a structured rollout plan	Define a clear sequence of setup steps and assign one person to be responsible.
Delayed Communication	End-of-semester workloads caused slow updates and task misalignment	Schedule mid-sprint check-ins and require status updates in the shared task board.
Final Sprint Cramming	UI fixes and deployments were left to the end of the sprint	Lock down the feature list earlier and set internal deadlines ahead of the sprint end.

Sprint Scheduling Plan:

When it comes to keeping the sprint on schedule, one of the main things that we do bi-weekly check-ins. Through this method, we make sure that the team is working on their tasks as expected and ensure that nothing unexpected has risen within the schedule. This also makes sure that team members have moments where they can ask questions and express any issues they have (technical or outside of work) that are impeding their work. Biweekly check-ins are also just a supplement to having at least 3 standup meetings throughout the week, which like the biweekly checkins, makes sure everyone is on track and ensures an environment where anyone can speak up about any problems. Along with that, we also have an open and active discord chat and also know each others' personal contacts so that urgent issues can be expressed and resolved as fast as possible, as to reduce the number of hurdles each person has to cross while doing their tasks.