**User-stories & critical user-journeys**

XOrithm – Server Status Dashboard

Written by: Mayer Mamdouh

**User Stories**

1. Login via Google or Github

* As a user, I want to log in securely using Google or GitHub, so I can access the dashboard and view server statuses.
  + I should be redirected to the home page if I'm not logged in.
  + Once I log in, I should be taken to the dashboard automatically.
  + I can choose either Google or GitHub login.

1. View Server Overview
   * As a user, I want to see a list of servers showing their current status (Up, Down, Degraded) with visual indicators.
     + Each server shows a status badge using colors (green, yellow, red).
     + The data is static and created manually to simulate real server status.
     + It's easy to quickly scan which servers are working fine and which aren't.
2. Filter by Status
   * As a user, I want to filter the servers by their current status to quickly focus on problematic ones.
     + There's a dropdown to choose between All, Up, Degraded, and Down.
     + The list updates instantly based on what I select.
3. Sort by Uptime
   * As a user, I want to sort servers by their uptime (from low to high or high to low) to detect the least stable servers.
     + Sorting works on top of the filtered results.
     + It's useful for checking performance trends.
4. View Server Details
   * As a user, I want to click on a server and see more info like IP, response time, uptime history.
     + A popup opens with full details about the selected server.
     + Each day in the uptime history is visual, and there's a tooltip that shows the day, uptime % ,date , and status
5. Search for Servers by Name
   * As a user, I want to search servers by name so I can quickly locate a specific system.
     + A search input is displayed near the filters.
     + Typing filters the server list live (case-insensitive)
6. Display Number of Matching Servers
   * As a user, I want to see how many servers match my selected filters.
     + A text indicator shows the current number of visible servers.
     + It updates dynamically when filters or search change.

**Critical User Journeys**

1. Journey 1: Logging in
   * I land on the homepage.
   * I click "Sign in with Google" or "Sign in with GitHub".
   * After logging in, I'm redirected to the dashboard page.
2. Journey 2: Checking Server Health
   * Once I'm in the dashboard, I can see all servers at a glance.
   * Color-coded icons tell me immediately which servers are fine or having issues.
3. Journey 3: Filtering & Sorting
   * I choose "Degraded" from the filter dropdown to only see degraded servers.
   * Then I sort from High to Low uptime to see which ones are still performing well.
   * The list responds instantly to my changes.
4. Journey 4: Getting Server Details
   * I click on one of the server names.
   * A detailed popup opens showing IP, uptime %, response time, and daily uptime history.
   * I hover over each day's history to see more details in the tooltip.

Note: All the server data used here was created manually as static data without using any API.