# Days Calendar Project - Task Breakdown (Group of 2)

## 📁 Setup & Base Structure

* Ticket 1 – Initialize project  
  Create GitHub repo, add index.html, calendar.js, style.css, and days.json. Acceptance: repo initialized, base files exist, server runs.
* Ticket 2 – Load days.json  
  Fetch days.json in JS and log its contents. Acceptance: JSON loads successfully and displays in console.

## 📅 Calendar UI

* Ticket 3 – Build calendar grid structure  
  Render a grid for the month (Monday-first). Acceptance: Days 1–31 show with correct blank padding.
* Ticket 4 – Add month navigation  
  Add Prev/Next buttons to change months. Acceptance: Calendar updates correctly.
* Ticket 5 – Add year/month selector  
  Add dropdowns to jump to a specific month/year. Acceptance: Selecting updates calendar.
* Ticket 6 – Accessibility pass  
  Ensure 100% accessibility in Lighthouse snapshot mode.

## 🧠 Date Calculation Logic

* Ticket 7 – Implement nthWeekdayOfMonth()  
  Reusable function to calculate nth weekday (e.g., second Tuesday). Acceptance: Passes test cases.
* Ticket 8 – Highlight commemorative days  
  Display commemorative days in correct month. Acceptance: Correct highlights (e.g. Ada Lovelace Day).
* Ticket 9 – Add day descriptions (optional bonus)  
  Fetch descriptionURL text on click. Acceptance: Modal shows fetched text.

## 🧾 ICS File Generation (Group of 2 Requirement)

* Ticket 10 – Shared date logic  
  Export calculation logic for reuse in Node and browser.
* Ticket 11 – Generate days.ics  
  Node script to create non-recurring whole-day events (2020–2030).
* Ticket 12 – Verify ICS import  
  Import .ics to Google Calendar and verify correct display.

## 🧪 Testing & Deployment

* Ticket 13 – Add a unit test  
  Write at least one test for date logic. Acceptance: Test runs and passes.
* Ticket 14 – Deploy site  
  Set up GitHub Pages or Netlify for auto-deploy on push. Acceptance: Public link updates.
* Ticket 15 – Final submission  
  Submit repo + deployed link. Acceptance: All rubric points met.

## ✅ Example Task Assignment

|  |  |  |
| --- | --- | --- |
| Ticket Range | Owner | Notes |
| 1–6 | Person A | Frontend (HTML, navigation, accessibility) |
| 7–12 | Person B | Logic + ICS generation |
| 13–15 | Both | Testing, deployment, QA |