HarvestHub: Technical Approach and Lessons Learned

Introduction

“Nothing beats a firsthand experience” by Hans Rosling

HarvestHub, a unique community allotment management system, was designed to streamline the allocation and management of garden plots for community members and the public. This project employs a comprehensive full-stack web development approach guided by modern web technologies and best practices. Our journey in developing HarvestHub was a valuable learning experience, shedding light on the intricacies of application development. This document delves into our technical approach, the lessons we learned, and the potential areas for further improvement.

Technical Approach

Frontend Development: HarvestHub uses three main technology stacks: HTML, CSS, and JavaScript (JS). HTML is the skeleton of the website; it holds everything together. CSS takes care of website styling, making things look pretty with colours, fonts, and layouts. Lastly, JavaScript, a magician who makes things happen on the screen, like enabling the `Assign` button. *Responsive Design:* Imagine the website is water, changing its shape to fit any container, whether a big computer screen or a tiny phone. That's what the Bootstrap Framework does for HarvestHub. It is like having water that fits all glasses, making sure the website looks great no matter where you're viewing it. Server-Side Development: Think of Node.js and Express.js as the traffic controllers of the internet highway. They help handle requests and responses smoothly, ensuring that everything gets to the right place at the right time. It is like having a super-organised mailroom in a big company. *Business Logic:* In HarvestHub, the controller functions, the brains of the operation, fetch data from the database, process it, and decide what to show on the screen. For example, a controller function confirms an assignment, updates the database accordingly, and deletes the corresponding waiting record once the assignment is confirmed. *Server-Side Rendering:* EJS helps create dynamic web pages, so things change based on the actions. For example, the user.ejs template displays content for the user dashboard. First, Express.js routes fetch necessary data from the database and render the EJS template with this data. *Modular Page Design:* EJS templates are building blocks that can be reused to make different parts of the website, like Lego pieces. This makes it easier to manage and keeps everything looking consistent, like having the same header and footer throughout the website. *Database Utilisation*: PostgreSQL keeps track of things like user info and requests, like the librarian organising all the books in the library. The controller functions talk to PostgreSQL, asking for info or updating stuff when needed, just like checking out a book or returning it to the library. *Environment Configuration:* The dotenv library is like a vault for secret codes. It keeps important stuff, like database credentials, safe and hidden from prying eyes. This ensures that sensitive info stays protected, like locking your diary with a secret key. This comprehensive technical approach ensures the success and reliability of HarvestHub, giving you the confidence that it meets the project goals.

Lessons Learned

Frameworks and Libraries Magic: Frameworks and libraries are like magic tools for developers. They speed up the work and make the project easier to grow and maintain. Instead of reinventing the wheel, we can use these tools to focus on the unique features of your project. It's super important to pick the right ones for our needs. *Power of Modular Design:* Reusing code in different parts of the project makes everything easier to handle. We can fix our issue in one place and not multiple places in our code. This approach not only helps us find and fix errors quickly but also makes updates simpler and ensures everything looks and works consistently across the application. *Mastering Database Management:* During our testing phase, I ran into some issues with duplicate keys in the database. This showed me how crucial it is to manage database sequences properly. I fixed it by resetting the sequences to match the highest values in the tables. I can imagine this problem will be hard to tackle, especially for a more complex project. *Feedback, Adapt and Repeat:* One big lesson from this project is the importance of always learning and adapting based on feedback. We improved the application step by step, making each iteration more user-friendly. This ongoing commitment to learning and improving is what makes HarvestHub special.

Conclusion

HarvestHub was created with clear goals in mind as a project to practice modern full-stack web development. While we've hit our initial targets, there are still a few areas where we can make things even better.

For example, we have a feature that automatically checks when a plot's agreed period, like a 12-month contract, is over. When it comes to security, we want to make user login even safer by using the best methods for password protection and session management.

The lessons we've picked up along the way are like a guide for future upgrades. Building HarvestHub taught us how important it is to keep learning and adapting, especially during the development process. It's all about getting and using feedback to make the system better from start to finish. HarvestHub shows how ongoing learning and adapting can make any web app reliable, not only HarvestHub but also easy to maintain, giving real value to its users.