**A Report on my Personal Startup idea “CodeWeed.”**

Prepared by ~Parthiv Gajjar (B. Tech, Computer Science and engineering)

To build a full-stack web app for “Code Weed,” which handles charity (food, clothes) and a tree plantation tracking feature, the app should be designed with clear front-end and back-end separation, secure data management, and a friendly user interface. The solution should allow users to fill in details for planting trees (location, name, tree type), store and fetch this in a MySQL database, and offer admin capabilities to view submissions.

**Core Architecture**

The app should have:

* A **front end** for users to interact with (submit forms, see status, browse info).
* A **back end** that stores/fetches data (user entries, tree details), sends emails, processes logic, and manages admin access.
* A **MySQL database** to store all submission records.
* Optional features (tree location map, progress tracking, etc.) can be added for value.[[4]](#fn4)[[1]](#fn1)

**Tech Stack Recommendations**

* **Front End:** Use React.js (or similar like Vue.js/Angular) for dynamic forms and smooth experience. HTML5, CSS3, JS for the basics.
  + Forms for tree submission, charity offerings, etc.
  + Google Maps API for users to pick/verify locations for tree plantation.
  + Admin dashboard for data viewing and management.
* **Back End:** Node.js and Express.js (popular with React), or Django (Python), or Laravel (PHP) are good options.
  + REST API endpoints (POST for new trees, GET for admin/fetching lists, etc.).
  + Integration with MySQL for storing data securely.
  + Email notification service (e.g., Nodemailer for Node.js or built-in for Django).[[2]](#fn2)[[6]](#fn6)[[5]](#fn5)
* **Database:** MySQL (suits your needs—relational, bright ecosystem).
  + Tables: users, trees, donations, email\_jobs, etc.

**Key Features and Components**

* **User-facing:**
  + Register/Login (optional, for future personalization).
  + Charity form (food/clothes requests).
  + Tree plantation form (fields: name, location [lat/lon or address], tree type).
  + Tree-mapping view (see where trees were/are being planted).
  + Confirmation email after submission.
* **Admin-facing:**
  + Dashboard with data of all users and their submissions.
  + Data filtering/search (by location, by name, by tree type).
  + Option to download/export data.

**Step-by-Step Build Plan**

**1. Design Database Schema**

* Create tables for trees, users, charity requests, and any other data models.
* Example (simplified):
  + trees: id, user\_name, tree\_type, location\_lat, location\_lng, submitted\_at

**2. Set up the Backend API**

* Design RESTful endpoints:
  + POST /api/trees — add new tree
  + GET /api/trees — list all trees (admin)
  + GET /api/trees/:user or /api/trees?location=xx — filter
* Connect backend to MySQL.
* Implement email sender.
* Protect admin endpoints (authentication).

**3. Build the Front End**

* Use React for form input and submission, tree listing, and summary.
* Use Axios/fetch to call API endpoints.
* Integrate Google Maps for picking/showing tree locations.
* Display feedback messages, loading spinners, etc.

**4. Finish, Test, and Deploy**

* Test front-end and back-end separately and together.
* Use tools like Postman for API testing and Jest/React Testing Library for front-end.
* Deploy to platforms like Vercel (front end), Heroku/Render (back end), or full-stack deployment on AWS/Azure (if budget allows).

**Example Tech Stack Table**

|  |  |  |
| --- | --- | --- |
| Layer | Recommendation | Details |
| Front End | React.js | Responsive UI, easy integration |
| Back End | Node.js + Express, Django, or Laravel | RESTful API, handles business logic |
| Database | MySQL | Stores all user/tree/charity info |
| Email | Nodemailer (Node), SendGrid, or Django’s email backend | Sends tree-plant emails |
| Maps | Google Maps API | Location picker and visual map |

This approach will give “CodeWeed” a sustainable, scalable system ready for your community-driven mission.[[3]](#fn3)[[1]](#fn1)[[2]](#fn2)[[5]](#fn5)

<div style="text-align: center">⁂</div>

1. <https://www.ijraset.com/research-paper/software-based-tree-plantation-and-live-tree-tracker>

1. <https://www.maxiomtech.com/backend-tech-stacks-empower-nonprofit-solutions/>

1. <https://syndicode.com/blog/web-application-architecture/>

1. <https://www.thetreeapp.org>

1. <https://www.abbacustechnologies.com/how-to-build-custom-donation-platforms-for-charities-3/>

1. <https://fullscale.io/blog/best-tech-stack-for-web-development/>

1. <https://dev.to/itsrakesh/the-flow-of-building-a-full-stack-web-application-447g>

1. <https://is.muni.cz/th/up1lu/thesis.pdf>

1. <https://play.google.com/store/apps/details?id=com.mahapwd.tree.plantation&hl=en_IN>

1. <https://www.capitalnumbers.com/case-study/backend-and-frontend-optimizations-for-a-nonprofit-community-portal.php>

1. <https://www.clickittech.com/software-development/web-application-architecture/>

1. <https://www.ecomatcher.com/treeapp/>

1. <https://www.grow-trees.com>

1. <https://dotdigital.com/blog/charity-tech-stack/>

1. <https://flora.appfinca.com/en>

1. <https://github.com/topics/donation-website>

1. <https://www.treeclicks.com>

1. <https://www.freecodecamp.org/news/how-to-build-a-full-stack-application-from-start-to-finish/>

1. <https://www.studiorepublic.com/blog/why-charities-can-benefit-from-a-custom-web-application/>

1. <https://www.assemblysoft.com/case-studies/charity-software-development-with-donations-for-endangered-species>