

Week 1-2

Assigned to each task is a timeline, and persons responsible. The first name represents the person that will be doing the task (**Assignee**), the second name is the person that will be reviewing the task (**Reviewer**). There is a lot of dependency in the tasks therefore Assignees don't have to wait until their task is completely done before submitting, so others can pick it up and move along with their own tasks. ***Please let's not create bottleneck*** Let's try to do it as fast as possible. If your task is pending or difficult or you can't meet up with the timeline indicated in the group, there is a high chance task is assigned or partly taken over by the reviewer. These tasks are considered last week and this week. On Thursday we have a short meeting to evaluate how far, and I look forward to submissions (PR) or completion latest Sunday by MIDNIGHT. TL will assign a pair of team members to review PRs as they come in.

Tasks

1. Project definition and tasks - 1 day / **Tane - Clifford**
2. Schema/Model design - 4 days / **Herve - Nfinyoh**.
 - Come out with a schema/entity document, identify the different entities from project definition. In case of doubts, discuss with team members.
 - Use tools that can export schema design to png or pdf, checkout this one or alternatives you most comfortable with
<https://hackolade.com/nosqldb/mongodb-data-modeling.html>
 - Regarding the structure of public contract, this document can help extract properties related to a typical project.
https://www.afdb.org/fileadmin/uploads/afdb/Documents/Project-and-Operations/Cameroon_-_The_National_Governance_Programme_Support_Project_-_Appraisal_Report.pdf consider section IV and V for project fields.
 - The more entities designed the better, schema design should cover at least 70% of project entities
3. Create schema/models - 4 days / **Nfinyoh - Herve**
 - Convert all design in **Task 2** to code or mongo models
4. Database configuration, linting configuration and test configuration- 2 days **Tane - Alex**
 - Configure code to follow guidelines recommended by Andela
 - Use Jest for tests
 - Use eslint for linting

5. API definition - 4 day - **Alex - Tane**

- Come out with endpoints for the project
- Use the openapi v3.0.0 specification
- Make use of platforms that can export the api docs to pdf,html, yuml or json
- You can checkout the following platforms <https://editor.swagger.io/>,
<https://mermade.github.io/openapi-gui/>
- Make use of schemas designed in **Task 2**
- In case of doubts please post in the group

6. Api implementation - 1 days/ **Nfinyoh - Herve**

- basic Authentication
- password recoveries

7. Api implementation - 1 days/ **Herve - Nfinyoh**

- User change password
- Update user profile

8. UI/UX theming (interface theme definition) - 2days/ **Tane - Clifford**

- Colors
- Button types and colors
- Backgrounds
- Typography
- Fonts
- Checkout <https://www.figma.com/>
- Resource to checkout <https://atomicdesign.bradfrost.com/table-of-contents/>

9. UI/UX Mockups - 10 pages - 4 days / **Clifford - Tane**

- Mockup 10 pages (Landing, home, authentication (login/signup), Project (CRUD), User CRUD)
- It will be great if more than 10 pages could be designed
- Resource to checkout <https://atomicdesign.bradfrost.com/table-of-contents/>

10. Front-end (page implementation) / 1 days **Alex - Clifford**

- Front end setup, css library , will suggest libraries with small file size or minimalist design like <https://milligram.io/> or other for faster load times and minimum js requirements.
- Equally configure front-end React unit testing
- Html implementation of Layout,

11. Front-end (page implementation) / 1 days **Alex - Clifford**

- Html implementation of login, singup, forget password pages/widget