

Team Members:

Aashish	19BCE0971
Priyanshu	19BCE0550
Srikanth	19BCE0158

Abstract:

Bug Tracking/Issue Tracking(Project Management tool)

It is a simple tool that provides a list of errors and bugs encountered by the developers during the development process. It will provide a summary of each bug that is reported, along with details such as the description of the bug, who has logged the bug, etc. It will use a database to store this list. The application will retrieve this list each time a user opens up the application so they are always working with the latest list of bugs/errors. It includes basic features like adding, deletion and searching for bugs, refresh bug list. Each bug report will have basic attributes like its name, bug description, name of the person who generated the bug, estimated time to fix it, etc. Furthermore, we aim to create an application that we ourselves can use in our projects in the future.

Stakeholders:

Stakeholders are the people who are directly or indirectly involved in the project and will either be developing or using the software there are two categories of stakeholders internal and external. Internal stakeholders are ones who are involved in making the software and external stakeholders are people who will be using the software. The stakeholders involved in our project are

- Internal Stakeholders

- Project Manager : Aashish
Responsible for managing time and deadlines of the project such as creating a gantt chart and keeping the teams within the deadlines, all the members contribute to project management equally.
- Project Team: Aashish, Srikanth and Priyanshu

Project team consists of 3 people and we are responsible for creating and managing the development of the project.

- **External Stakeholders**

- Software Developers: They are the users(customers) that will use the software to aid them in the development of their software. Not only will they be able to remotely keep track of bugs among their team but also be able to prioritize the bugs based on their nature and impact on the software.
- Software Testers/Users:
They will be testing the software to see if there exists any exploits or bugs in the system, they will be the users that will report any bugs that have not been foreseen in the development phase and thus send the feedback to the developer team.

Process Model:

Our project will be using the **Evolutionary development** model as it is a relatively small project. Evolutionary model suggests breaking down work into modules, prioritizing them and then working on them one by one. The main advantage is that the developers as well as the testers can work on each module at a time ensuring that the project is tested thoroughly at each stage. So for us, this means that we can first create a basic bug logging functionality having both frontend and backend and keep adding features based on suggestions from our project mentor. The model allows for changing requirements as well as all work is broken down into maintainable units. We will be going along with the **exploratory development** method and modify our frontend(UI) and backend, based on feedback from our mentor and from our own experience from using the application as a user.

Justification:

- 1) Our bug tracking system has a particular range of modules so following an evolutionary model would allow us to implement these features one by one in a modular method.
So we can learn as we progress and incorporate features based on feedback as well as our own experience.
- 2) Using this method will ensure that each unit/feature like bug search, modify, etc., gets tested thoroughly before proceeding. This will help prevent major bugs in the final deployment.
- 3) Evolutionary process model also provides a better understanding of the structure of our bug tracking system.
- 4) Specific features can be given priority in the development phase like bug deletion.

Justification against other models:

Waterfall

Since the entire structure is not clear to us from the start i.e. all the features to be included as well as design of the project, it is illogical to use the waterfall model.

If we have more time, we may decide to add more features like advanced bug search and suggestions/comments section for logs submitted by other users.

We may also decide to not implement certain functionalities

Incremental Dev

Since the number of requirements are less (as of starting point) there will be no need to develop the project in phases.

There aren't enough modules for us to follow this approach, i.e., We cannot deliver the database component, frontend and backend incrementally. It has to be one integrated unit.

Rapid Application Development

Our stipulated time period (2-3 months) will be enough for us to make the tool.

We also don't have enough people involved to facilitate parallel and simultaneous development of different components of the project.

Spiral Model

Spiral model is more suited for large projects which also require risk management and our project is not very large or complex and does not have much risks involved

Work Breakdown(Development Process)

Drawn using lucidchart

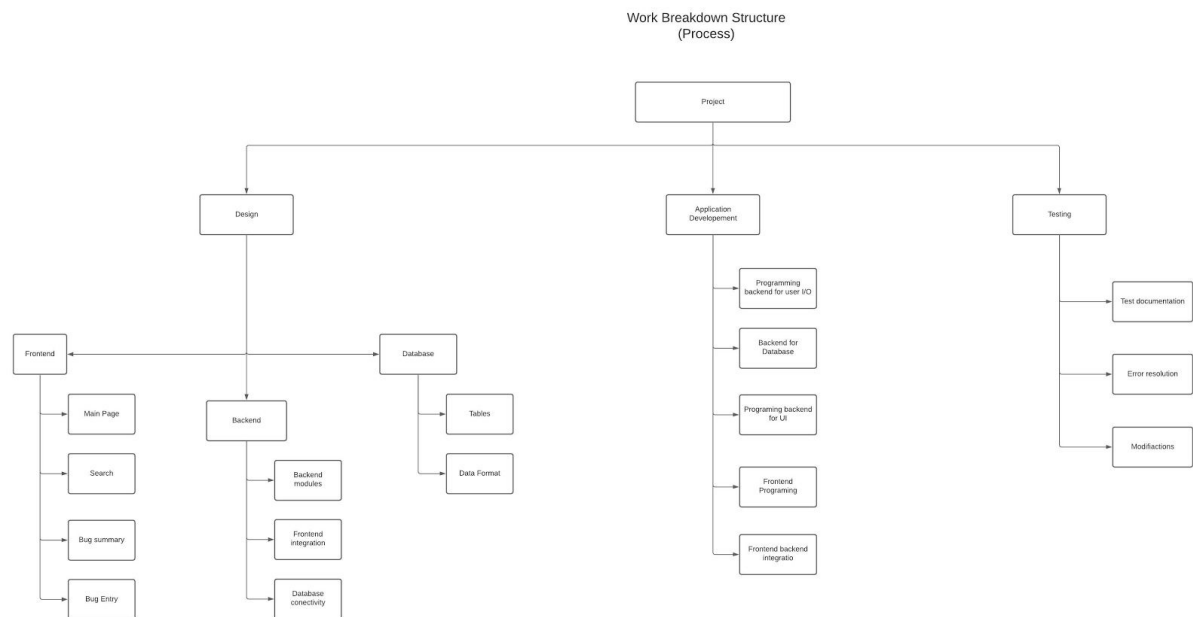


Fig.1 Work breakdown structure(Development Process)

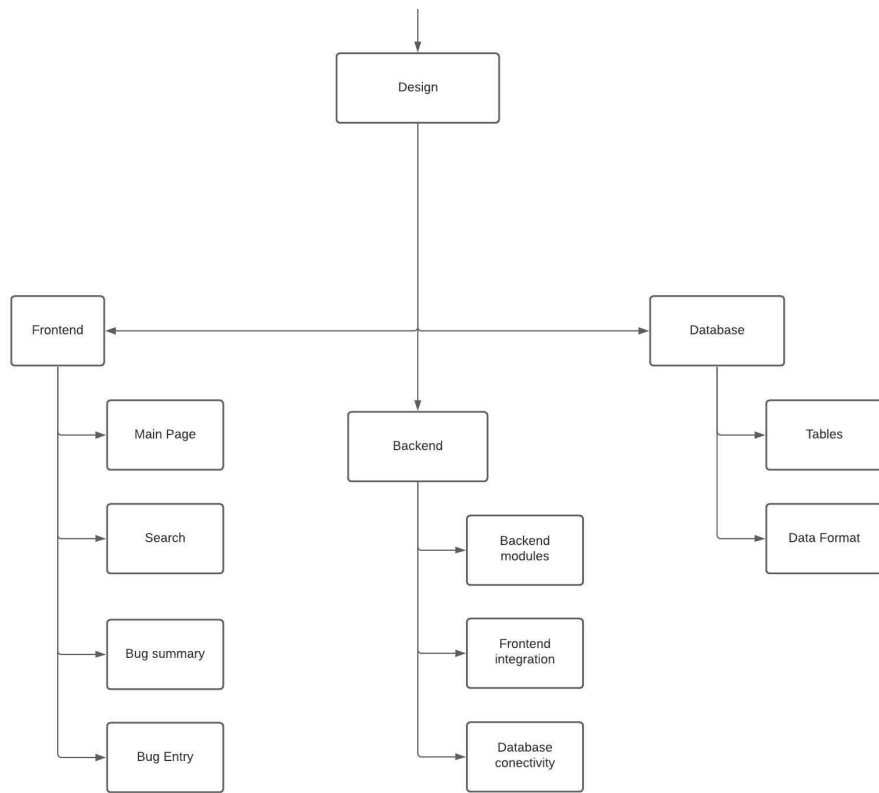


Fig 2. Design Work Breakdown structure

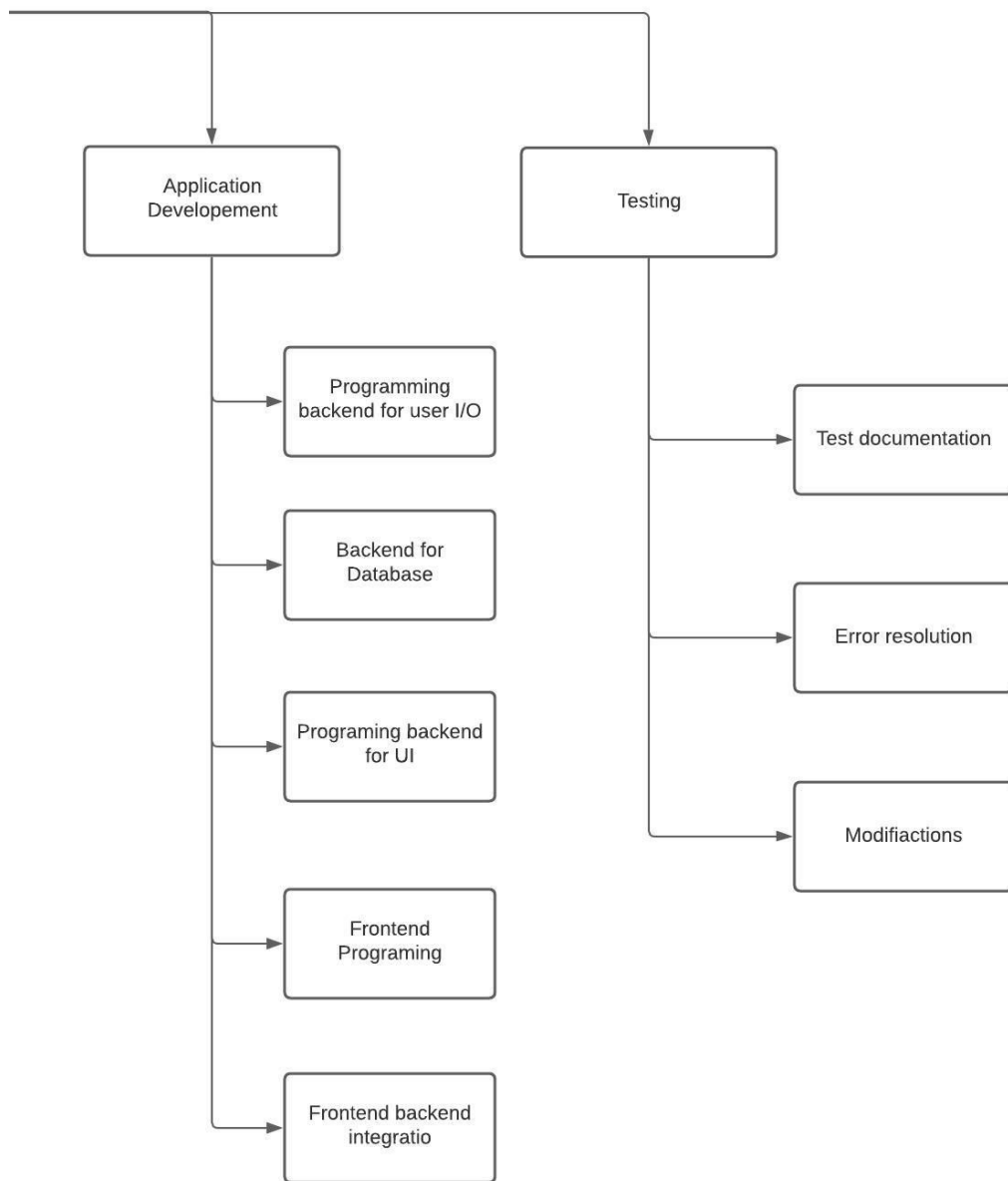


Fig 3. App. Dev. and Testing work breakdown

Work Breakdown(Product)

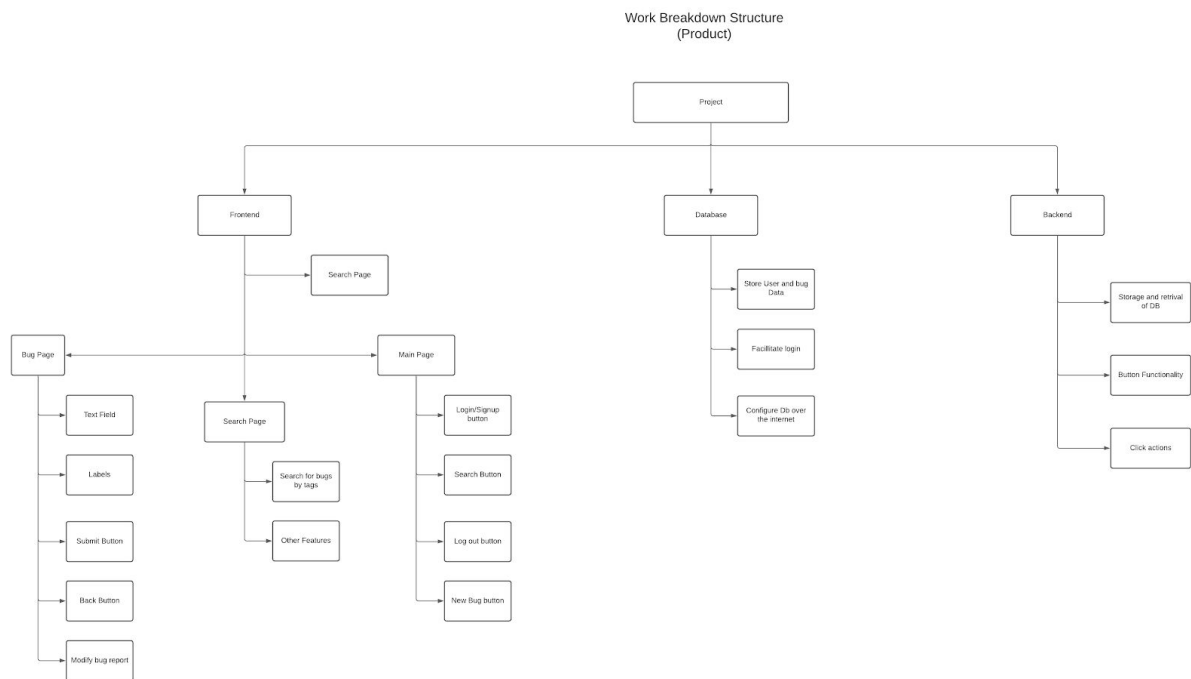


Fig.1 Work Breakdown Structure overview (Product)

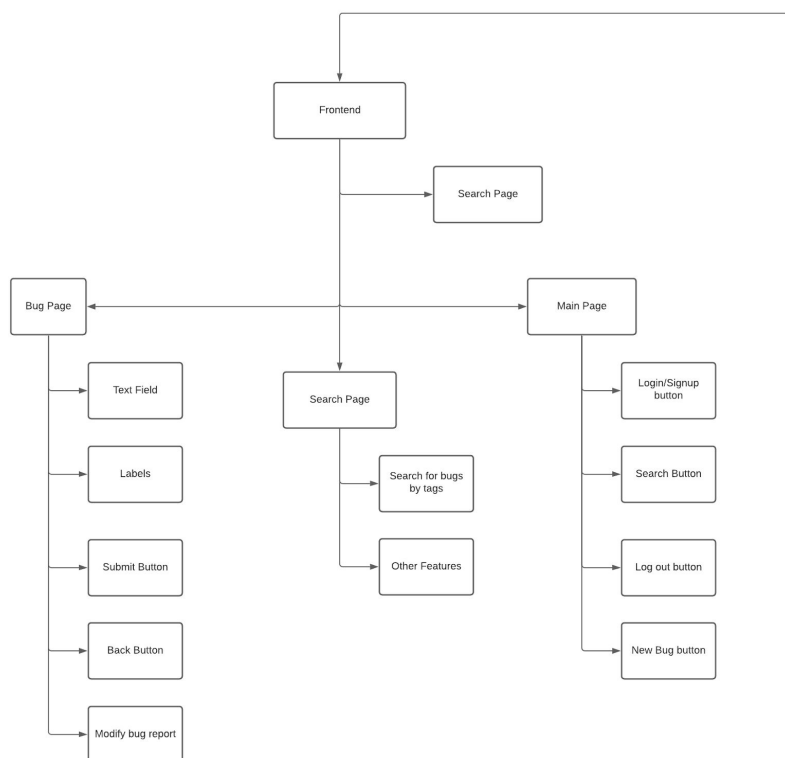


Fig. 2 Frontend Project breakdown structure

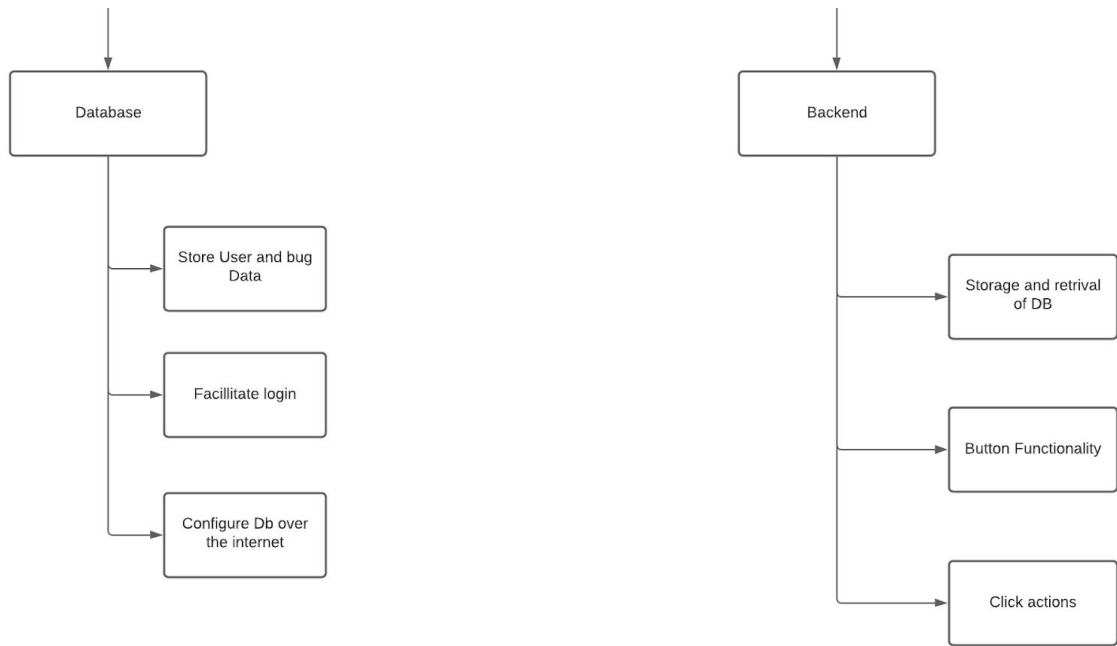
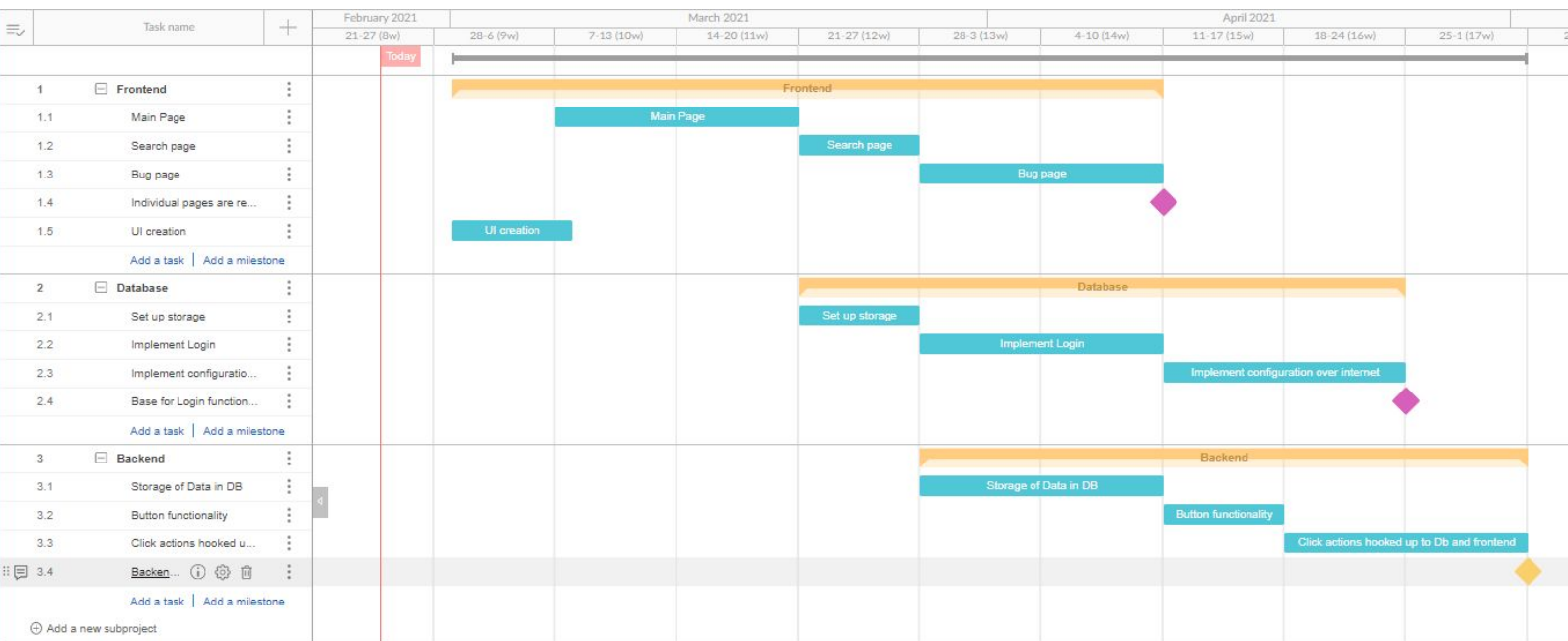
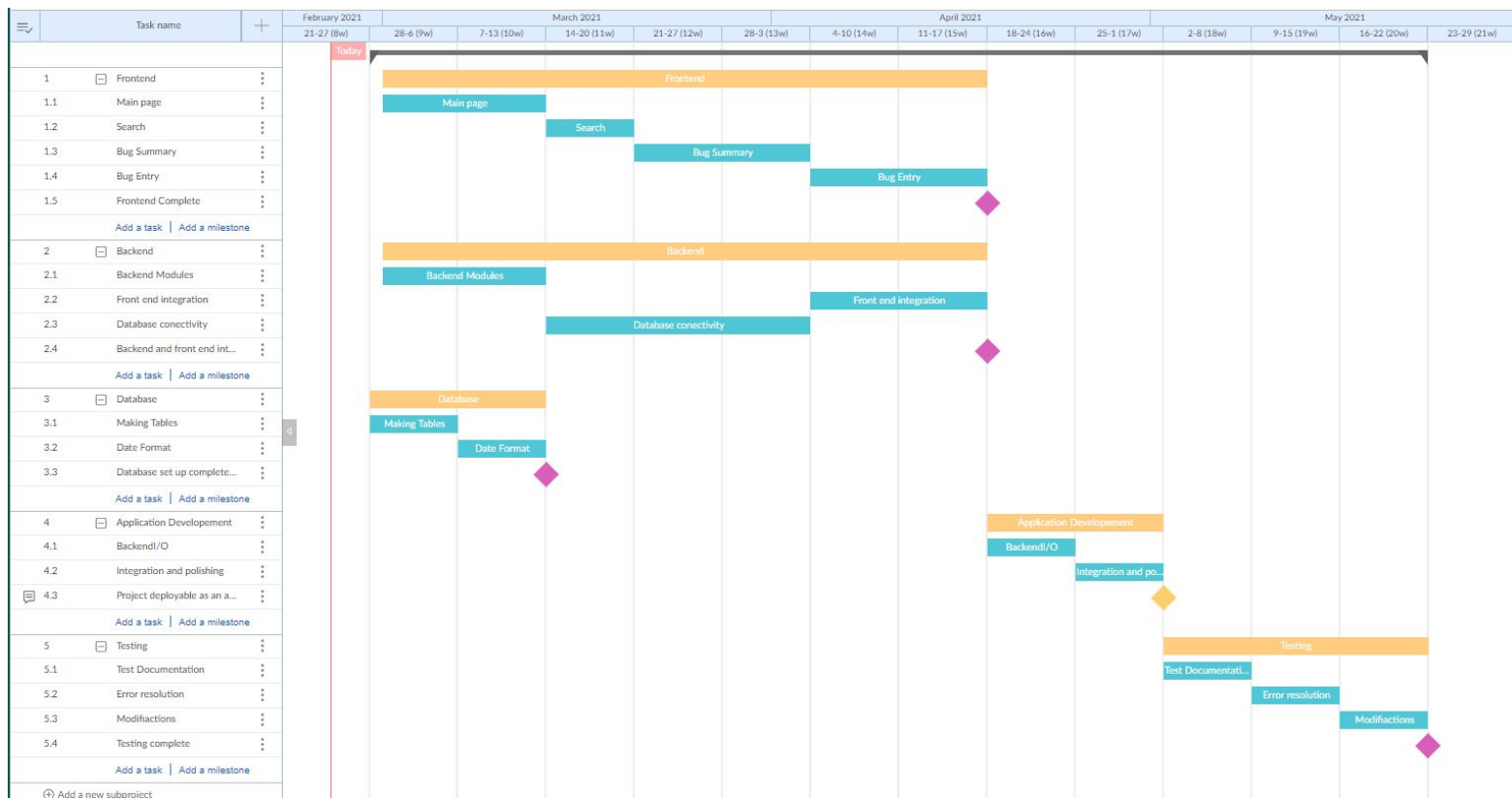


Fig.3 Backend and Database Project Breakdown structure

Gantt Chart (Tool: Gantt Pro)



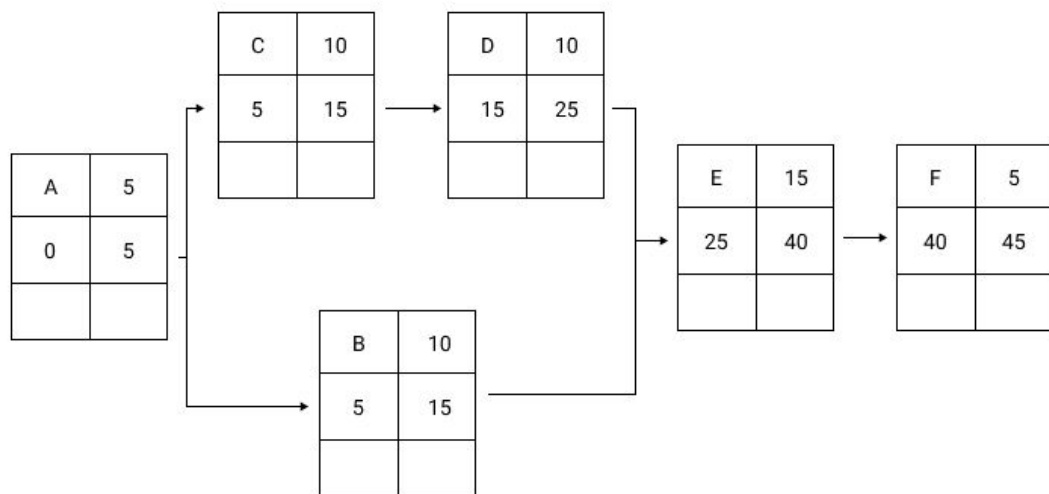
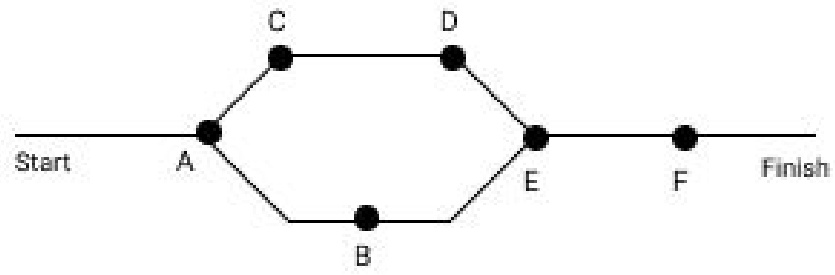
Development process



Activity Network for Development Process as well as Product development

Tool used - Figma

Task	Label	Predecessor	Staff Required	Duration
Design	A	-	1	5
Frontend	B	A	2	10
Backend	C	A	2	10
Database	D	C	2	10
Application Dev	E	ABCD	3	15
Testing	F	ABCDE	3	5



Project Timeline

