**Documentation**

**Why MongoDb ?**

MongoDB is ideal for storing contact information because of its flexibility, scalability, and schema-less nature, allowing dynamic fields for contacts without a fixed structure. It uses a document-based model to store contact details like fname, lname, and email in BSON format, making it easy to represent hierarchical data. MongoDB integrates seamlessly with JavaScript and Node.js using libraries like Mongoose, simplifying development and enabling efficient CRUD operations. Its support for indexing and advanced querying enhances search performance, while built-in scalability through sharding ensures smooth handling of growing data.

**Challenges and Solutions**

**Time Management**

The most important and difficult challenge that I faced while building this project was the fact that I had my exams also during building this. I tried my best to manage the exams with the deadline, but I was lacking behind a bit, because of the technical errors I was facing also the exam pressure was constantly on my head.

Solution:

I had to create a detailed timetable and had to dedicated follow it. Time management and to handle both was a bit challenging but this was something new to me and I didn’t want to lose such an opportunity. Being dedicated has helped to complete this project not exactly on time but I’m happy I was able to complete this.  
 **Technical Errors/Challenges:**

**Pagination Issues with Table Component**

About the technical challenges, I initially faced issues when I was trying to setup the pagination part as it was not being done correctly. For instance I have set to 5 rows per page so even after 5 rows it wasn’t able to move to the next page.  
Solution:   
I had to go through the pagination logic once again and use console.log to check the errors. Then I found I was not correctly integrating the pagination value correctly.

**Data Fetching from Backend to Frontend**

In the start, I had some errors in fetching the data from the backend or even sending the data to the backend. The result I used to get is “500: Internal Server Error” . I checked for multiple times. After going through the code again and again, I finally understood the logic.

Solution:  
The solution I found to this was

1. Cors policy had been disabled, so I had to enable that.
2. The schema and values which were being passed were not matching so I had to modify the value and keep a note everywhere.

During these errors, I realised that we need to check each and every small data value that we declare.