**Design Patterns:**

1. **MVC**
2. Singleton (Single Instance of service or database)
3. Adapter (Sort) **🗸**
4. Chain of responsibility (Send requests to backend as string and the appropriate class/function will handle it)
5. Filter
6. Builder (Create new profile)
7. Proxy (Between frontend and database, etc..)
8. Prototype (Send mail to multiple people)
9. Immutable (mail and contact) **🗸**

**To Do (till Deadline):**

1. **Change username to Email (compose) (Done)**
2. **Add option to search in all attributes (Done)**
3. **Move to folder should show a dropdown list of all available folders**
4. **Checkbox all for checking all emails**
5. **Sort: remove by attachment size, add by body (Done)**
6. **Research how to handle upload/download between frontend**
7. **Requests: (Not 100% Done)**
8. **Login and Sign Up (Login.jsx and SignUp.jsx) (Done)**
9. **Compose (handleMoveToDraft, handleSubmit) (Compose.jsx) (Done Front: I POST an object that has the username and the formData(email), change it if you want)**
10. **Drafts Folder (handleEditOrSend) (Drafts.jsx) (Done)**
11. **Handle adding/deleting/renaming/requesting folder names to/from backend (EmailPage.jsx and Folder.jsx) (Done)**
12. **Contacts all handlers (Contacts.jsx) (Done)**
13. **Drafts and DefaultFolder all handlers (Drafts.jsx, DefaultFolder.jsx) (Done)**