

A Portal for sharing and digitally signing documents

*Phase - I mid-term report Submitted
for the B.Tech Project*

By

Dasari Sravan Kumar (111701010)

and

Muhammed Yaseen (111701032)

under the guidance of

Dr. Albert Sunny



INDIAN INSTITUTE
OF TECHNOLOGY
PALAKKAD

Contents

1. Introduction.....	02
2. Tools and technology.....	03
3. Design.....	04
4. API calls.....	11
5. Database design.....	14
6. Conclusion.....	15

Chapter 1

Introduction

In any organization, there are multiple occasions where many documents have to be passed across multiple people for verification and they will be required to sign in this document to testify that they have verified the same. We had understood the issue quite closely in the context of IIT Palakkad. This process can involve a lot of delay and wastage of time and efforts. Through this project, we propose a solution for this problem by developing a web-based portal for sharing and digitally signing the documents. In the phase - I of the BTP, we are working on developing the portal (complete front-end and some parts of the back-end). This is a summary of the works done so far from August to September 2020 towards the project.

1.1 Organization of the Report

- **Chapter 1, Introduction** - Gives an introduction to the project work, explains the motivation behind it and gives a brief description of its implementation and application.
- **Chapter 2, Tools and Technology** - Describes the learning process involved to match our skills with the prerequisites for the development of the portal.
- **Chapter 3, Design** - Describes the overall design and layout of the portal.
- **Chapter 4, API calls** - Describes the API calls used by the client to communicate with the server.
- **Chapter 5, Database schema** - Gives an overview of the database schema
- **Chapter 6, Conclusion** - Gives a summary of the works done so far.

Chapter 2

Tools and Technology

We will be using the following technologies extensively for the project

- The Tech Stack We would be using for the project is **MERN Stack**
 - **React.Js** : This is a Client Side Javascript framework mainly used to develop single page web applications . This will be used in the front-end of the application. This is developed by Facebook. So we can guarantee the support from the react developers in the future.
 - We have chosen React.js instead of other peer technologies like Angular,Vue because React has a lot of developer base working on bugs and rapidly developing new features.
 - **MongoDB** : MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and functions which is the equivalent of relational database tables.
 - The reason to choose MongoDB is that we can store unstructured data and its integration with the other frameworks we are currently using.
 - **Node.Js** : This is a server side Javascript framework used for the backend of the application.
 - The reason to use Node.js is that it uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.
 - **Express.Js** :Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications

We have finally arrived at MERN stack as the tech stack for our project after detailed comparisons with the alternate options primarily due to reasons mentioned in the above paragraph. We have spent some time getting used to these technologies and we are confident that we will pick up the rest once we start the implementation.

Chapter 3

Design

In this chapter, we demonstrate the UI design that we will be following to develop the portal. Images are attached wherever it was possible to communicate the design idea clearly.

3.1 Front Page

1. Login / Sign-up
2. About
3. How it works

About

How it works

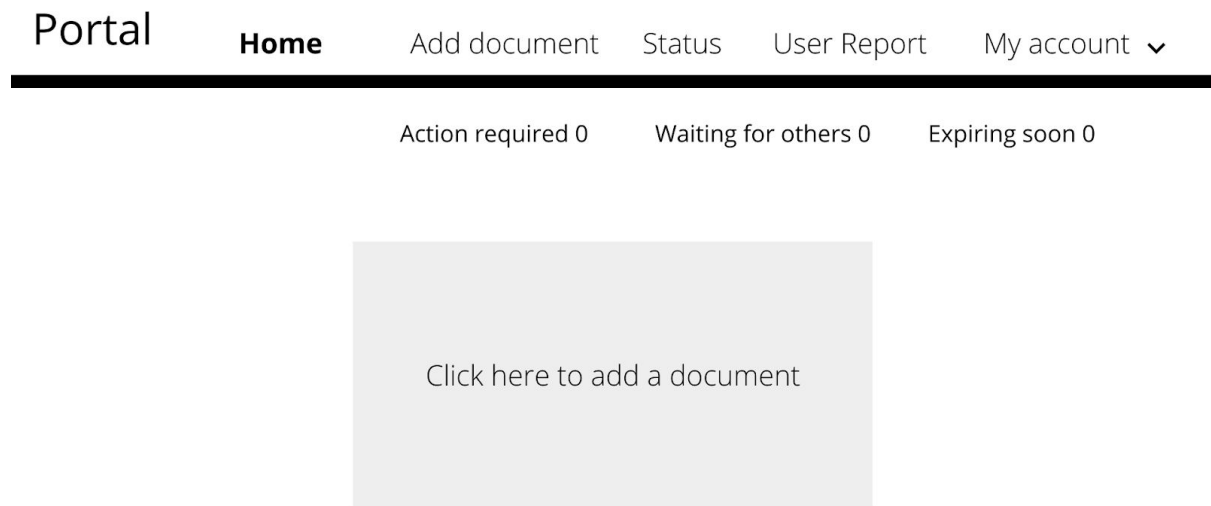
Login / Sign-up

**IIT-PKD
Digital
Signing
Portal**

fig(1)

3.2 Dashboard

1. Home
 - a. To upload a document
 - b. Action required
 - c. Waiting for others
 - d. Needs immediate attention



fig(2)

2. Status
 - a. Action required
 - i. If someone else requests the sign, it will be received under the *actions required* section.
 - b. Waiting for others
 - c. Expiring soon (nearing the deadline)
 - d. Completed
 - e. Failed

Sections

Action required 0

Waiting for others 0

Expiring soon 0

Completed 1

Failed 0

You are done with all documents!

fig(3)

Portal

Home

Add document

Status

User Report

My account

Sections

Action required 1

Waiting for others 0

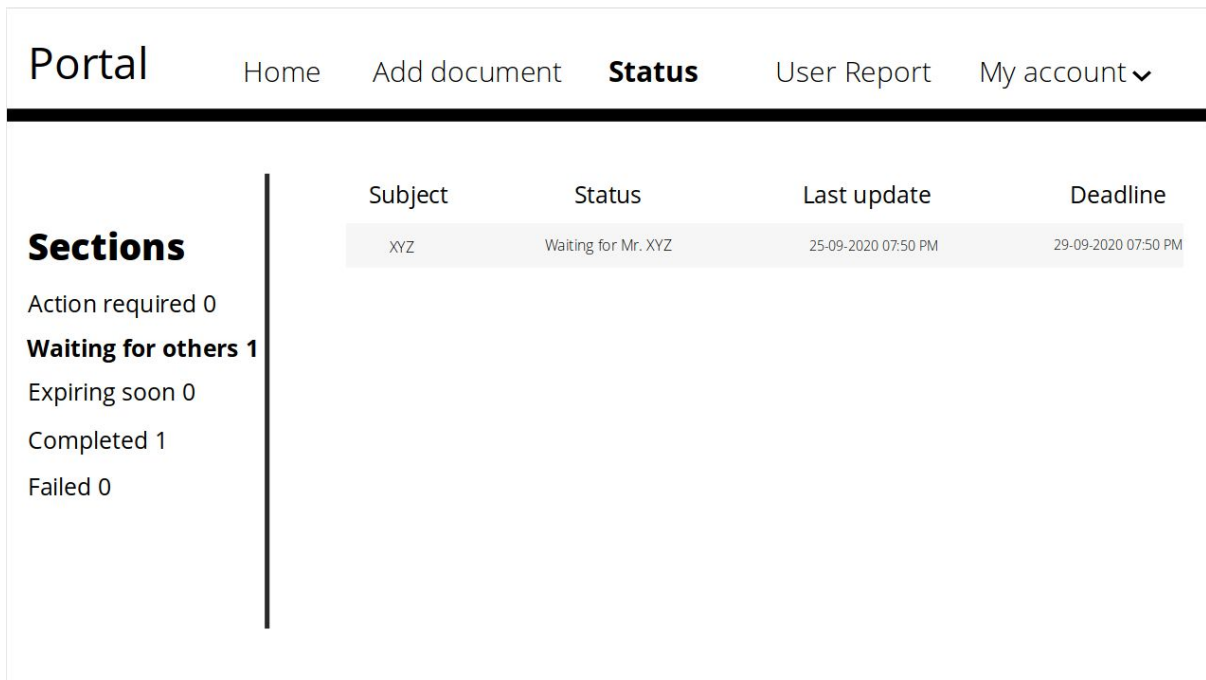
Expiring soon 0

Completed 1

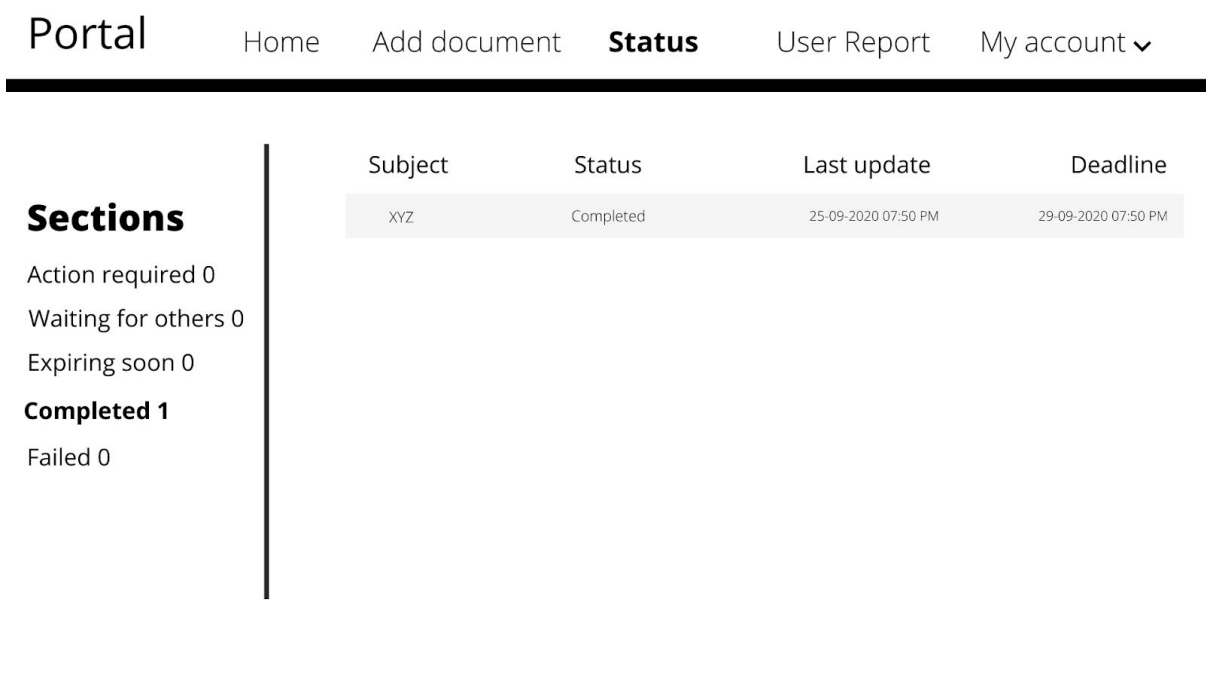
Failed 0

Subject	Status	Last update	Deadline
XYZ	<div>Waiting for your sign</div> <div>Sign this</div>	25-09-2020 07:50 PM	29-09-2020 07:50 PM

fig(4)



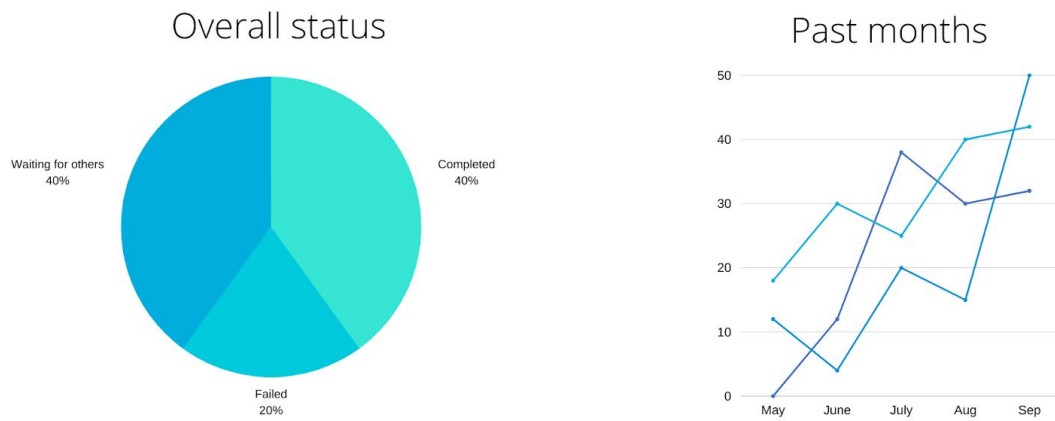
fig(5)



fig(6)

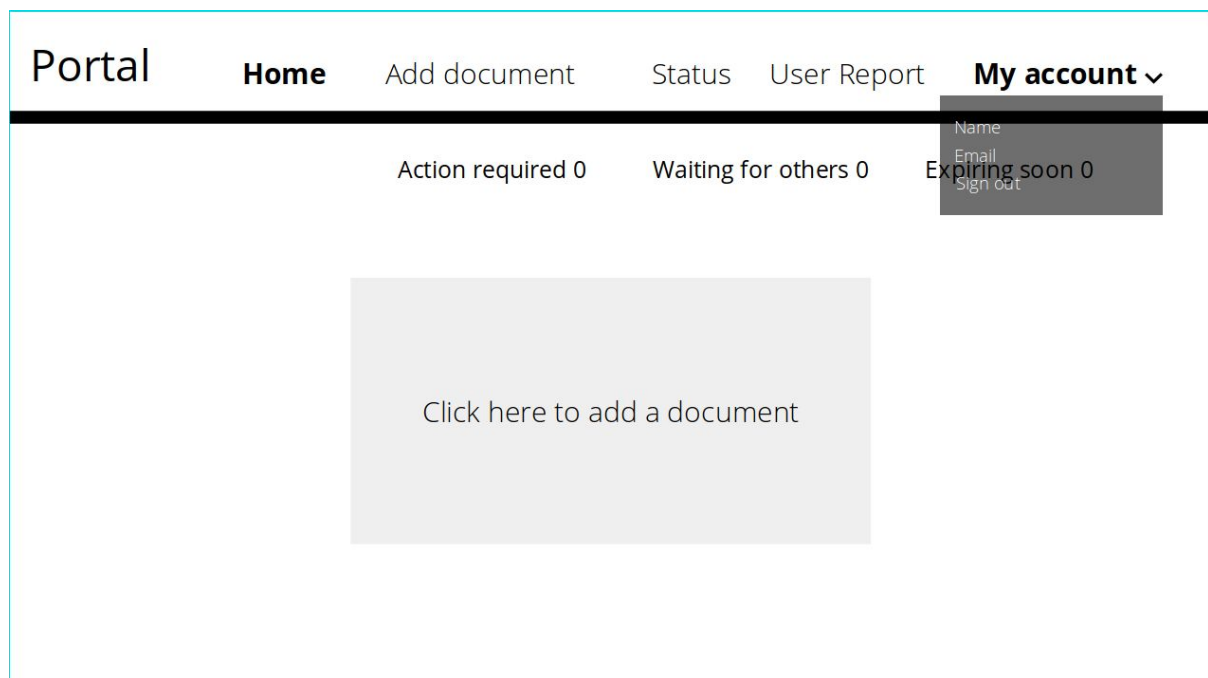
3. Report

- Statistics related to my documents (completed, failed, etc.). More statistics will be added later if it is feasible then.



fig(7)

4. Top right pop-up
 - a. Name, e-mail, sign-out option



fig(8)

5. Upload documents

- Add document (file upload from drive/system)
- Add other people to sign (along with setting the serial order for signing)
- Optional whether the sender should sign or else skip
- Set a deadline for signing
- Generate an automated email with a message and subject to others who are required to sign on the document
- Finish

Portal Home **Add document** Status User Report **My account** ▾

■

■

■

■

Upload Add other people to sign Generate Email Finish

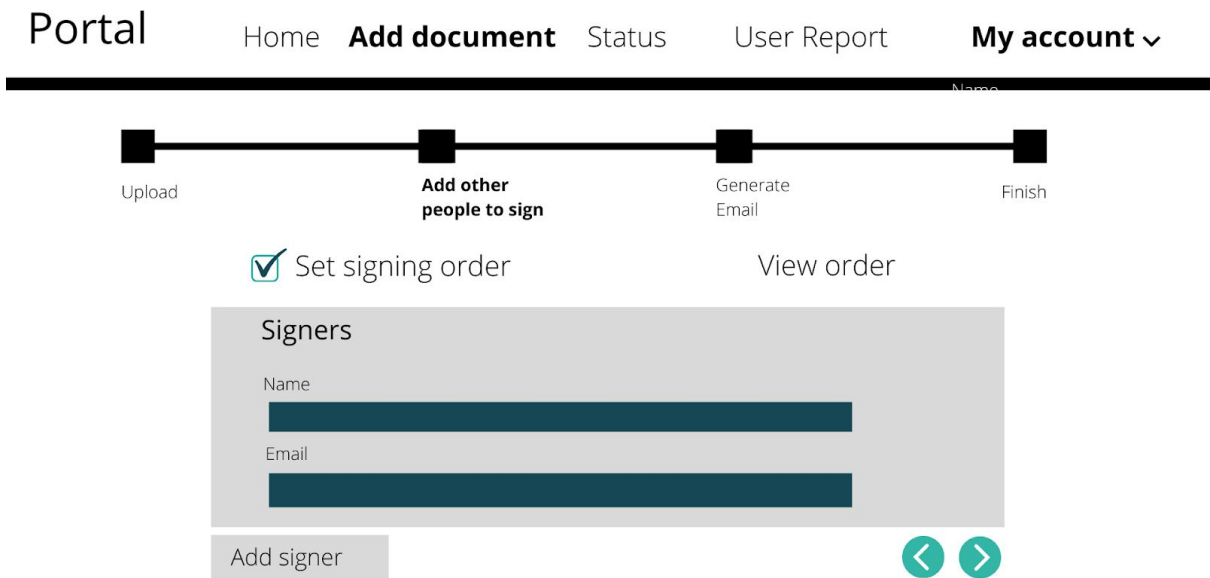
file1.pdf uploaded

Upload from your device

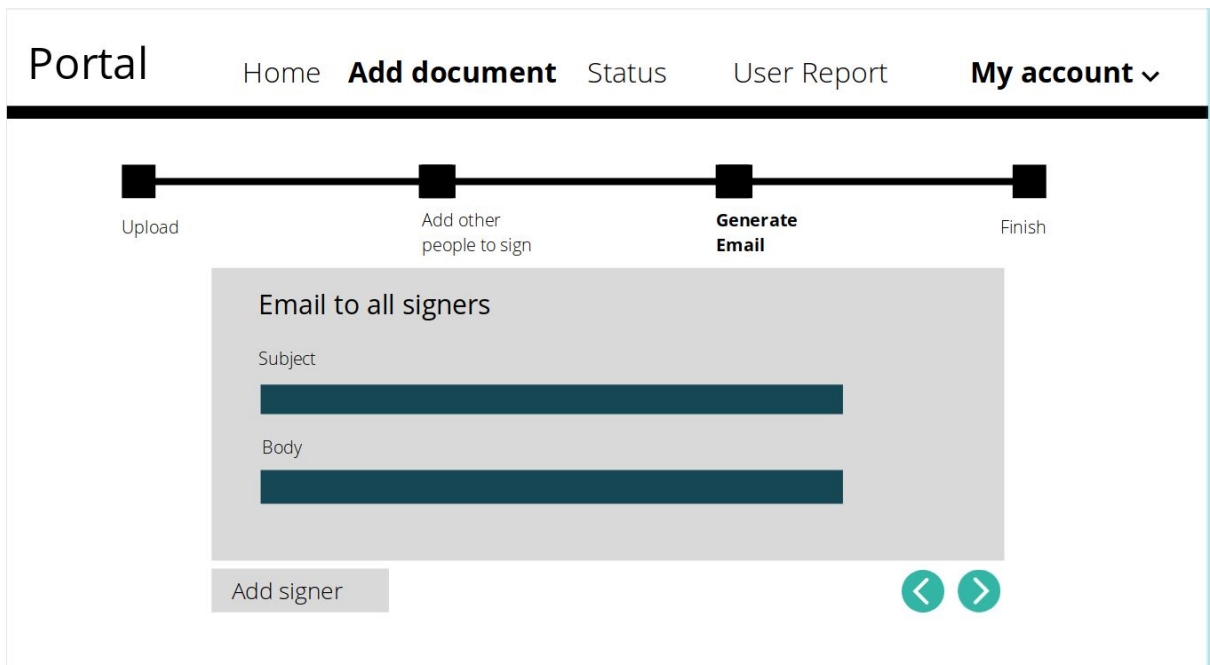
Insert from Drive

☒ I am also a signer < >

fig(9)



fig(10)¹



fig(11)

¹ An additional option to set a deadline will be also implemented which is missing at present from the design.

Chapter 4

API calls

Resource: Admin

/GET

admin/users - gets all the users in the application

admin/user/:id - gets the user with the ID id.

/POST

admin/create/user - Creates a user in the application

/DELETE

admin/delete/user/:id - deletes a user given the ID of the user

Resource: User

/GET

user/logout - Logout the user from the application

user/status - checks whether the user is logged in or not.

user/report - Gets the overview of documents of the user

user/profile - Gets the profile of the user

/POST

user/register - register with the email if the user doesn't exist

user/login - Logins with email and Password

Resource: Document

/GET

document/:id - returns a Document with the corresponding Document ID

document/all - get All documents related to the User

document/status/:id - get Status of the document (Pending, Completed) .

document/share/:id - users Shares the document with the other User .

document/requestsign/:id Request the recipients to sign the document .

document/:id/recipients - Gets the recipients of the document.

/POST

document/sign/:id - Signs the Document

document/upload - Upload a document to the server.

/DELETE

document/delete/:id - deletes the document given an id

/PUT

document/:id/add/deadline - Add a deadline to sign the document after which the document expires

document/:id/add/recipients - Add recipients of the document

document/:id/change/status - set the status of the document

document/:id/change/deadline - changes the deadline of the document.

Chapter 5

Database Schema

The database management system proposed to meet the data related requirements of the portal will be MongoDB. The following section contains the database schema² required for the portal.

User:

```
User_id      : string
Email        : string
Role         : {type:string,enum [admin,user]}
Password     : string
Public key   : string
Private Key  : string
```

Document:

```
Path          : string
Status        : {in_progress| finished | failed}
Owner         : UserId
Created_at    : Timestamp
Modified_at   : Timestamp
Recipients    : [{
    user : Object(User)
    Deadline: DateTime
    Status: {type:string , enum: [signed,rejected,
        Waiting]}
    Modified_at: Timestamp
    Comment: string
    Serial_id_for_sign: Integer }]
```

² The schema proposed is not complete and can be treated as a basic design only. We will be open to changes as we proceed further with the development phase.

Chapter 6

Conclusion

The complete design of the portal is done. Also, we have explored various technologies and have finalised the tech stack to be used for the development part. We have made a basic list of API calls and a basic design of the database schema to be used for the development of the portal. The API calls and database schema are made keeping in mind that the majority of the back-end work of the portal (signing, encryption, decryption, other cryptographic techniques, etc.) will be done in the Phase - II of the project. In Phase - I, a simple backend will be implemented which can support the front-end of the portal and hence only the relevant API calls and database schema are described in their respective chapters.

We will be implementing the portal with the above presented design towards the end of the semester.