A REPORT ON

E-BPM PORTAL

By

Names of the students	**Registration No.**
Wudaru Tejeswara Reddy	1800271C203
Gelle Abhiram	1800217C203

Prepared in the partial fulfillment of the

Practice School II Course

AT

EAIESB Software Solutions,
Bachupally, Hyderabad,
Telangana, India.

A Practice School II Station of

BML MUNJAL UNIVERSITY

July 2020

Table of Contents

CONTENTS **Page No.**

[1] Certificate of Completion ... 3

[2] Acknowledgment ... 4

[3] Objective ... 5

[4] Problem Statement ... 5

[5] Company Profile ... 6

[5.1] Innovation ... 6

[5.2] Middleware Experts ... 6

[5.3] Contact ... 6

[6] Project Methodology ... 7

[6.1] Problem ... 7

[6.2] Analysis ... 7

[6.3] Solution ... 7

[6.3.1] Back-end ... 7

[6.3.2] Front-end ... 11

[6.3.3] Control flow ... 14

[7] Data Collection ... 15

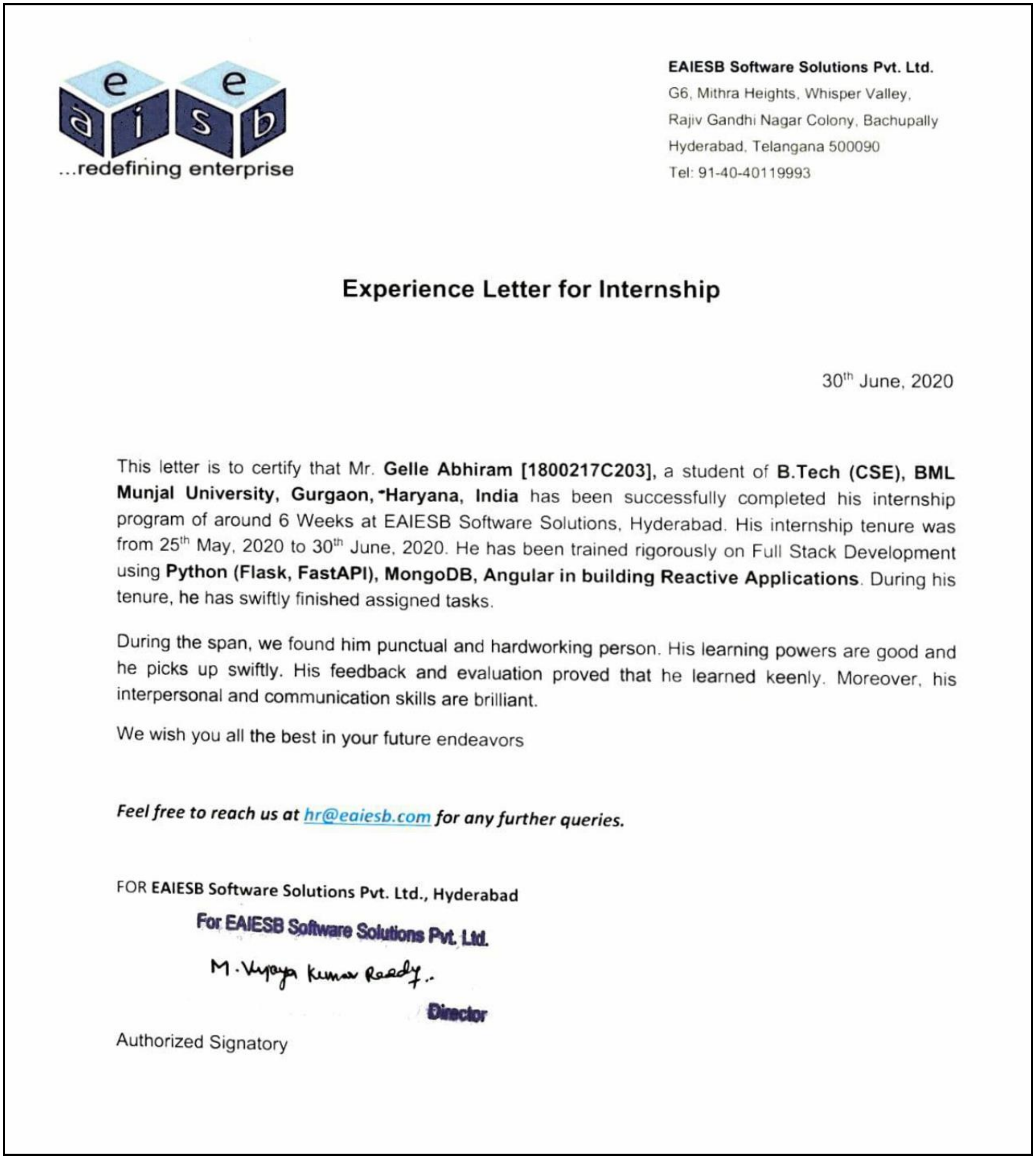
[8] Results & Discussion ... 17

[9] Conclusions ... 17

[10] Appendix 1: Tables & Figures ... 18

[11] Appendix 2: References ... 19

CERTIFICATE OF COMPLETION



...redefining enterprise

EAIESB Software Solutions Pvt. Ltd.
G6, Mithra Heights, Whisper Valley,
Rajiv Gandhi Nagar Colony, Bachupally
Hyderabad, Telangana 500090
Tel: 91-40-40119993

Experience Letter for Internship

30th June, 2020

This letter is to certify that Mr. **Gelle Abhiram [1800217C203]**, a student of **B.Tech (CSE), BML Munjal University, Gurgaon, Haryana, India** has been successfully completed his internship program of around 6 Weeks at EAIESB Software Solutions, Hyderabad. His internship tenure was from 25th May, 2020 to 30th June, 2020. He has been trained rigorously on Full Stack Development using **Python (Flask, FastAPI), MongoDB, Angular in building Reactive Applications**. During his tenure, he has swiftly finished assigned tasks.

During the span, we found him punctual and hardworking person. His learning powers are good and he picks up swiftly. His feedback and evaluation proved that he learned keenly. Moreover, his interpersonal and communication skills are brilliant.

We wish you all the best in your future endeavors

Feel free to reach us at hr@eaiesb.com for any further queries.

FOR **EAIESB Software Solutions Pvt. Ltd., Hyderabad**

For EAIESB Software Solutions Pvt. Ltd.
M. Vijaya Kumar Reddy..
Director

Authorized Signatory

ACKNOWLEDGMENT

It is indeed a great pleasure for me to present this PS-2 training report on e-BPM Portal in EAIESB Software Solutions Pvt. Ltd. as a part of the curriculum of the Bachelor of Technology (Computer Science and Engineering) degree.

I am very grateful to express my thanksgiving and respect to my instructors: Vijaya Kumar Reddy Maddela, Akhil K, Vamshi Krishna & G Manoj Kumar Reddy for helping and encouraging us to complete our Internship program. I would also like to assert my special thanks of gratefulness to Professor Manoj K. Arora, Vice Chancellor and Professor Maneek Kumar, Dean, SOET and Dr. Pradeep Arya, coordinator of PS-II programme who provided me the valuable opportunity to perform this PS-2 Internship as part of the plan in the SOET academic year and also to Dr. Purnendu Shekhar Pandey who guided me in this program as Faculty Mentor of our University. This program additionally helped me in doing a lot of exploration and came to think about many new things. Furthermore, I might likewise want to thank my companions who helped me completing this project in limited span of time.

Signature of Student
Gelle Abhiram

Objective:

The main objective of **e-BPM Portal** is to optimize business processes to make them more efficient and also to ensure that they are robust enough so that tasks do not "fall through the cracks".

Often business processes have been in place for many years, and have been set up before the people who currently undertake these processes even worked at the organization. In large organizations, these processes can be affected by inertia and are very hard to change. In these cases, large projects are often undertaken to modernize these processes taking new technology and best practices into account.

It helps organizations to identify all the processes in their organization, capture them, modelling change to analyses the likelihood of success and implementing change.

The results will often automate tedious tasks and free employees from paperwork in order that they may focus more on special cases and exception management.

A company can have thousands of processes and making them all run more efficiently could save a business a good amount of time and money.

Problem Statement:

Let’s say our team is creating more reports/tasks than ever and you don’t have a clear idea of the status of different projects. You know all information is available but buried somewhere in corporate email and spreadsheets. It’s easy to say for being more patient and searching works of individuals but practically it’s more difficult because there will be no limitation or fixed value for choosing employees or workers in any enterprise. So, when time comes into looking works/tasks created of individuals or tasks assigned to different groups, it will be more complex situation for us to separate parts. It’s even more difficult when you are a manager as you also need to look into your reportees jobs.

Manual Intervention for keeping processes/tasks also does not help when there are a huge number of working agents/employees, that is when you think there should be a better way of doing this – Then BPM Portal (Business Process Management Portal) comes up, in no particular order. There begins your search for a software like BPM to automate your business processes. You are probably looking for a system not very complicated but something that still brings your job easy rather than chaos whirling all around.

Company Profile:

EAIESB is an innovative and award-winning IT Professional Services which delivers enterprise software solutions to Automate and Optimize Complex Business Processes from Full-Scale implementations to Ongoing Support.

Industries:	Computer Software
Company size :	11-50 employees
Headquarters:	Hyderabad, Andhra Pradesh
Type:	Partnership
Founded:	2008
Specialties:	Oracle BPM, Oracle SOA, JCAPS, Java CAPS, Oracle Fusion, SOA, B2B, MDM, A2A, jUDDI, IDM, ODI, ADF, MAF, TIBCO Business Works, TIBCO Business Connect, TIBCO SPOTFIRE, TIBCO BPM, TIBCO BUSINESS EVENTS, API, EDI, APIGEE, WSO2, Azure API, DEVOPS, AWS, GCP, Azure, Dell Boomi, KONG, Docker, OpenShift, MuleSoft, Kubernetes

Innovation:

As trusted sources who move and inspire people with innovative ideas, EAIESB turns these ideas into reality. Our innovative implementation won the award from Oracle.

Middleware Experts:

EAIESB has proven expertise and project experience in middleware tools like MuleSoft, Oracle Fusion & TIBCO stacks. Migration Experts from SeeBeyond, Sun products, Oracle Fusion, TIBCO, and MuleSoft vice versa.

Contact:

- **Phone:** +91 (040) 4011-9993
- **Email:** info@eaiesb.com
- **Website:** https://eaiesb.com
- **Address:** G6, Mithra Heights Whisper Valley,
  Opposite Rajiv Gandhi Nagar Colony,
  Bachupally, Hyderabad – 500090,
  Telangana, India.

Project Methodology:

Problem:

As we discussed in the problem statement, there rises an issue of mismatching or becomes chaos when needed to find data of a large group of people's work or individual’s work. So, in those situations we need to find any access such that people’s tasks and jobs are assigned separately and data of one individual will have information with themselves.

Analysis:

Logic of sending work packages will be better which means that the data for the work to be told are routed from one working place to the next with the support of a system, so that manual intervention can be reduced to a minimum and also there is no need of remembering the tasks of everyone or no need to worry about the submission of tasks on particular margins since the cluster of whole lots of dates need not to be remembered since data is stored and visible at any time and serves best comfort for both manager/head or even employees of any enterprise.

Solution:

Based on the analysis’s thought, we built an application called e-BPM Portal. This application is developed in two layers i.e. front-end and back-end.

As we are working in a group on this project, we decided to work individually. So, myself **Gelle Abhiram**, worked on back-end part and my teammate Wudaru Tejeswara Reddy worked on front-end part of this project.

To make you understand how we designed our web application in both front and back end. We explained how the whole application works in detailed way below.

Back-end:

In the backend part, we used FastAPI (Python) and MongoDB for managing and storing data. We built several CRUD (create, read, update, delete) operations to create and manage data. These CRUD operations are named differently in REST services context. They are defined as

GET (read), POST (create), PUT (update), DELETE (delete)

For creating users, groups and managing their tasks requires models/tables that stores data with certain attributes. We created four models/tables to store different data. They are:

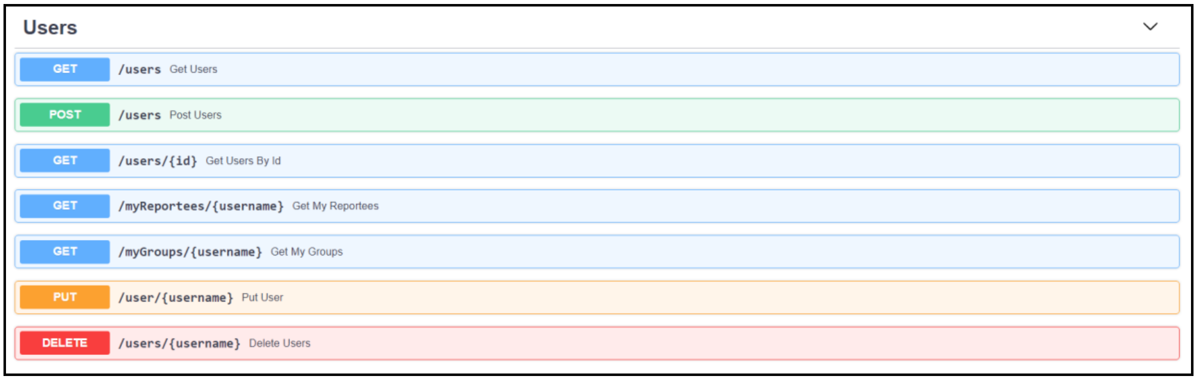
- ## Users:

Users is collection of user details to be stored in database.

Attributes in users:

username, password, firstName, lastName, dateOfBirth, status, manager, role, pan, aadhar, passport, displayPicture, joiningDate, releasingDate

CRUD operations:



Users

GET /users Get Users

POST /users Post Users

GET /users/{id} Get Users By Id

GET /myReportees/{username} Get My Reportees

GET /myGroups/{username} Get My Groups

PUT /user/{username} Put User

DELETE /users/{username} Delete Users

Fig 1.1: *FastAPI user-related services*

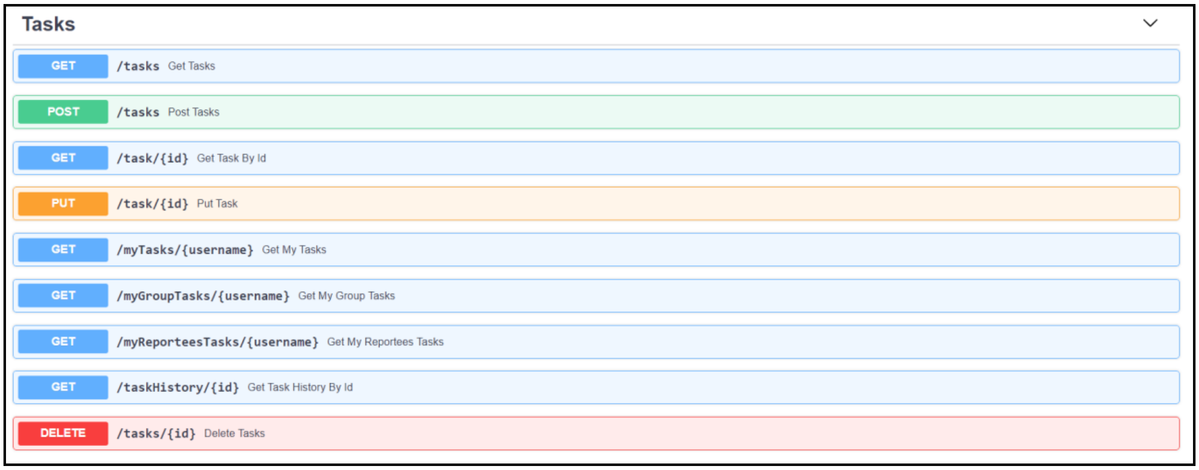
- ## Tasks:

Tasks is collection of task details of a user/group.

Attributes in Tasks:

taskTitle, taskDescription, payload, currentOwner, createdBy, assignedType, assignedTo, status, updatedBy, createdTime, updatedTime

CRUD operations:



Tasks

GET /tasks Get Tasks

POST /tasks Post Tasks

GET /task/{id} Get Task By Id

PUT /task/{id} Put Task

GET /myTasks/{username} Get My Tasks

GET /myGroupTasks/{username} Get My Group Tasks

GET /myReporteesTasks/{username} Get My Reportees Tasks

GET /taskHistory/{id} Get Task History By Id

DELETE /tasks/{id} Delete Tasks

Fig 1.2: *FastAPI task-related services*

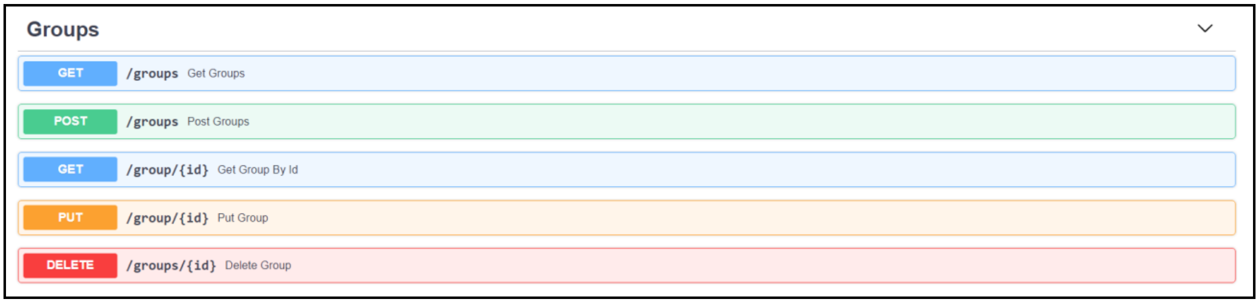
- ## Groups:

Groups is a collection of group name and group description of the company.

Attributes in Groups:

groupName, groupDescription, createdTime, updatedTime

CRUD operations:



Groups

GET /groups Get Groups

POST /groups Post Groups

GET /group/{id} Get Group By Id

PUT /group/{id} Put Group

DELETE /groups/{id} Delete Group

Fig 1.3: *FastAPI group-related services*

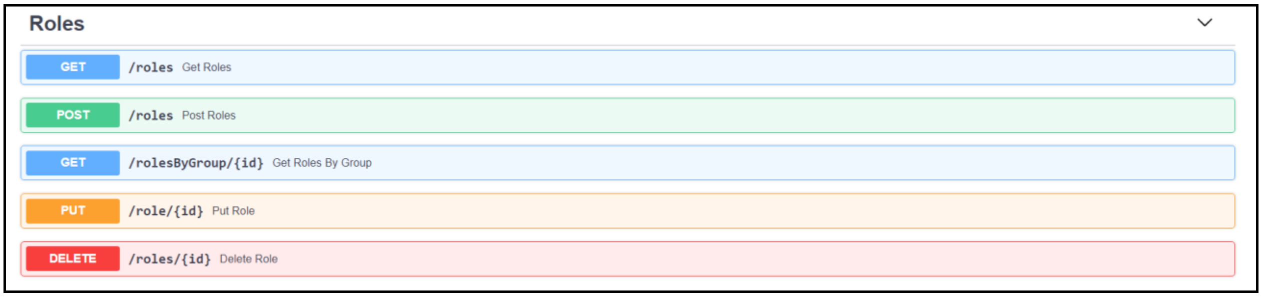
- ## Roles:

Roles is a collection that shows which user is present in which group.

Attributes in Roles:

groupName, username, createdTime, updatedTime

CRUD operations:



Roles

GET /roles Get Roles

POST /roles Post Roles

GET /rolesByGroup/{id} Get Roles By Group

PUT /role/{id} Put Role

DELETE /roles/{id} Delete Role

Fig 1.4: *FastAPI role-related services*

Task History:

Task History is a collection of logs that occur when a task is updated.

Attributes in Task History:

taskId, beforeChanges, afterChanges, updatedBy, updatedTime, comment

CRUD operations:

Task history will be created/updated when the task is updated automatically.

One method is listed under Tasks that uses Task History model i.e. ‘Get Task History by Id’

Front-end:

We made a user interface using angular in a modern way that users are familiar to operate i.e. understandability and readability so that they can understand where to go and what to choose in the application. Some of our UI faces are presented below.

- ## Login/Sign-Up Page:

  Our login/sign-up page have simple and modern look and the background image will be changing automatically in a small interval of time.

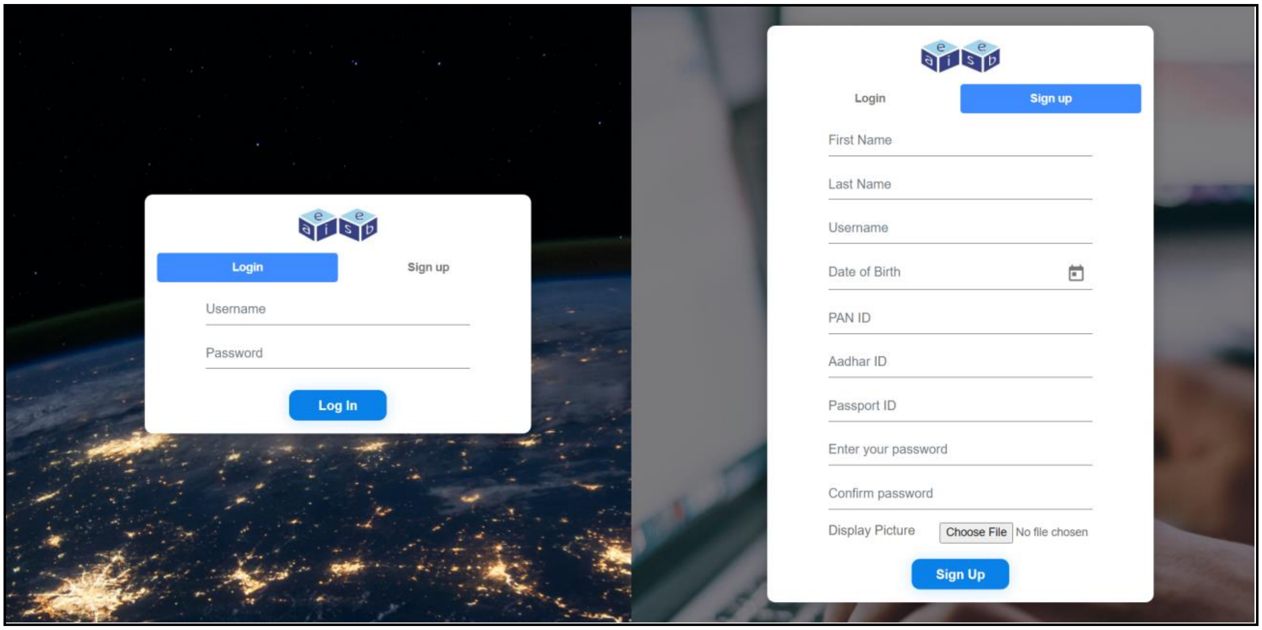
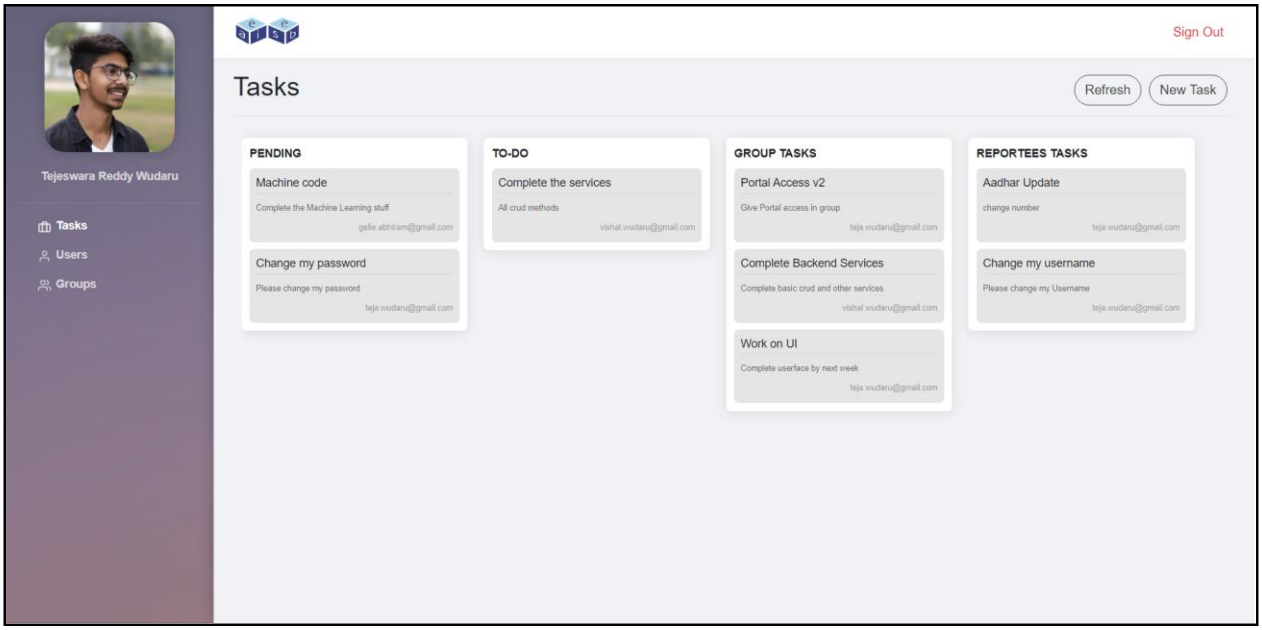


Fig 2.1: *Login/Sign-up Page*

- ## Tasks Page:

  After logging in, the user will be redirected to Tasks page.

  In Tasks, all activities that need to be performed by him/her and the activities that need to be done are displayed under waiting and to-do card. All the tasks that are assigned to the logged-in user groups under group tasks and if user is a manger of some employees, their tasks will be listed under reportees tasks.



Tasks

PENDING

Machine code
Complete the Machine Learning stuff
gelle.abhiram@gmail.com

Change my password
Please change my password
teja.wudaru@gmail.com

TO-DO

Complete the services
All crud methods
vishal.wudaru@gmail.com

GROUP TASKS

Portal Access v2
Give Portal access in group
teja.wudaru@gmail.com

Complete Backend Services
Complete basic crud and other services
vishal.wudaru@gmail.com

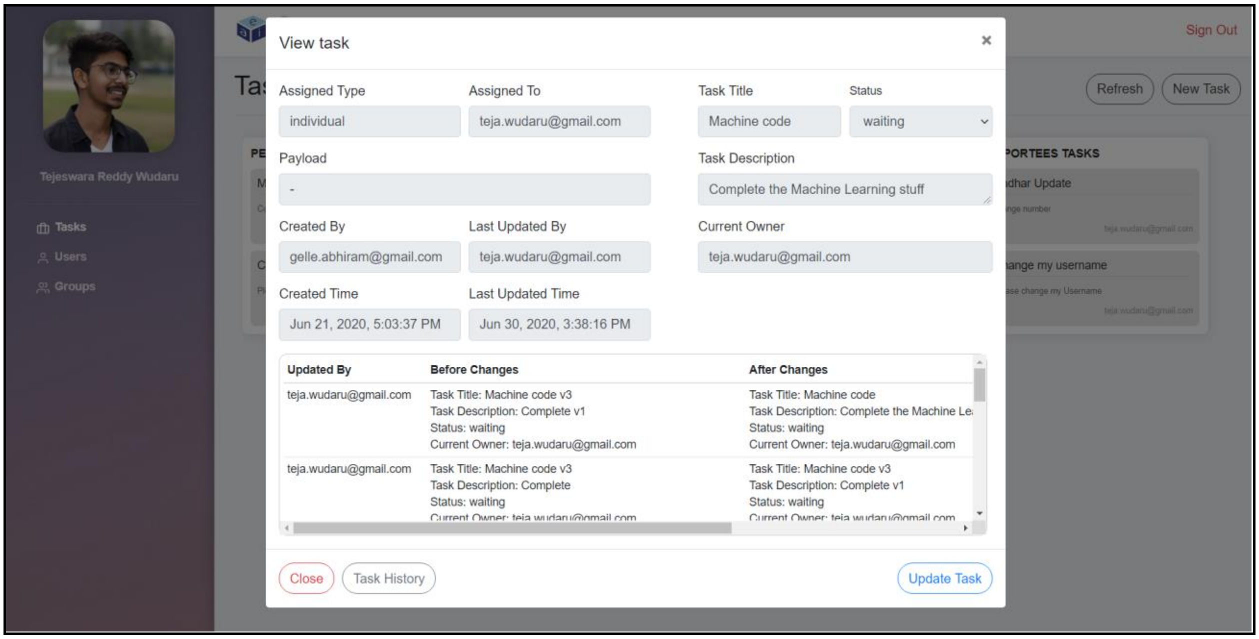
Work on UI
Complete userface by next week
teja.wudaru@gmail.com

REPORTEES TASKS

Aadhar Update
change number
teja.wudaru@gmail.com

Change my username
Please change my Username
teja.wudaru@gmail.com

Fig 2.2.1: *Tasks page*



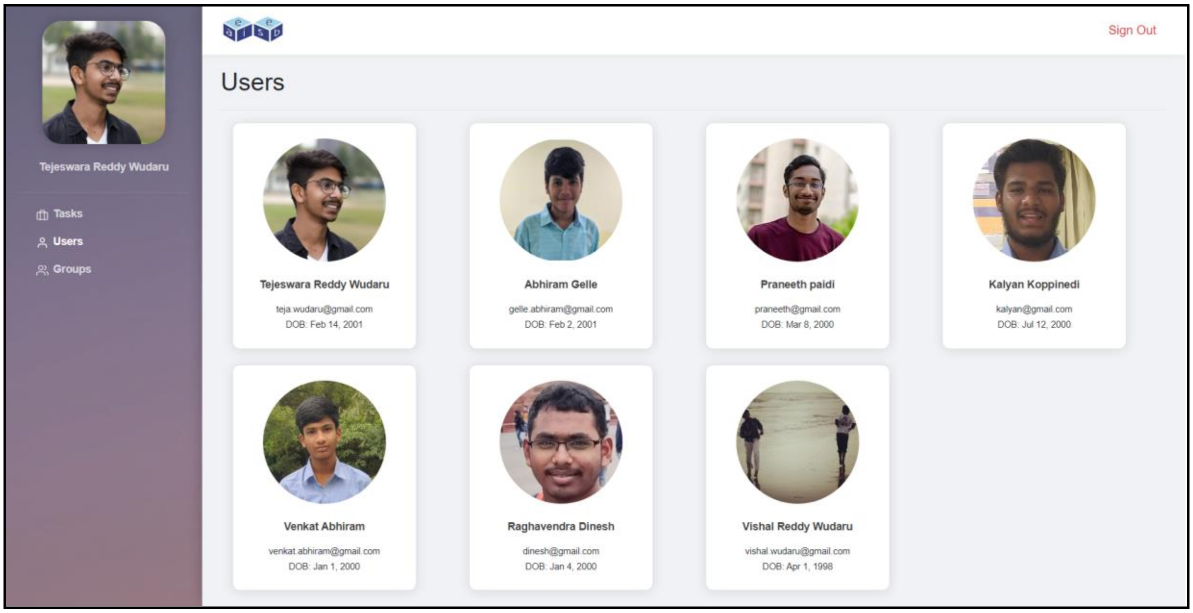
View task

Assigned Type: individual
Assigned To: teja.wudaru@gmail.com
Task Title: Machine code
Status: waiting
Payload: -
Task Description: Complete the Machine Learning stuff
Created By: gelle.abhiram@gmail.com
Last Updated By: teja.wudaru@gmail.com
Current Owner: teja.wudaru@gmail.com
Created Time: Jun 21, 2020, 5:03:37 PM
Last Updated Time: Jun 30, 2020, 3:38:16 PM

Updated By	Before Changes	After Changes
teja.wudaru@gmail.com	Task Title: Machine code v3 Task Description: Complete v1 Status: waiting Current Owner: teja.wudaru@gmail.com	Task Title: Machine code Task Description: Complete the Machine Le Status: waiting Current Owner: teja.wudaru@gmail.com
teja.wudaru@gmail.com	Task Title: Machine code v3 Task Description: Complete Status: waiting Current Owner: teja.wudaru@gmail.com	Task Title: Machine code v3 Task Description: Complete v1 Status: waiting Current Owner: teja.wudaru@gmail.com

Fig 2.2.2: *View task pop-up*

- **Users Page:**
  This page displays all the users in company with their name and date of birth.



Users

Tejeswara Reddy Wudaru
teja.wudaru@gmail.com
DOB: Feb 14, 2001

Abhiram Gelle
gelle.abhiram@gmail.com
DOB: Feb 2, 2001

Praneeth paidi
praneeth@gmail.com
DOB: Mar 8, 2000

Kalyan Koppinedi
kalyan@gmail.com
DOB: Jul 12, 2000

Venkat Abhiram
venkat.abhiram@gmail.com
DOB: Jan 1, 2000

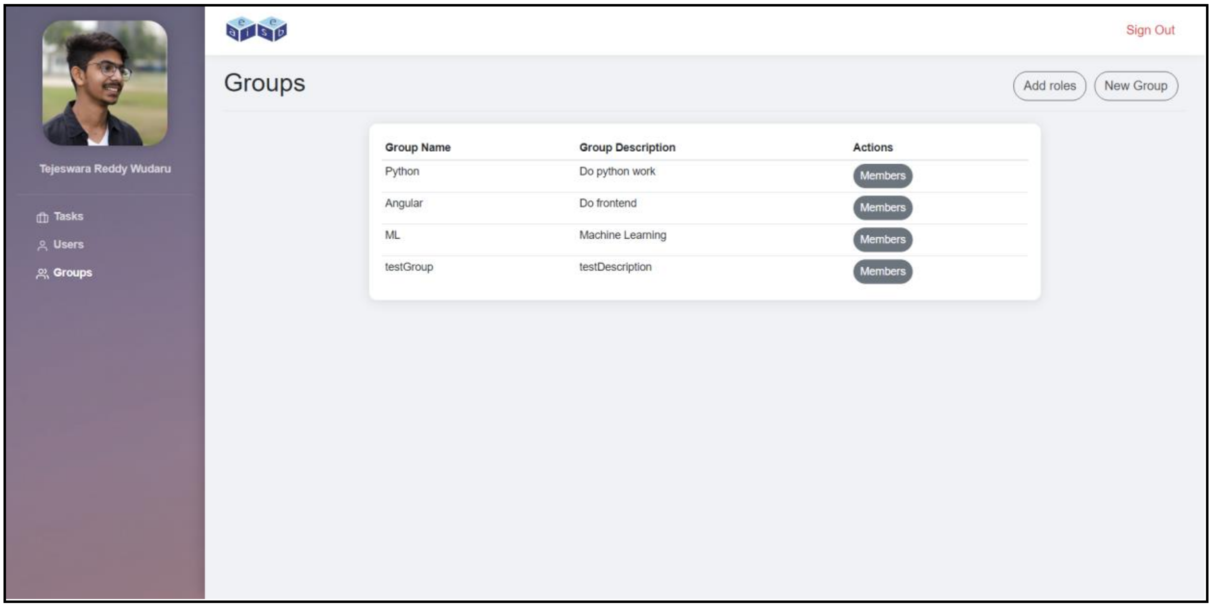
Raghavendra Dinesh
dinesh@gmail.com
DOB: Jan 4, 2000

Vishal Reddy Wudaru
vishal.wudaru@gmail.com
DOB: Apr 1, 1998

Fig 2.3: *Users page*

- **Groups Page:**
  This page lists the groups and its members. It has two versions: for user and admin.

  Admin: An admin can see all the groups, add groups and members for a group.

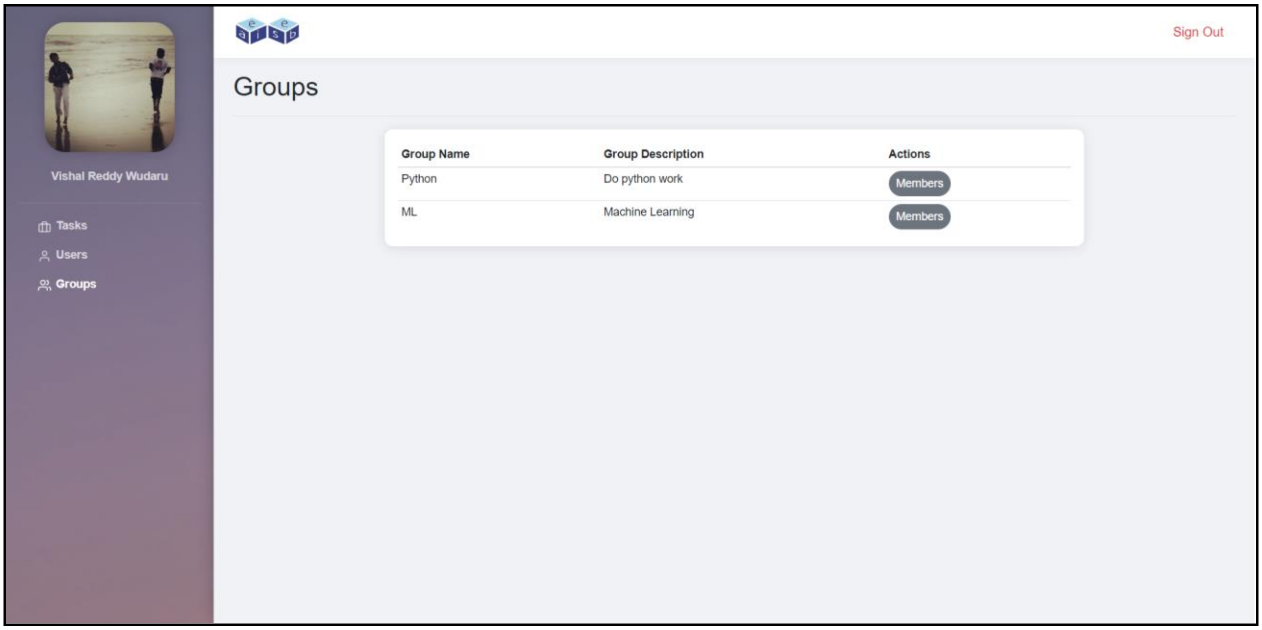


Groups

Group Name	Group Description	Actions
Python	Do python work	Members
Angular	Do frontend	Members
ML	Machine Learning	Members
testGroup	testDescription	Members

Fig 2.4.1: *Groups page for admin*

User: A user can only see his groups and their members



Groups

Group Name	Group Description	Actions
Python	Do python work	Members
ML	Machine Learning	Members

Fig 2.4.2: *Groups page for normal user*

Control Flow:

After knowing about the front-end and back-end of our project, we need know how the two layers work together to respond for user's input/request. Let's break it down into steps to understand it better.

I. Firstly, when a user performs an action like clicking a button it creates a request/service call to the backend FastAPI services.

II. Based on the request, the service that gets called, it will fetch data from the database accordingly.

III. Database returns the data in the form the service requests from it.

IV. After receiving the data, the service processes the data and sends response to front-end it may be either a message or data that is requested.

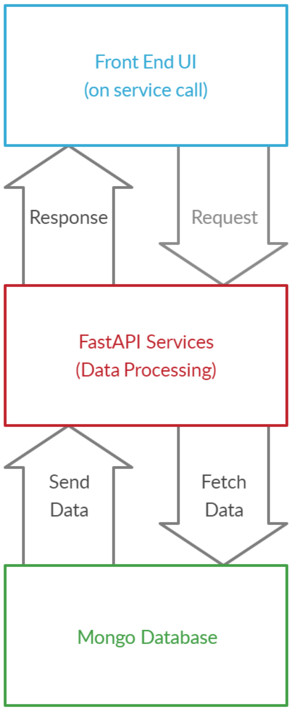
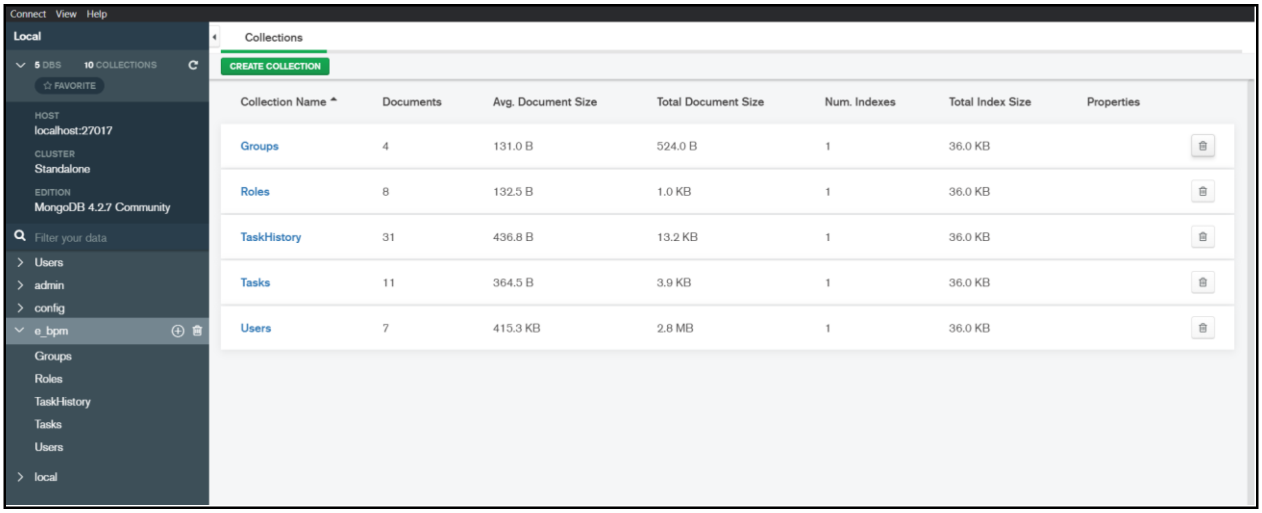


Fig 3: *Control flow of the application*

Data Collection:

We used MongoDB for storing/collecting data. We created a database and created collections like Users, Roles, Groups, Tasks, Task History for storing information.



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 4.2.7 Community

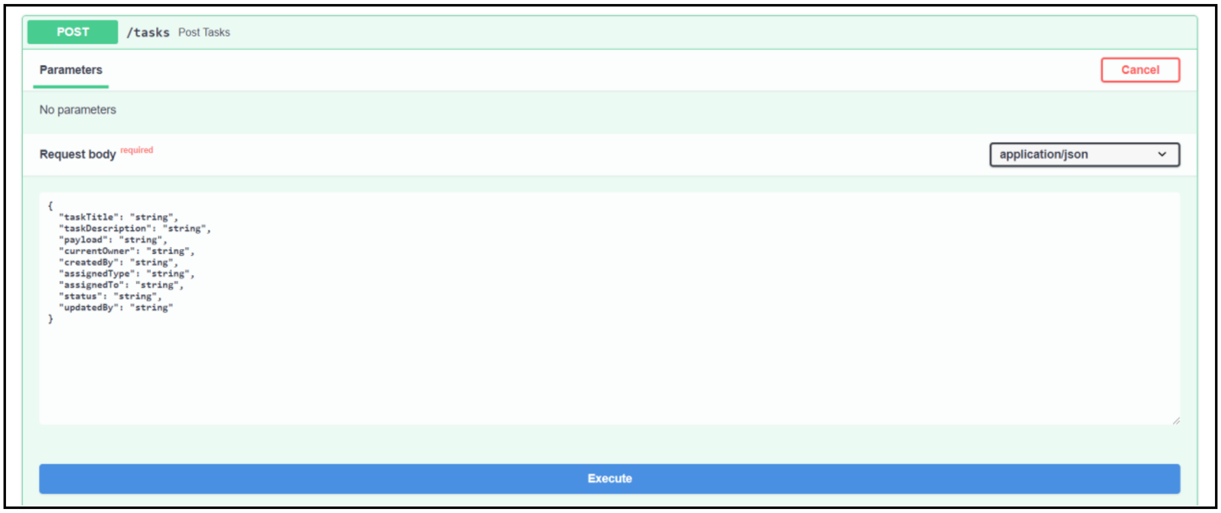
Collection Name ▲	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
Groups	4	131.0 B	524.0 B	1	36.0 KB	
Roles	8	132.5 B	1.0 KB	1	36.0 KB	
TaskHistory	31	436.8 B	13.2 KB	1	36.0 KB	
Tasks	11	364.5 B	3.9 KB	1	36.0 KB	
Users	7	415.3 KB	2.8 MB	1	36.0 KB	

Fig 4.1: *MongoDB - Collections*

Creating and retrieving data is processed by the python service s (CRUD methods) that we coded using FastAPI as a part of backend. We can either create and retrieve data from Swagger UI (that lists all the CRUD methods) or from UI that we created which is internally connected to the same python services.

For example, for creating task:

1) Swagger UI



```
{
  "taskTitle": "string",
  "taskDescription": "string",
  "payload": "string",
  "currentOwner": "string",
  "createdBy": "string",
  "assignedType": "string",
  "assignedTo": "string",
  "status": "string",
  "updatedBy": "string"
}
```

Fig 4.2: *Create Task – Swagger UI*

2) UI application

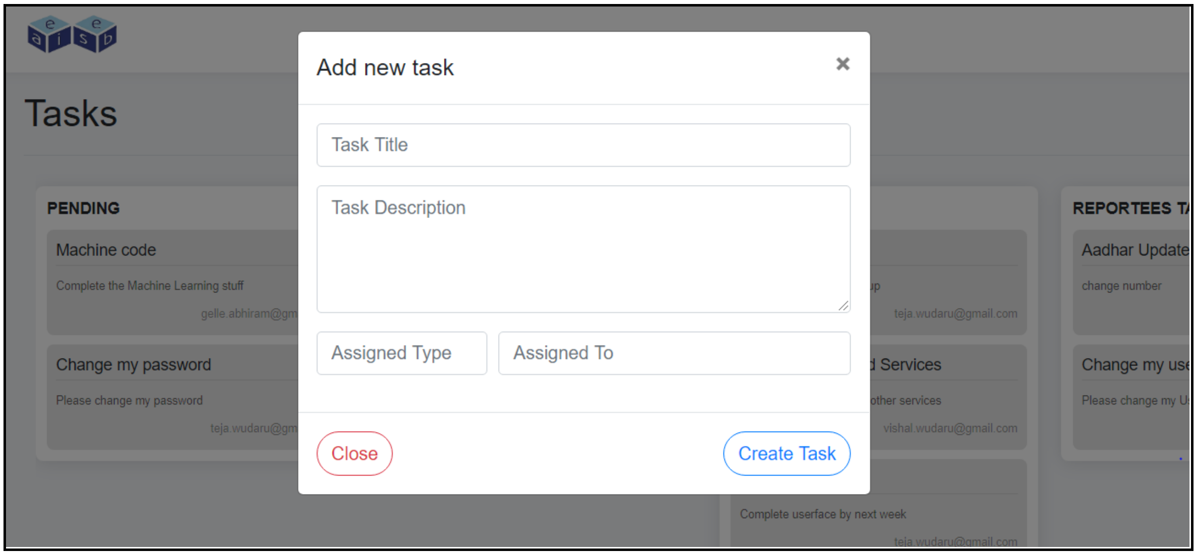


Fig 4.3: *Create task – Application UI*

Results and Discussion:

We are able to develop the whole functioning e-BPM Portal web application that overcomes all the problems that employees facing for organizing their tasks. In a detailed way:

- Building a user interface using angular that is easy usable for users. A user can be able to check their tasks, group tasks and their reportees tasks if he is a manger and can create tasks and assign to someone else. Groups can be created and add members to them by an admin.
- Whereas in back-end, to retrieve data from the database many services (create, read, write, update) are created so that the data is processed accordingly. All the passwords are encrypted on high level so that it can cannot exposed. These services are built using FastAPI (Python) that can retrieve data as fast as possible.

Conclusion:

BPM-Portal (Business Process Management) is a must-use application for companies because it allows the company to make the flow of work to be efficient and also obtaining the chances that involve changes in the environment. Company that is having a well-coordinated/well-planned management ensures the work of flow smooth and clear objectives so that it is then likely to have greater process management. The thought that implied/learned from the above which is mentioned is that it is best to make sure that the flow of work is made smooth and flexible in the workplace and making sure that the company is enhanced for a good well-coordinated/well-planned process management of our enterprise/business.

APPENDIX-1
List of Figures

Fig 1.1 FastAPI user-related services.......8

Fig 1.2 FastAPI task-related services.......8

Fig 1.3 FastAPI group-related services.......9

Fig 1.4 FastAPI row-related services.......9

Fig 2.1 Login/Sign-up Page.......11

Fig 2.2.1 Tasks Page.......12

Fig 2.2.2 View task pop-up.......12

Fig 2.3 Users page.......13

Fig 2.4.1 Groups page for admin.......13

Fig 2.4.2 Groups page for normal user.......14

Fig 3 Control flow of the application.......15

Fig 4.1 MongoDB – Collections.......15

Fig 4.2 Create Task – Swagger UI.......16

Fig 4.3 Create Task – Application UI.......16

APPENDIX-2
References

FastAPI Documentation	https://fastapi.tiangolo.com/
Angular Documentation	https://angular.io/docs
MongoDB Documentation	https://docs.mongodb.com/manual/
BPM by EAIESB team	https://www.youtube.com/watch?v=JNn0ecS_xUk
Refer my project	https://github.com/Abhiramgelle/PS--Project_eBPM-Portal