REAL-TIME DYNAMIC WEBSITE USING MERN STACK

TEAM MEMBERS:

Abhijeeth Baregal (C00220009), Shivanjali Khare (C00221719),

Padmaja Priyanka Kasaraneni (C00220794)

Supervised by

Dr. Henry Chu

Contents

Bitbucket URL

https://bitbucket.org/abhimike17/frontend-590
https://bitbucket.org/abhimike17/backend-590

INTRODUCTION

The project aims to develop a web application for a NGO. The application will provide all information about the NGO's current events, projects, on-going research. The website will also provide the facility to the interested users to apply for open-positions/job opportunities available within the company.

TECHNOLOGIES USED:

Platform: Windows

Middleware: Express

Runtime environment: Nodejs

Library: ReactJS

Database: MongoDB

MONGODB

MongoDB is a free and open-source cross-platform document-oriented database program. MongoDB uses JSON-like documents with schemas.

EXPRESS

A web application framework for Node.js. It is designed for building web applications and APIs. It is in fact the standard server framework for Node.js.

REACTJS

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. React can be used in the development of single-page applications and mobile applications. It aims primarily to provide speed, simplicity, and scalability. As a user interface library, React is often used in conjunction with other libraries such as Redux.

NODEJS

Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionality. Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams, and other core functions. Node.js modules use an API designed to reduce the complexity of writing server applications. Node.js applications can run on Linux, macOS, Microsoft Windows, Nonstop and Unix servers.

Main Features:

➔ Ad hoc queries

➔ Indexing

➔ Replication

➔ Load balancing

➔ File storage

➔ Aggregation

➔ Server-side JavaScript execution

➔ Capped collections

APPLICATION DEVELOPMENT

The objective of the application is to develop in agile methodology. The promise of agile methodology is to make software development faster, at the same time, have higher code quality and higher customer satisfaction.

The term Agile Software Development is a container that includes all the methodologies that share the values and principles stated on the Agile Manifesto. The main characteristic of the methodology is that they are people-centric—i.e., they emphasise teamwork, face-to-face communication, and short feedback-loops.

MILESTONES

- **Sprint 1** (To know the requirements of the customer and plan)
- During Sprint 1,user requirements were collected. The scope of the application was determined and modules were prioritized. Based on this information, technical and economic feasibility was assessed to decide which modules are worth pursuing.
    - **Customer profile [admin/owner of company]**
        - Admin
            - Profile
        - Privileged user managing the entire application.
            - Services
                - Grant privileges to the user.
                - Post events.
                - Update the information about the events.
                    - Name
                    - Description
                    - Date
                    - Time
                    - Pictures

- **User profile**
  - Users/Job Seekers
    - Profile
      - Job Seekers looking for career opportunities with the company.
    - Services
      - View all the jobs
      - Apply for any jobs posted
  - Recruiter
    - Profile
      - User with recruiting rights approved by the admin.
    - Services
      - Monitor job applications submitted.
      - Notify the job status.
      - Conduct interview.
      - Select the appropriate candidates for the job

USER STORIES

[Home Page UI] - Develop home page user interface with clear navigation facility for the entire application.

[About Us UI] - User interface with information about NGO and their projects.

[Support Us UI] - Develop User interface to list about donations to the NGO.

[Contact Us UI] - User interface with information to get in touch with the NGO.

[Post a job] - As a recruiter user, I need to post job details for recruiting appropriate candidates.

[Edit a job] - As a recruiter user, I need to edit the information of the posted job.

[Check views] - As a recruiter user, I need to check the number of views for the job.

[Review Profile] - As a recruiter user, I need to check the profile of the applicant who applied for the posted position.

[Notify job seeker] - As a recruiter user, I need to inform the job seeker about the status of the application and decision made during the interview.

[Post an event] - As an admin user, I need to post event.

[Modify an event details] - As an admin user, I need to edit/delete event details.

[Give access to recruiter] - As an admin user, I need to approve/decline recruiter signup request.

[Post details about research] - As an admin user, i need to post the details about the current/past researches in the company.

[Post details about project] - As an admin user, I need to post details about projects in the company.

[Create profile] - As a job seeker, I need to create profile to apply for jobs.

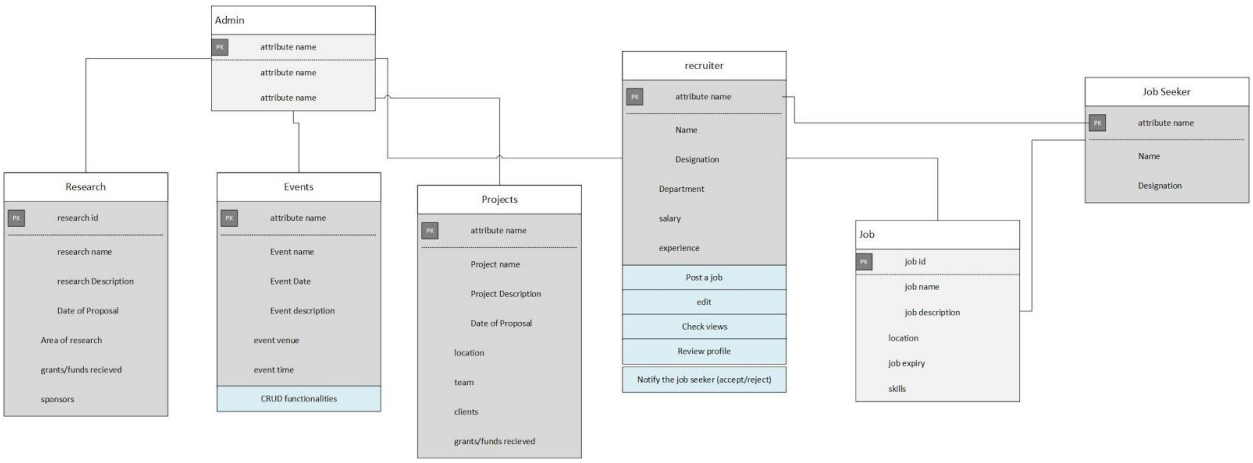
[Find jobs] - As a job seeker, I need to look for open job positions in the company and apply to the matching position.

[General Information] - As a general user, I need to get the information about the company from the website- contact details, projects etc.

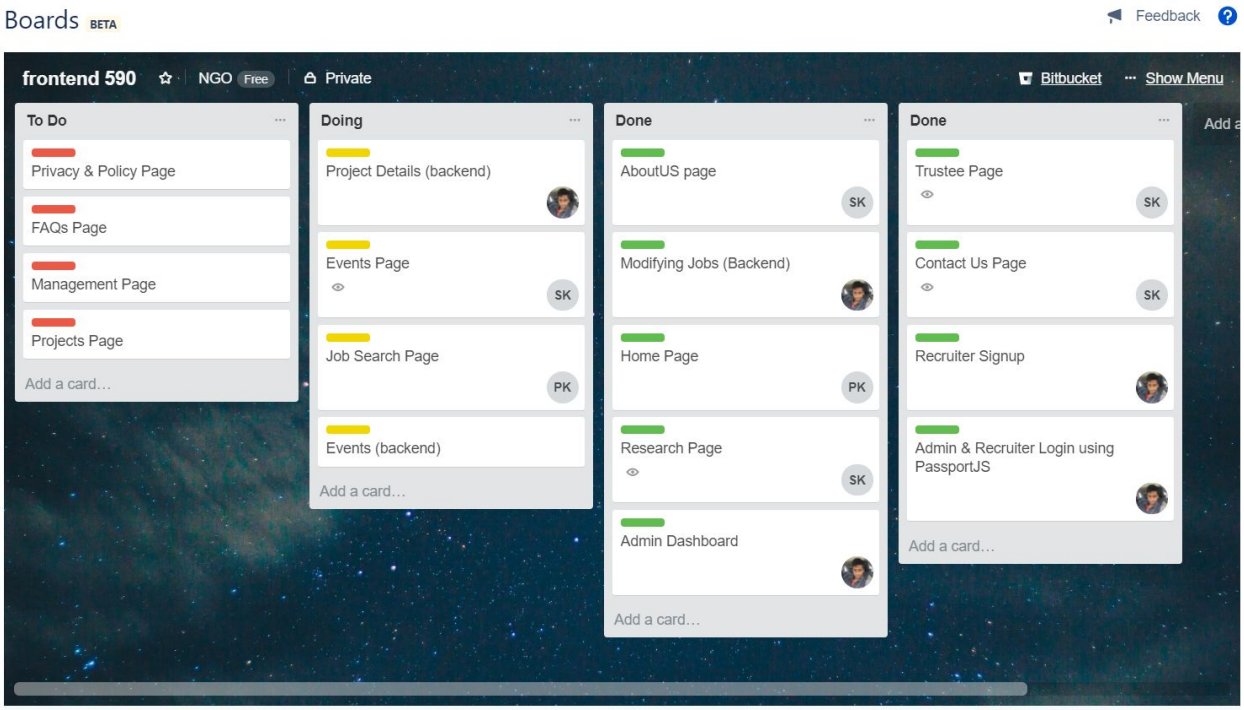
- **Sprint 2** (Create db table, start developing pages ~ aboutus, research page, contact us, home page, trustees page, backend-admin login authentication, admin signup up form)

Here, the high - level UML diagram is used to demonstrate how the feature should function and how it will fit into the system.

UML-Diagram



Trello Board



- **Sprint 3** (Create terms & conditions page, privacy policy, FAQs page, management page, backend- forgot password functionality, email-verification, job posting with CRUD functionality). In this sprint, the key functionalities required to make the working product is focussed on.

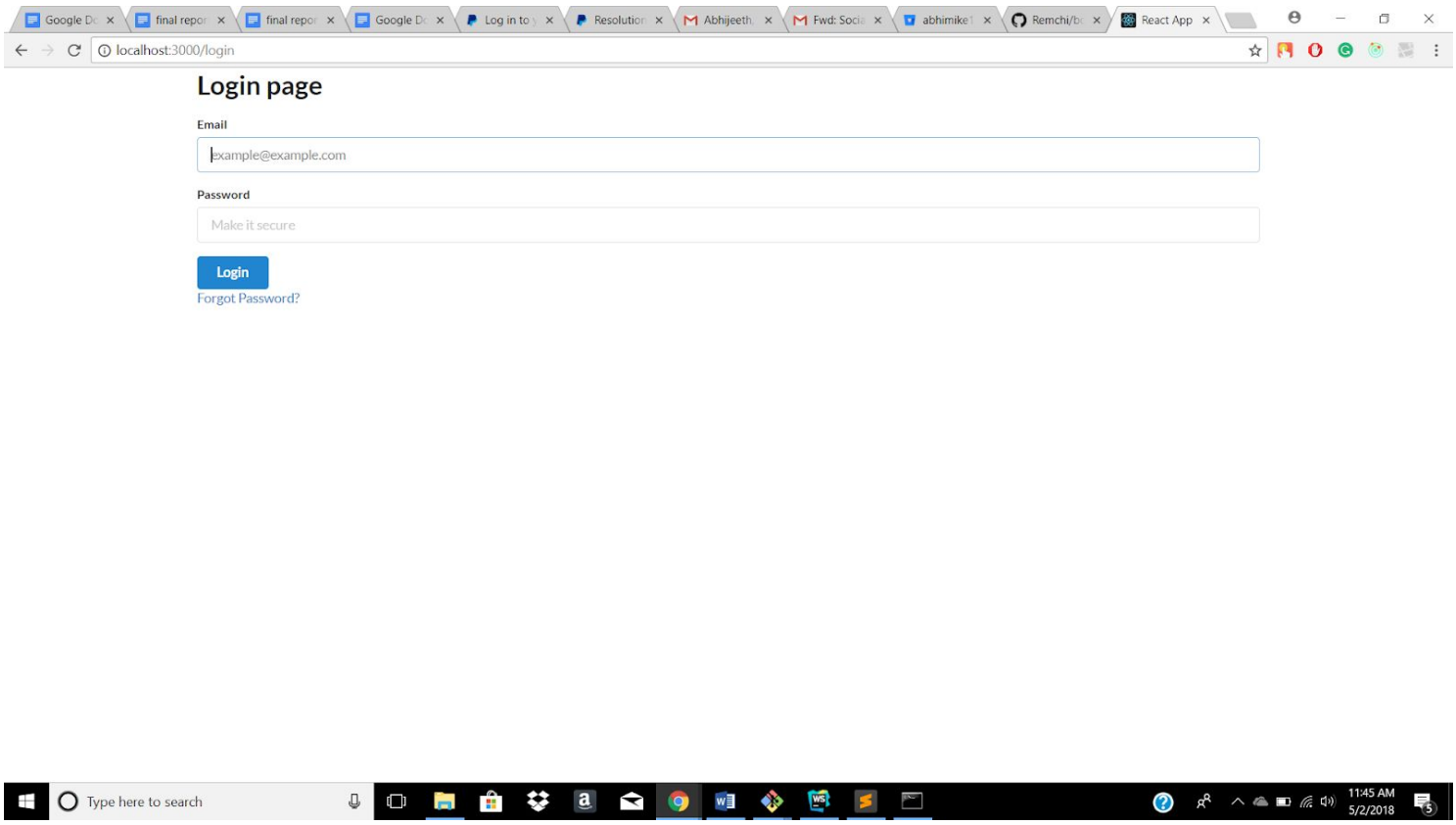
- **Sprint 4** (Improve the design and layout of all pages using semantic UI making interface attractive and easy to use)

  To present the working software to stakeholders and customers, ongoing support for the software release is involved here. During this phase, focus is completely on keeping the system run smoothly and show users how to use it.

Detailed description of Pages and their functionalities

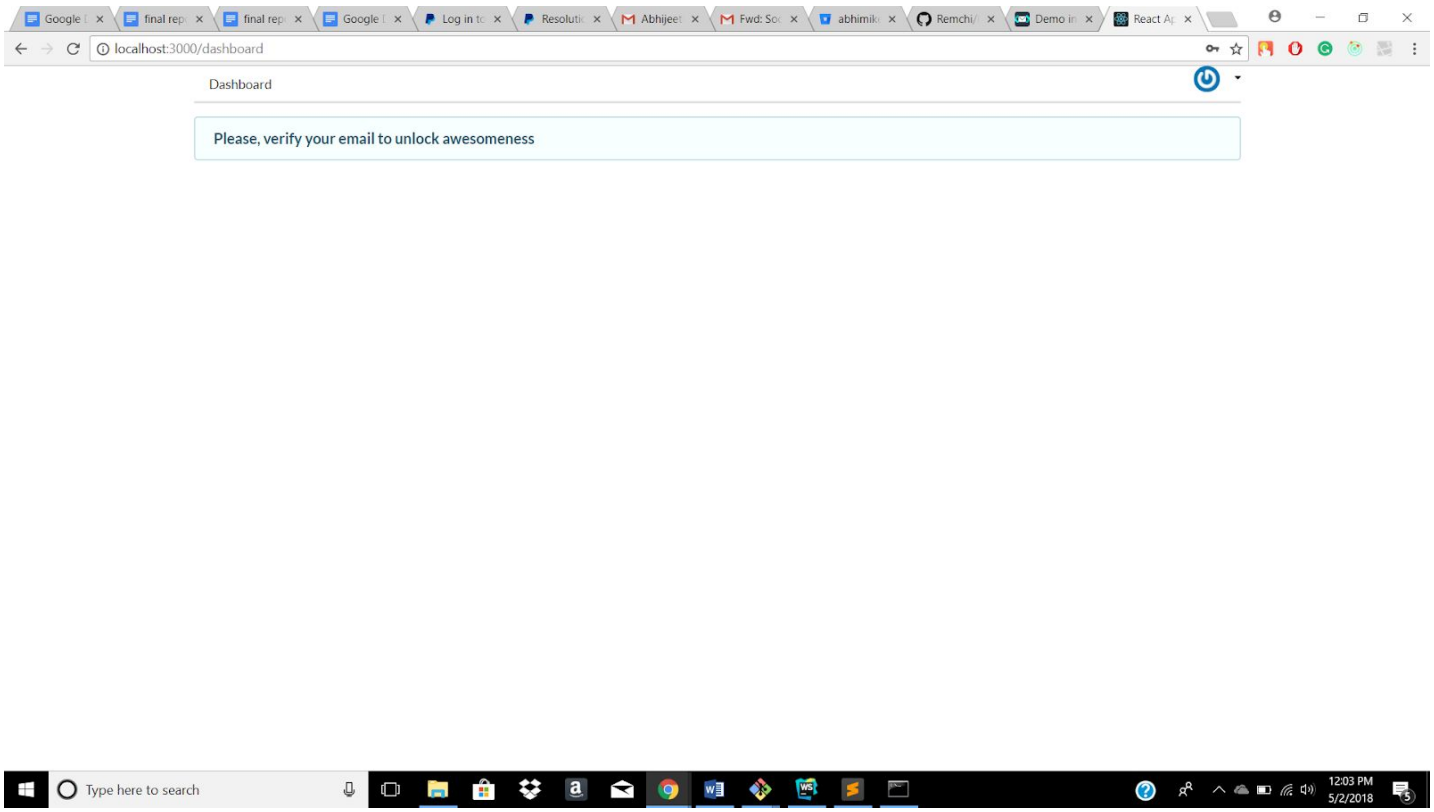
Signup & Login Page

This page includes the User Authentication which is done using bcrypt hashing , json web token. This facility is only available for admin by using http:localhost:3000/login or http:localhost:3000/signup. The user has to fill the signup form which will be stored in the database and later move to login page, enter credentials to move to Dashboard



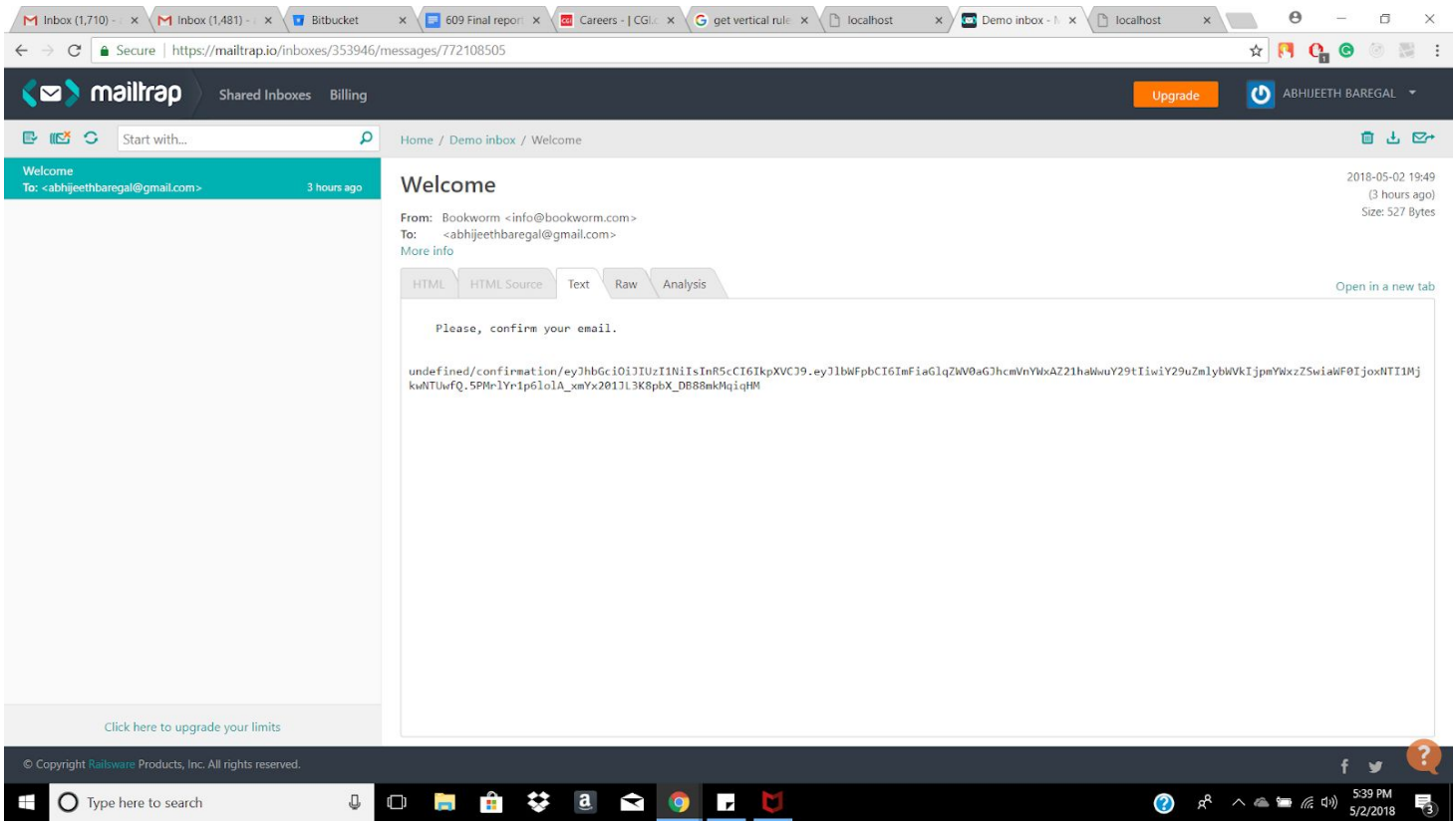
Dashboard

After the user is logged in, the admin will receive a verification link which he/she will send to the requested user to grant permission to access the admin portal



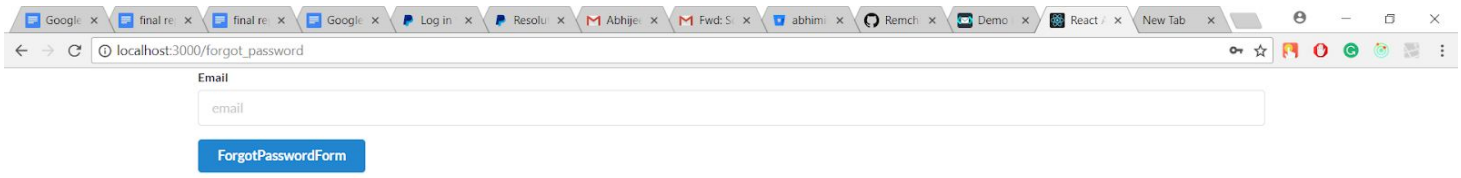
Mailtrap.io

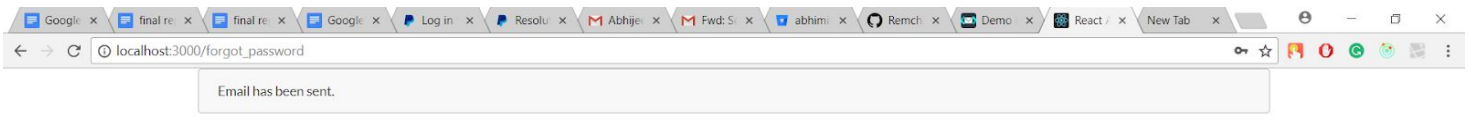
The admin portal receiving user access request token which is to be forwarded to the User



Forgot Password Request

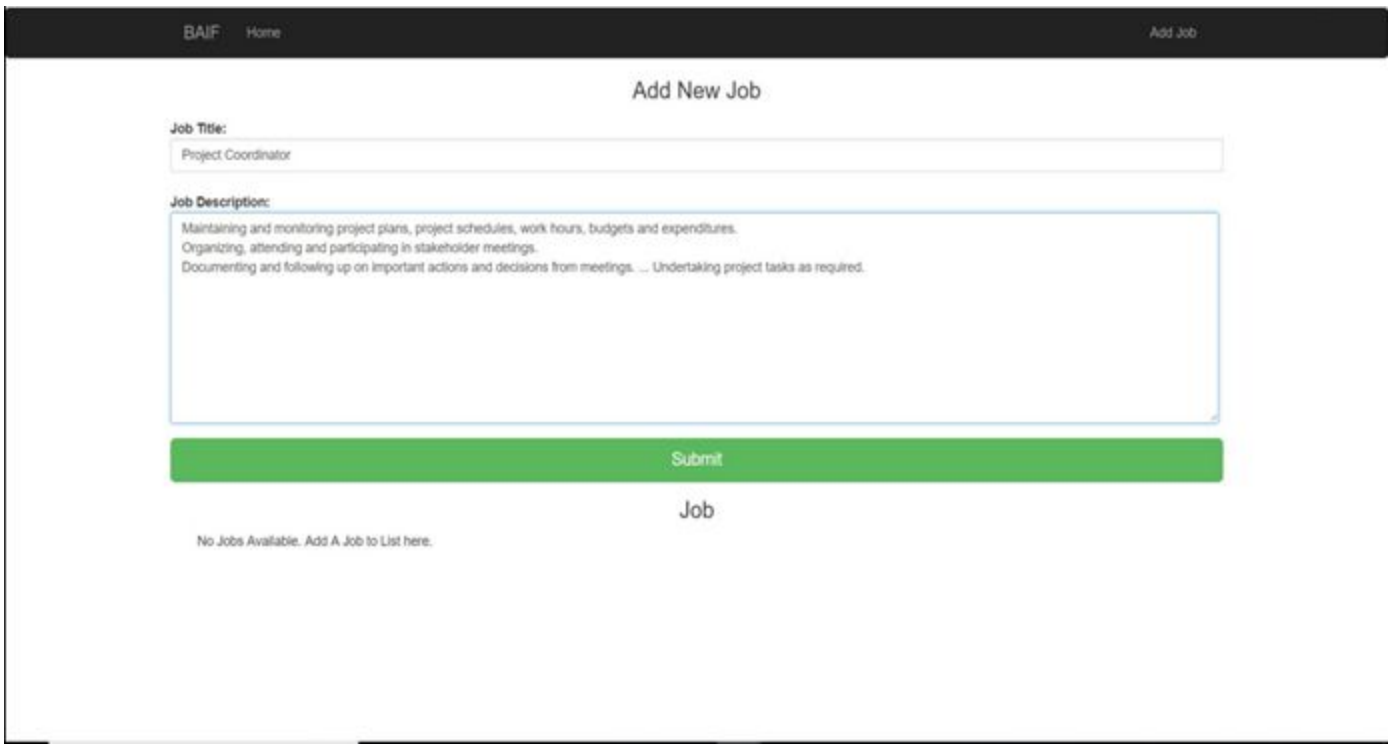
The User has to make a request in case if he/she forgets the password and admin has to give them access and they will have to set a new password to access the portal





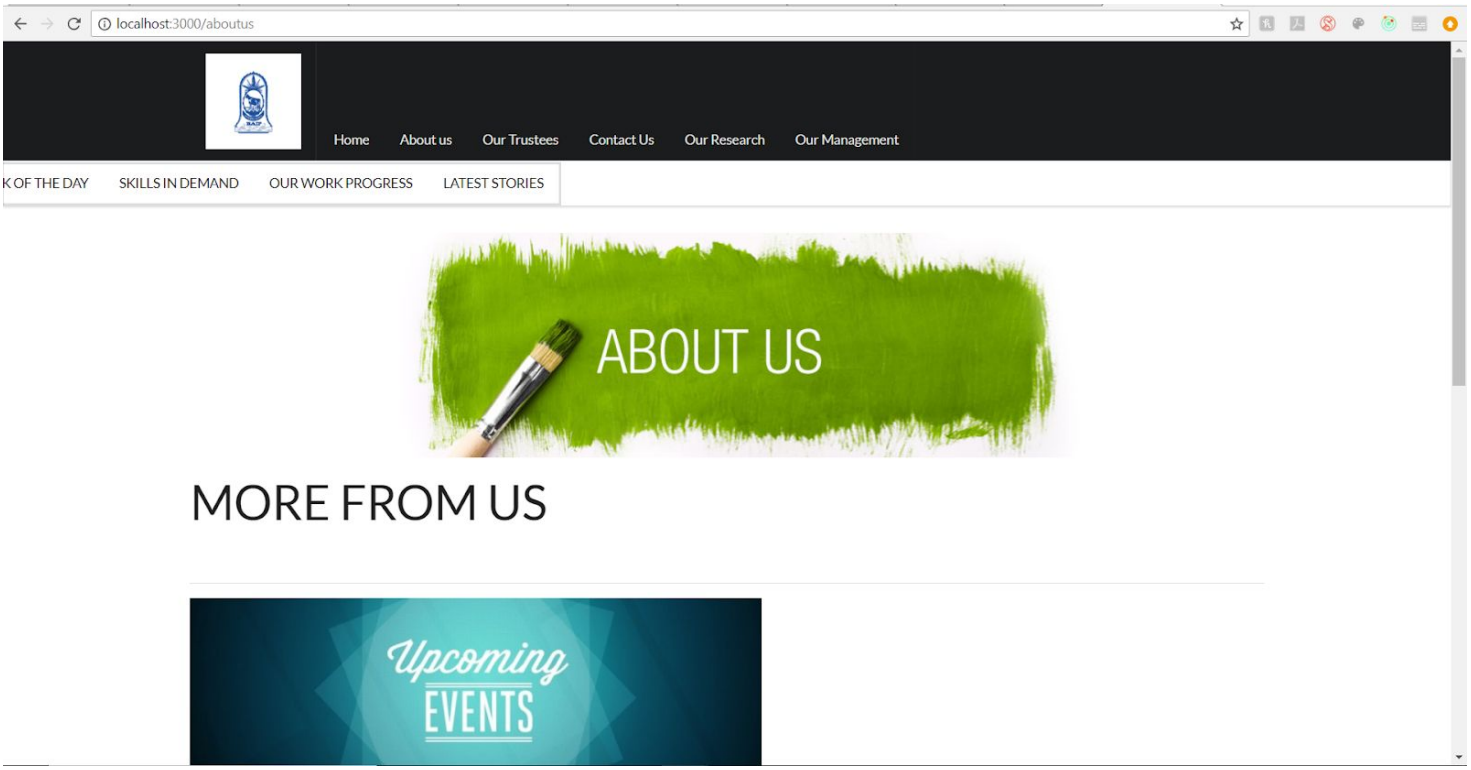
Job Portal

The respective user can add a new Job, they will be able to edit the jobs they posted, update/ delete them. Users can also see the jobs posted by other Users but cannot perform any operations on them



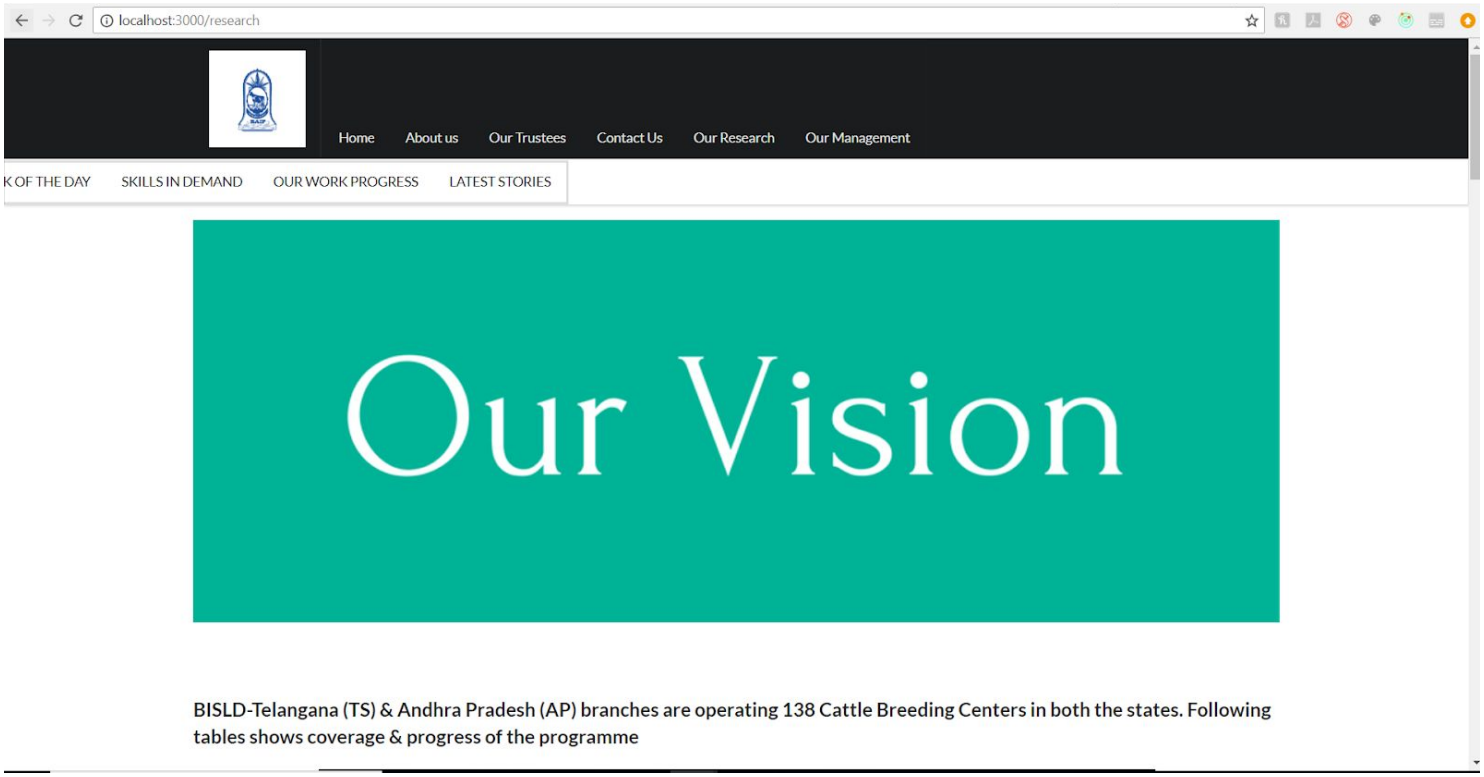
About Us Page

The About Us page gives details about various upcoming events, ongoing researches and projects with attractive image hover and popup features done using Semantic UI and JS.



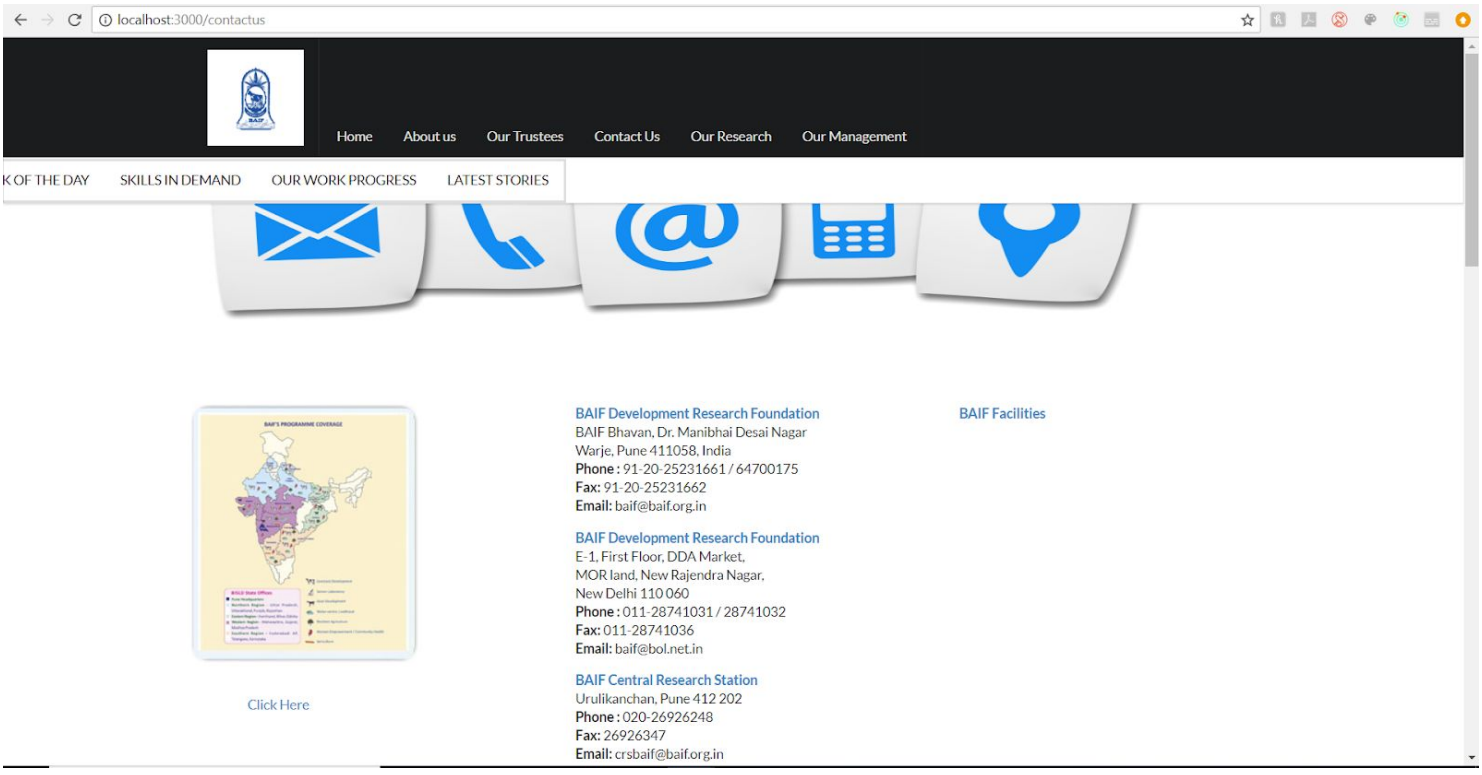
Research Page

This page includes information about various areas of researches being done in the company with image hovering feature and details of the contact person for the same.



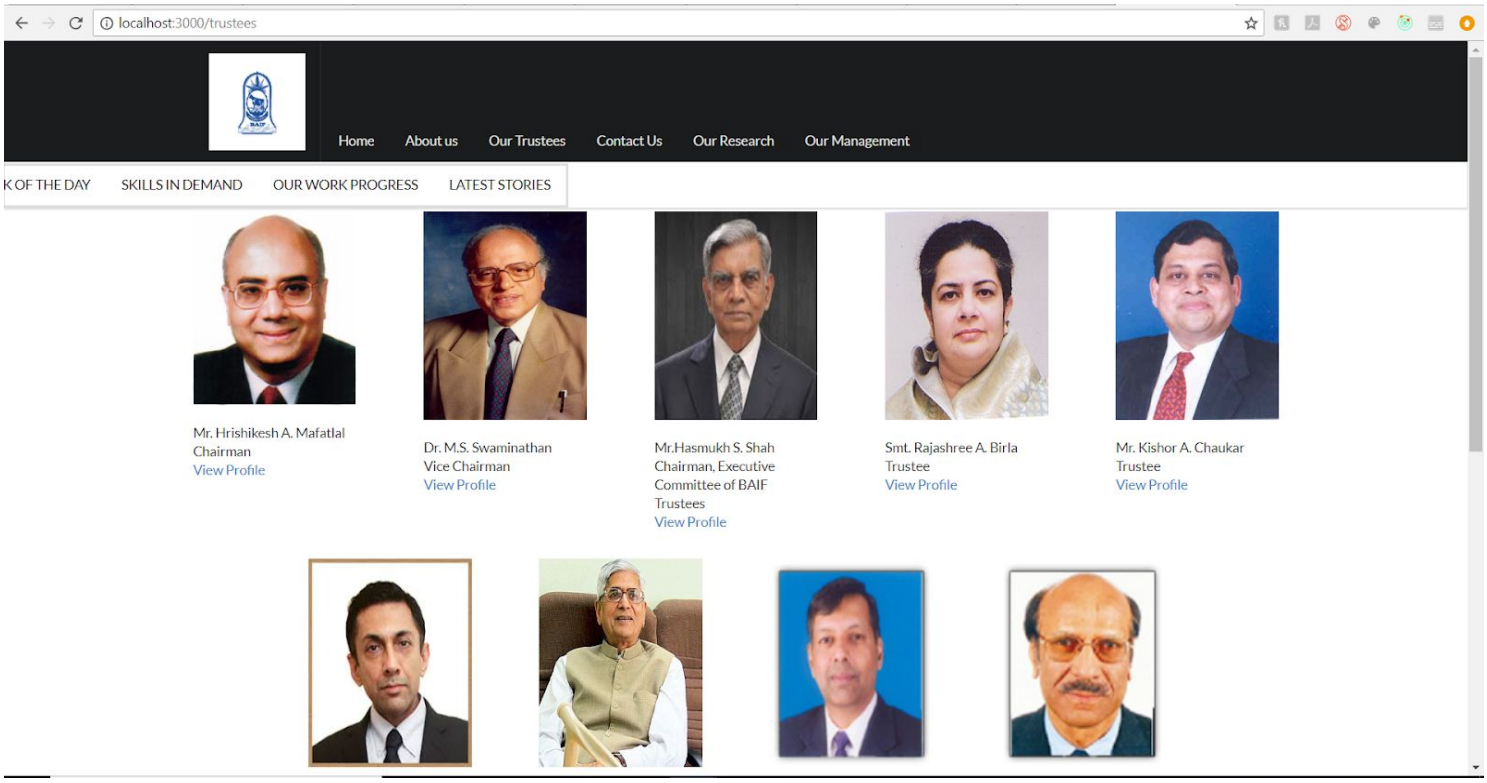
Contact Us Page

Includes details of all branches of the company, there email-ids, phone numbers, fax etc



Trustees Page

This page introduces all the trustees of the company in an attractive grid form along with "View Profile" option that opens a more detailed file about them.

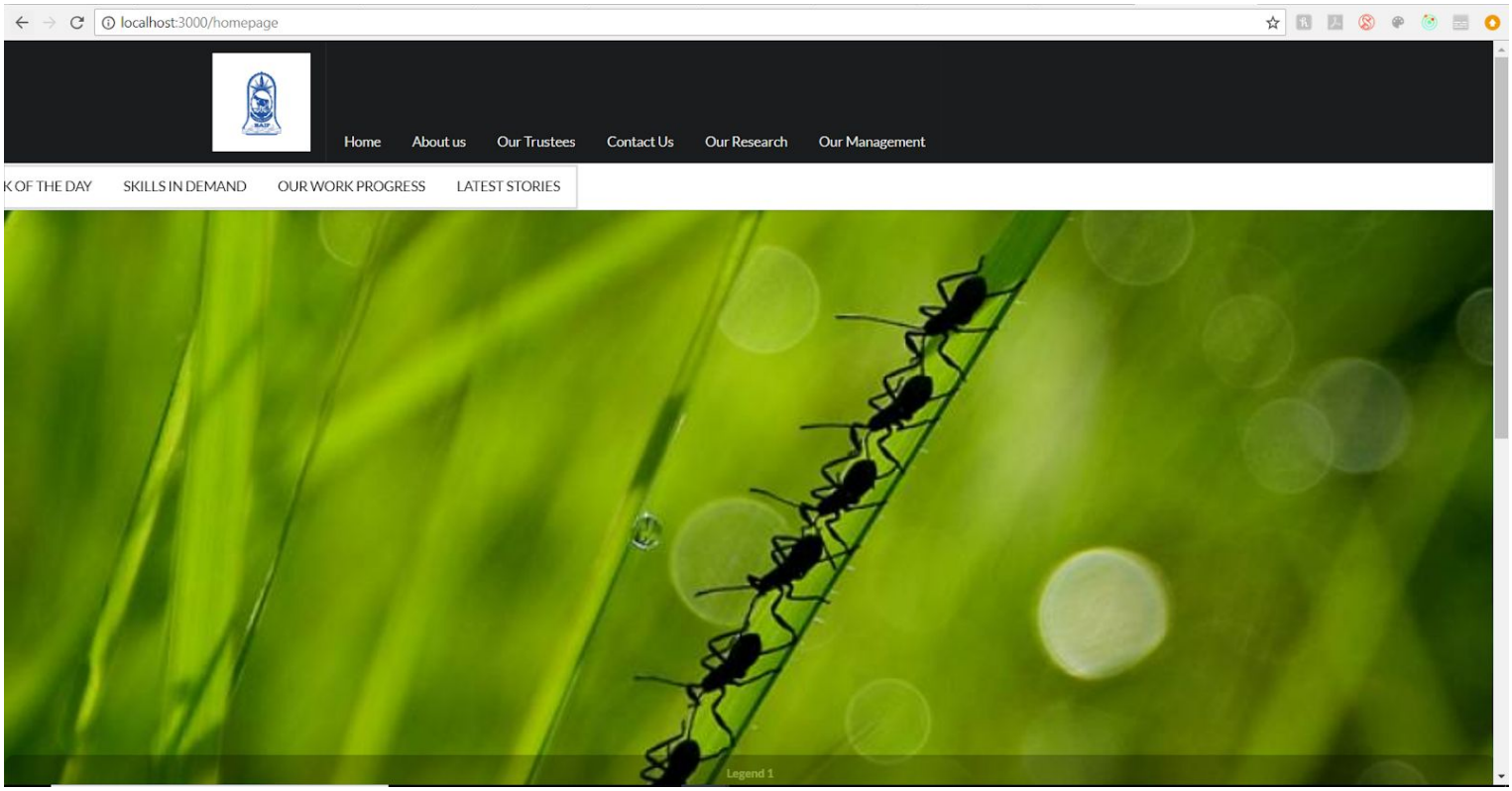


Home Page

HomePage facilitate navigation to other pages on the site by providing links to prioritized and recent articles and pages. An attractive feature with sliding images which provides information about the work and mission of company. There is easy access to various other pages through two navbars

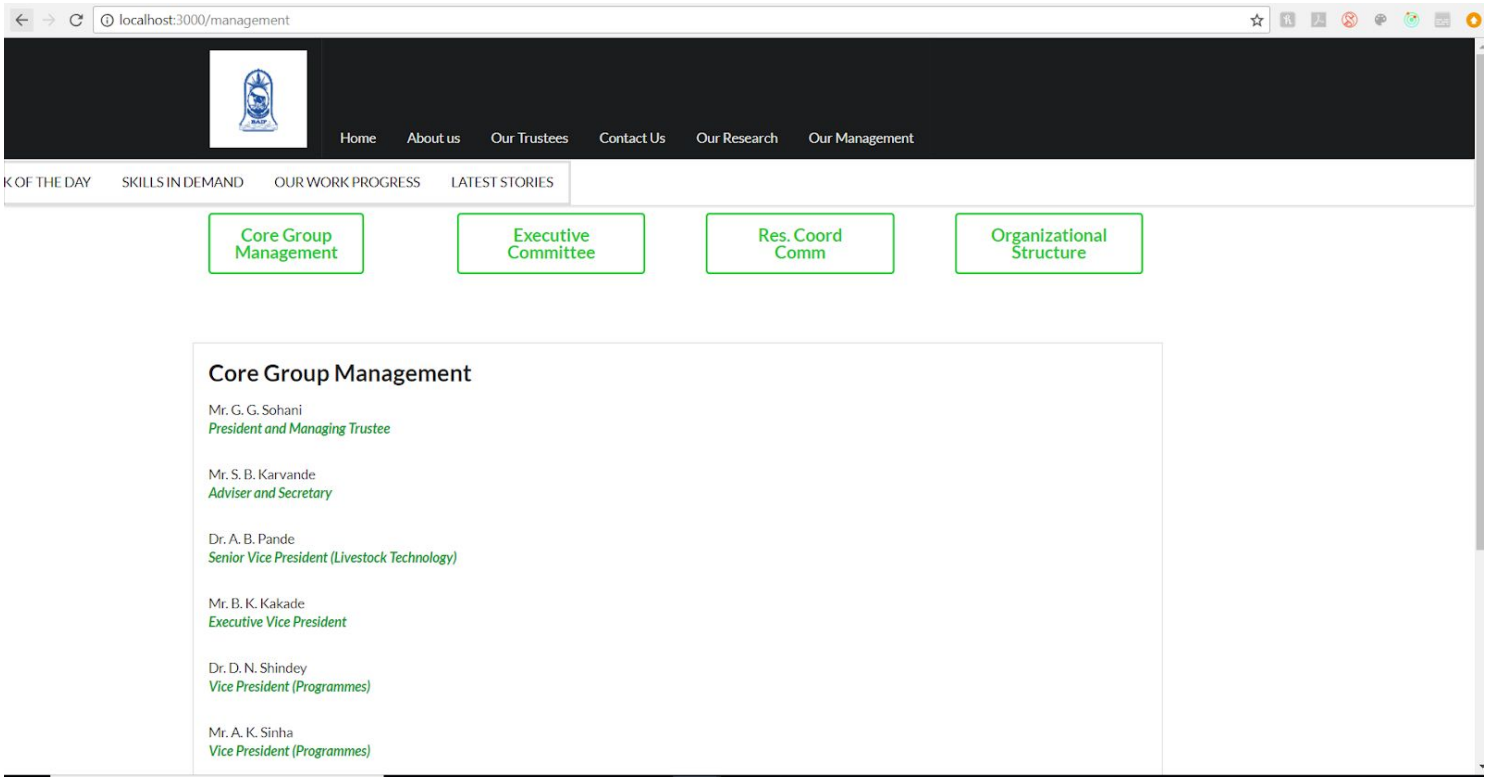
Contents

1. About us
2. Contact us
3. Our Research
4. Our Management
5. Our Work Progress so on..



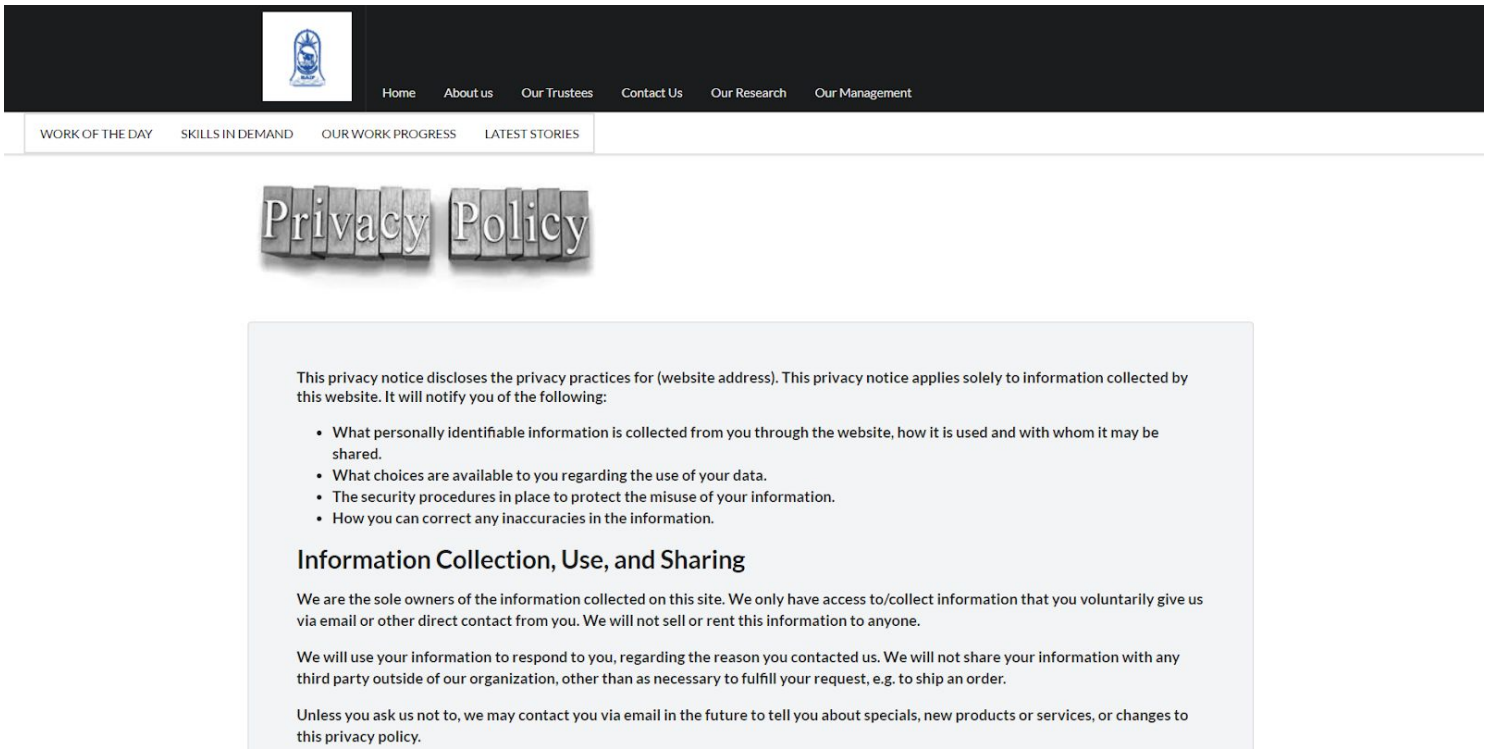
Management Page

This page includes details about the managing committee of the company with different designations such as core group, executive committee, research committee and organisational committee which are loaded on the same page but with different content as needed by using “Tab” feature.



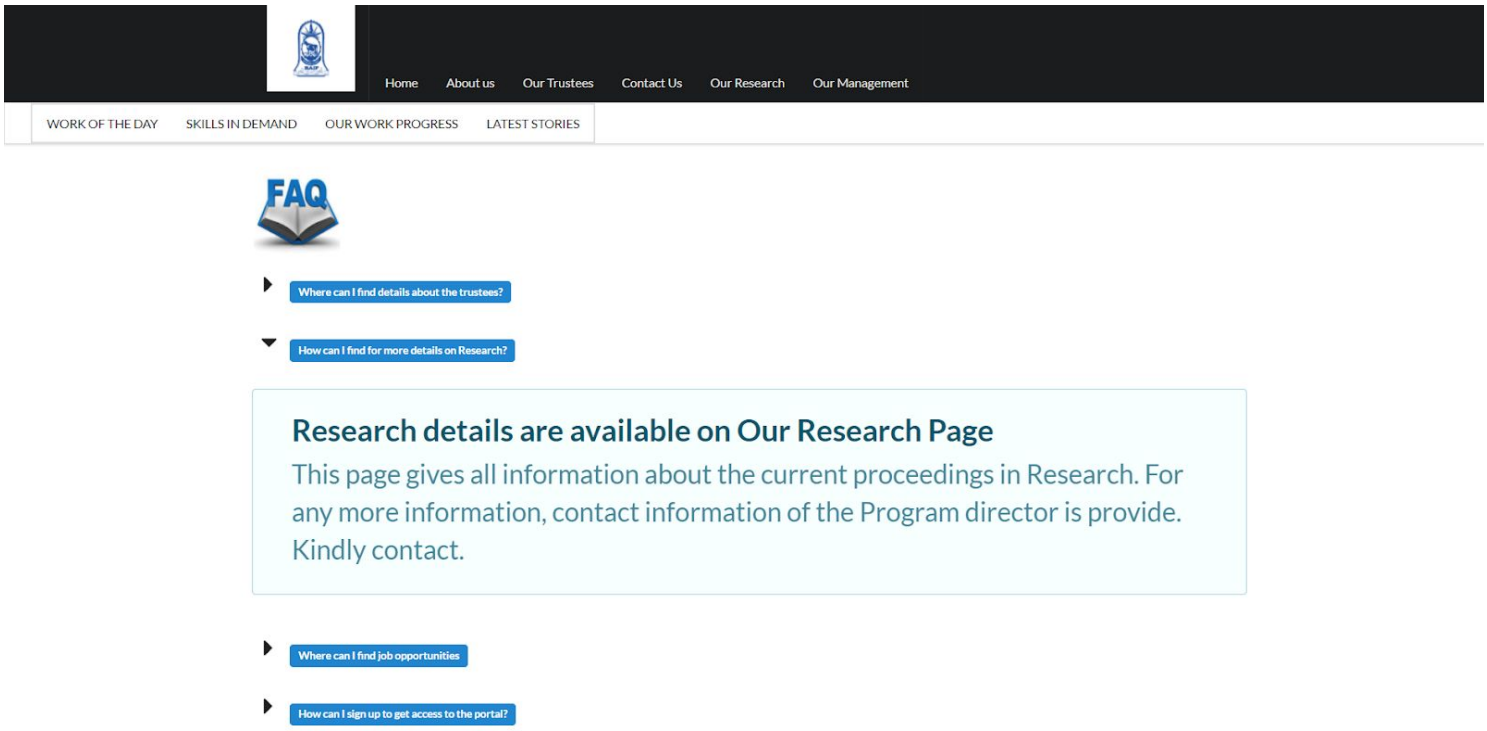
Privacy & Policy Page

A **privacy policy** page is developed that contains the statement or a legal document disclosing some or all of the ways a party gathers, uses, discloses and manages a customer or client's data. This is not something to take lightly.



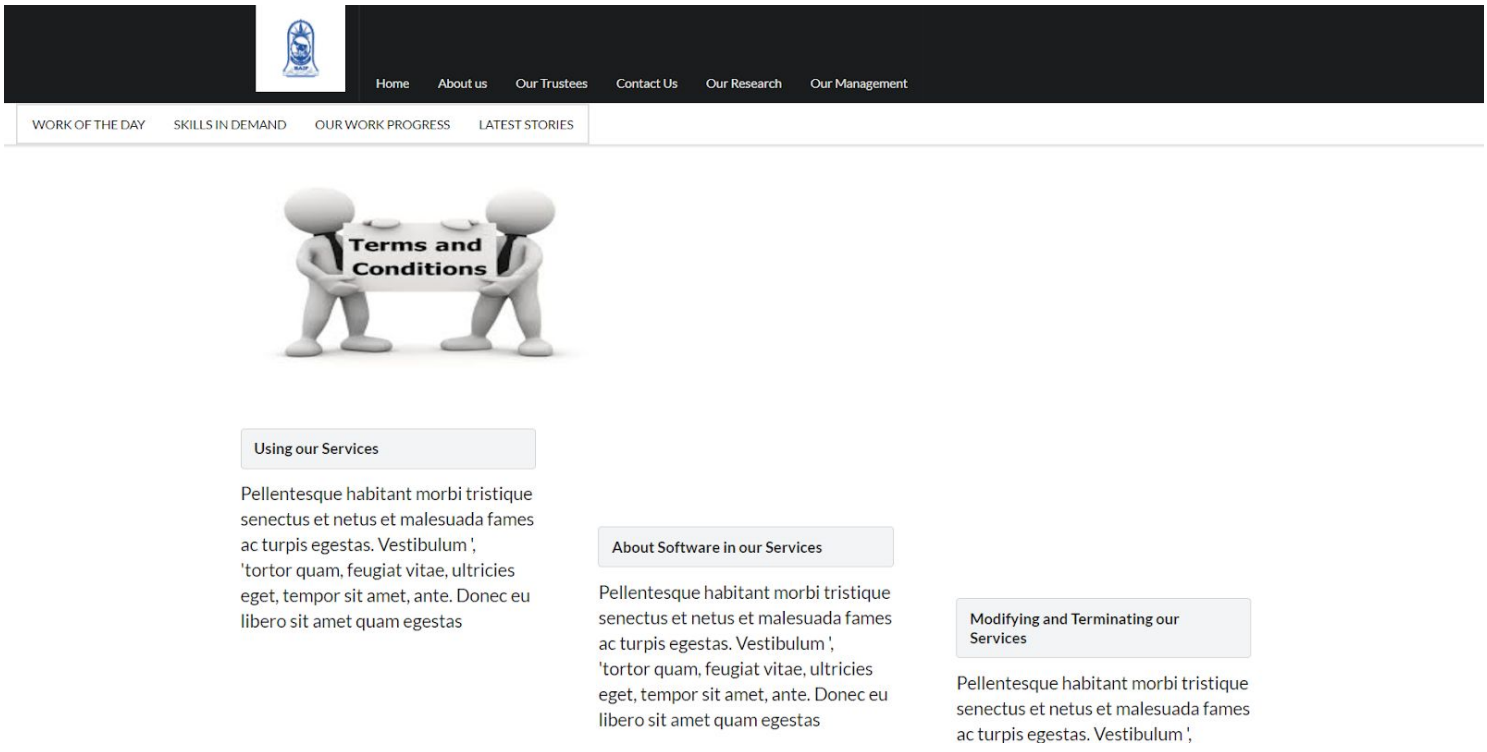
FAQ Page

An **FAQ** page is developed dedicated to listing and succinctly answering **frequently asked questions** from **website** visitors.



Terms and Conditions agreement

Web Page is developed to include the terms, the rules and the guidelines of acceptable behavior, plus other useful sections



CHALLENGES FACED

- To clearly identify customer's requirement
- Writing a maintainable code
- Time-boxed development
- Merging the branches
- Getting the auth token to work

Steps to Run

1. Clone the two front-end and back-end project files in different folders- git clone URL
2. To install dependencies-npm install
3. Install and setup MongoDB
4. Run MongoDB using mongod --dbpath “location of MongoDB”
5. Database- “use merndb”
6. Open localhost:3000 in your browser

References

- https://reactjs.org/
- https://expressjs.com/en/api.html
- https://nodejs.org/en/docs/
- https://semantic-ui.com/
- https://www.youtube.com/watch?v=A71aqufiNtQ&t=19s
- https://stackoverflow.com/
- https://www.youtube.com/watch?v=s1swJLYxLAA