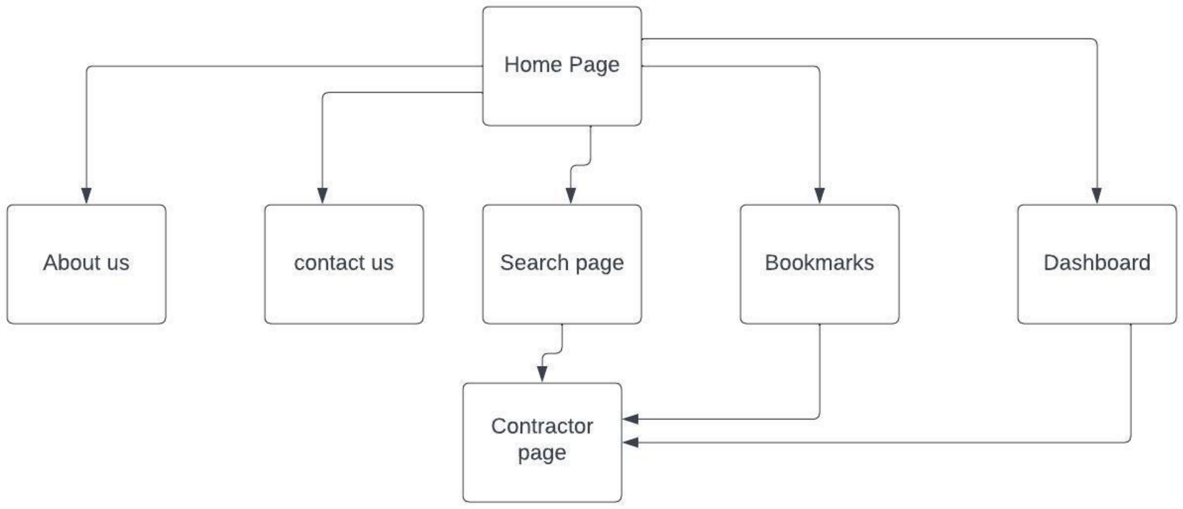
1. **Overview**:

proVis

A website to make the process of house construction(interior design, furniture etc) contracting frictionless and reliable. The idea is for the site to act as a middleman between Contractors and customers by showing all contractors according to customer's requirements and enabling them to contact them via email.

Team Members-
Eklavya Agarwal - 2022VST9017
Mihir Kaskhedikar - 2021CS10551
Aaveg Jain - 2021CS10073
Tejas Anand - 2021CS50595.

2. **Content Structure**:



3. **ER Diagram**
4. Github link- repo
4. **Suggestions given in the Demo**

    1. We prevented the user from being blocked from using our website while their API response for translating text was executed.

5. **API descriptions -**
    The main API endpoints that are used in our website are :

- Sign Up Page: On this page the user can create an account on the proVis website. It would make use of the **email_register** API endpoint in user controllers.
- Login Page: On this page an already existing user in the database can login their account. It would make use of the **email_login** API endpoint in user controllers.
- Product Listing Page : On this page we can search for products (and their contractors) by locations and categories. This uses the **get_product_by_tags** API endpoint. From the list of contractors we can connect with any contractor and it would call the **get_product_by_id get_contractor_by_id** and **get_company_by_id** endpoints. We can also bookmark any product and this uses the **post_bookmarks_by_customerid** endpoint.
- Product Profile Page: On this page we can see the details of the contractor and the company he/she is associated with. We can order the product linked with a contractor by calling the **post_order_by_customerid** API. This would send an email to the contractor telling them that we want to meet them and also give them the customer's email-id and message who is ordering their product.
- Dashboard: On this page we can see our account information as well as our current orders. This would use the **get_orders_by_customerid** API endpoint.
- Bookmarks Page: On this page, we can see all of our product bookmarks. This would use the **get_bookmarks_by_customerid** API endpoint.

5. **Functionality**: A detailed description of the features of each webpage.

Description -

Login- Home page has a signup and login modal accessible from navbar.
1) Sign Up Modal with OTP feature.
2) Login Modal

Home page: The home page is the first page that customers will see when they visit your app. It should provide an overview of your app's features and functionality, and make it easy for users to navigate to other sections of your app. Has testimonials etc.

About Us: This page should provide information about your company, including its history, mission, and values. It can also include details about the construction companies listed on your app and any partnerships you may have with them.

Contact Us: This page should include contact information for your company, including phone numbers, email addresses, and physical addresses.

Dashboard: This page should provide users with a view of their account, including their current orders and personal details.. Once the request form is filled for a company, the request is added to the dashboard.

Search page: This page should allow users to search for construction contractors based on their location, experience, service or other criteria. Users should be able to see a list of all the companies that match their search criteria. On clicking on any company will take the client to the company profile page. If logged in, users can also bookmark any contractor for later review (displayed on the bookmarks page).

Also, the users can translate the contractor description to Hindi and back to English if they want.

Contractor Profile Page: This page should provide detailed information about each construction company contractor listed on your app, including their background, experience and portfolio. The customers, if logged in, can send a request to the contractor via email.(uses SMTP, so can't be used on Baadal. We have attached a screenshot of the same and it was also demonstrated in the demo). The email has the email address of the customer, and hence the contractor can contact them back.

Bookmarks Page- This page displays all the bookmarks of the currently logged in user, and on clicking on the contractor card displayed we can go to the respective contractor profile page.

6. Design Decisions-

We share our design decisions by going through the workflow. We started by two members each working on the frontend and backend. We decided to use the popula**r NextJS framework** to make our frontend due to the **large support available** and flexibility . To make our API calls we had access to a large number of libraries lik**e fetch, axios** etc.We **initially used Promise.All for multiple calls togethe**r, but **switched to async axios and fetch** along the way due to interval server error caused due to its use. We used several libraries like **antd, headlessUI, materialUI, radixUI** to recycle UI components instead of making them from scratch.

While making the frontend and backend, we observed that our initial API and database design was too cumbersome and messy to implement. Thus we had to redesign our API and database to make it easily implementable for both frontend and backend.

Backend design decisions (Login and Signup)-

1. We have used swagger_codegen to generate code of our server using the specification of our API given in the .yaml file
2. For the database setup, initially we had thought of using SQLAlchemy with sqlite database. But then we thought that this wouldn't scale well with the size of our potential database, so we went with a production level MySQL with PyMySQL as the python library used to make SQL queries.

3. The swagger_codegen generated all the templates of the controller functions that we needed to complete to get a working API. We just needed to write the logic of all the API endpoints
4. We decided to secure our users with a 2 factor Email based OTP Authentication.
5. Whenever a user tries to Register to our website, the following sequence of events take place :-
   1. Frontend sends registration request to the backend
   2. Backend generates the OTP and emails it to the user
   3. User enters the OTP to the website
   4. Frontend sends verification request to the backend
   5. Backend verifies the OTP and based on the verification status,
      If the OTP entered is correct, we add the user to the database and send a JWT token to the frontend.
6. While adding the user to the database, we never add the password directly. We first generate a salt, which is a random string that is generated at the time of registering, and then hash the password after mixing it with the salt. This ensures that even in case of a database breach if a hacker tries to decrypt the password with precomputed hash tables like the rainbow table, he would not be able to get the passwords of our users.
7. We are sending the OTP through an email service SMTP. Unfortunately, SMTP is not allowing gmail login from Baadal VM, so Abhilash Sir told us to remove it from the final submission. But we showed this feature in the presentation. We would be attaching screenshots of the working OTP system in the report.

7. ML API used-

We decided to use this text translation api to incorporate English to Hindi conversion (and vice versa). We felt that in a majorly Hindi speaking nation, people would be more comfortable reading Hindi instead of English.

8.Unique features-

1. The site uses two factor authentication using OTP, which is used by all commercial sites nowadays.
2. Also, the user is allowed to communicate with the contractor via email through our site.
3. We have added multiple animations in our frontend such as the home page, about us page etc.
4. The UI of our site is very user friendly and appealing.

9.Challenges encountered-

1. For quite some time we kept receiving interval server errors on our API calls at random. After a lot of debugging and on exploring other libraries, we found that the problem was

caused due to the use of Promise.All to make multiple fetch requests together, and was solved by using async axios and fetch libraries.

10.Coverage Report

Coverage report proVis.pdf

We got a coverage of 83%

Baadal - http://10.17.50.185:3000 - link of website hosted on Baadal VM.

11.Future Work/Limitations-

Currently the site doesn't support reviews (the functionality has been implemented on the local machine but is causing an internal server error on Baadal, hence the submitted code has this part commented. The associated files can still be seen on the repo link and Baadal code)

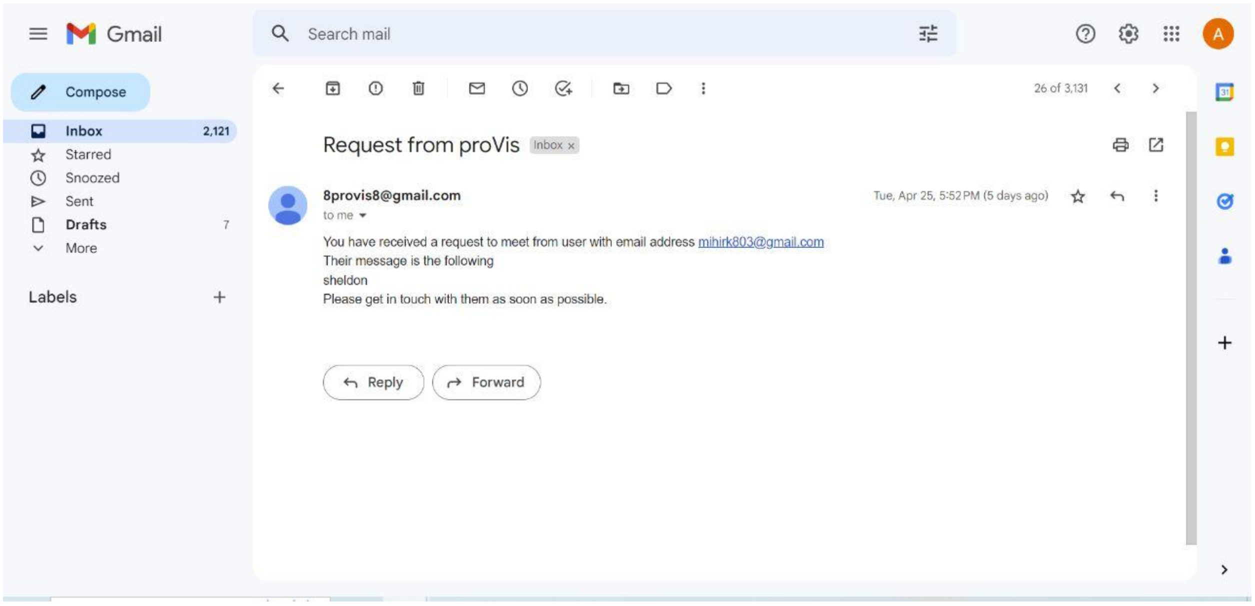
12.Scope for improvement-

We are planning to add a recommendations ML API to display contractors according to the past history of the logged in user. Also, we are going to incorporate a ChatBot-AI known as proVIsAI that will guide new users through our website, we will do this using ChatGPT ML API.

We are also planning to add a contractor login to our site who will be able to add his/her products to the site using forms. The signup of the contractor will require proper verification manually to ensure fake contractors and fake companies are not able to sign up. Also the product uploaded will have to be verified rigorously to ensure correct data is displayed on the site.

We also plan to add a budget management tool and track request progress page.

13.LLMs- DALLE-2 Was used to make logos for our website
ChatGPT was used to answer various queries like terminal commands, It wasn't used
For code generation because the code it generated was non-functional most of the times.



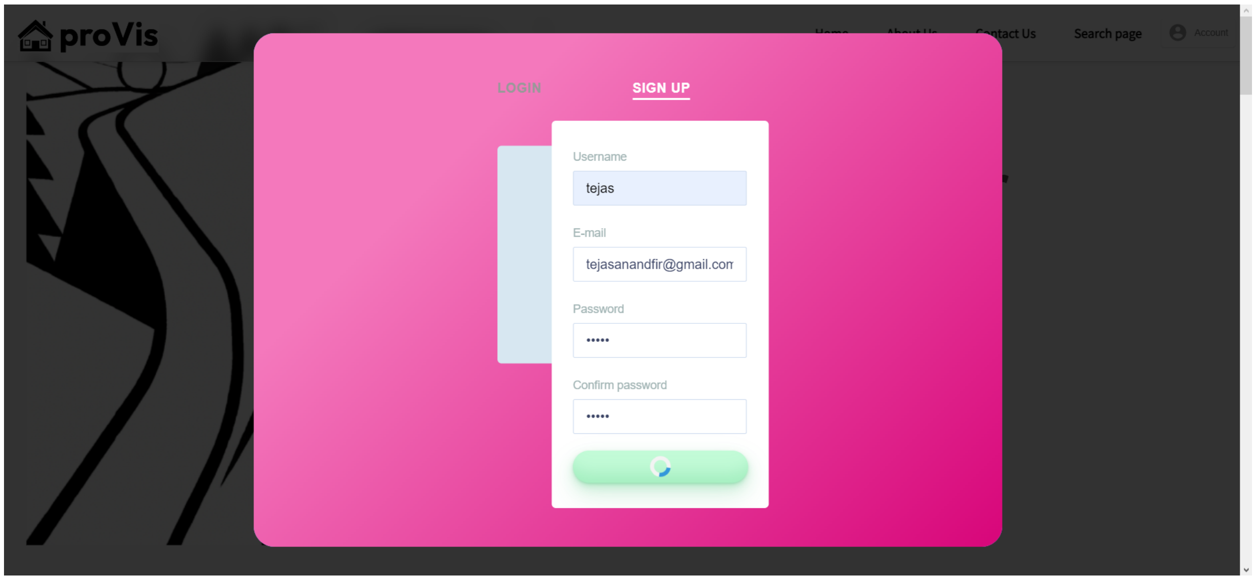
Request from proVis Inbox

8provis8@gmail.com
to me

Tue, Apr 25, 5:52 PM (5 days ago)

You have received a request to meet from user with email address mihirk803@gmail.com
Their message is the following
sheldon
Please get in touch with them as soon as possible.

Sent Mail with the user request order

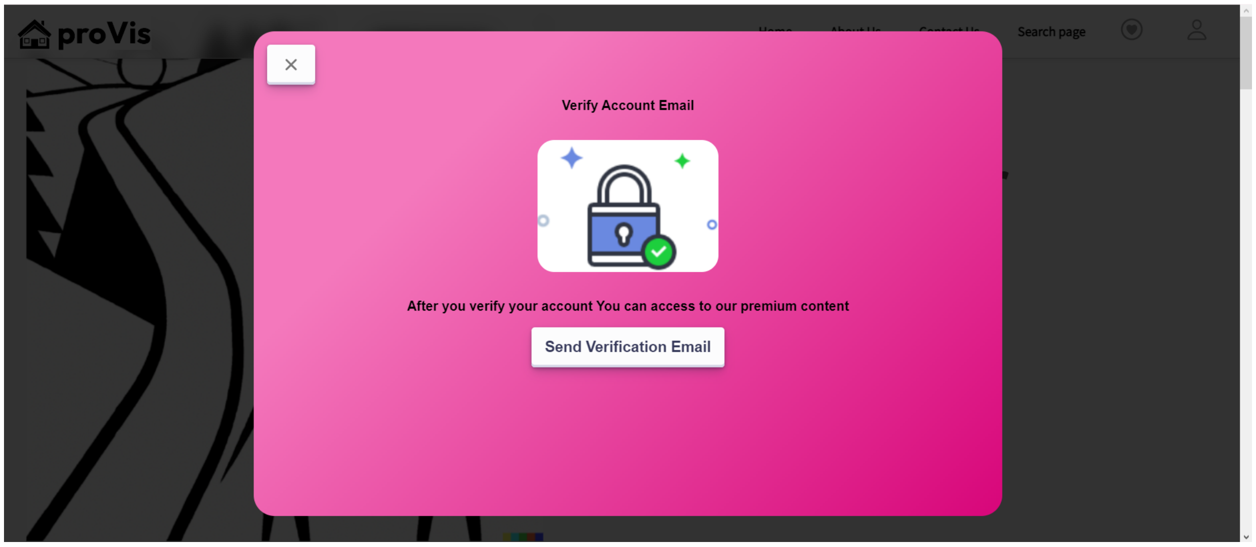


Username: tejas

E-mail: tejasanandfir@gmail.com

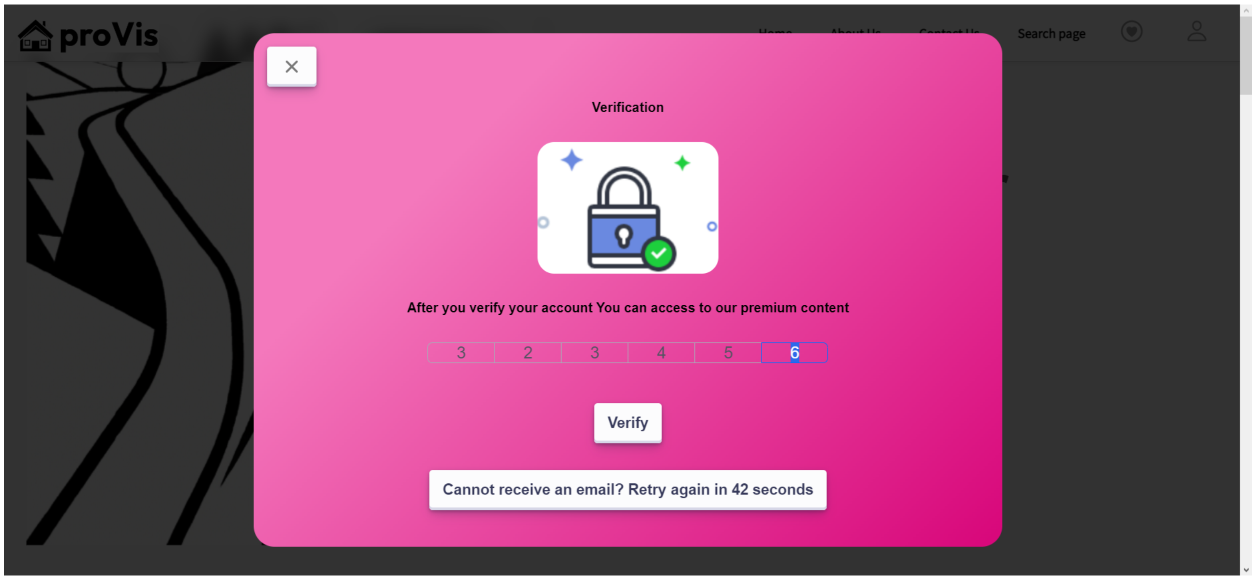
Password: •••••

Confirm password: •••••



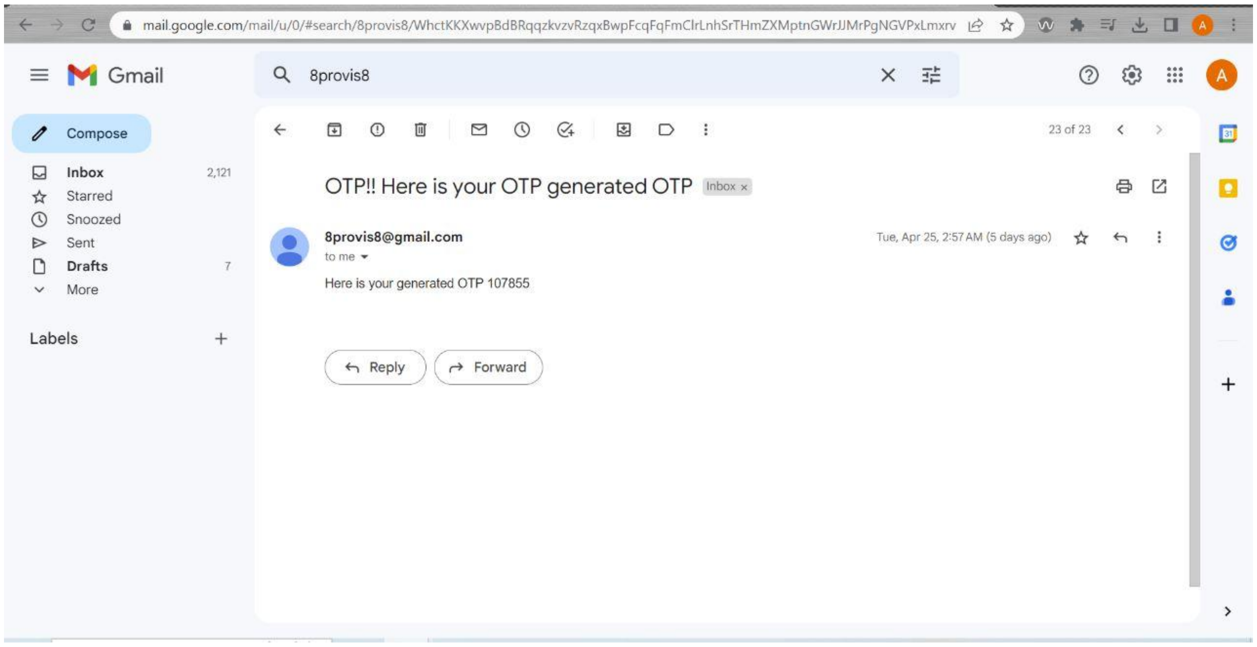
Verify Account Email

After you verify your account You can access to our premium content



Verification

After you verify your account You can access to our premium content


mail.google.com/mail/u/0/#search/8provis8/WhctKKXwvpBdBRqqzkvzvRzqxBwpFcqFqFmClrLnhSrTHmZXMptnGWrJJMrPgNGVPxLmxrv
Gmail
8provis8
Compose
Inbox 2,121
Starred
Snoozed
Sent
Drafts 7
More
Labels
23 of 23
Inbox
Reply
Forward

OTP!! Here is your OTP generated OTP

8provis8@gmail.com
to me
Tue, Apr 25, 2:57 AM (5 days ago)

Here is your generated OTP 107855

Working 2 Step Verification Screenshots

Week 1 submission - We made all our submissions with the same google doc and also made the report on the same doc (then converted to pdf). Thus we cannot attach the google doc links for past submissions. If possible please refer to our previous submissions for them. The figma links etc submitted previously are available however below.
Dataset -
https://drive.google.com/drive/folders/1mh5v3DLZxEoP4r8BWO1FztRrzEatiDWF?usp=sharing

Week 3 submission -

Figma project link -
https://www.figma.com/file/8PLvE2kwBMbR2JgPEu1EXT/provis-site?node-id=0%3A1&t=XczGgCDEVnwsuEYy-1

Google drive link of folder containing yaml file, video of interaction, and er diagram -
https://drive.google.com/drive/folders/1xsPuu3B00IV0zntltZ92LKoiw2R0JMlp?usp=sharing

Week 6 Submission:-
Video :-
https://drive.google.com/file/d/1RYYrU7bjkvoA48gww1DqEofemLmxfC7q/view?usp=drivesdk
Repo:-https://github.com/andTEJAsan/proVis (unit tests included in swagger server)

http://10.17.50.185:3000 - link of website hosted on Baadal VM.

Token Split

Aaveg Jain	10	frontend
Tejas Anand	10	backend
Eklavya Agarwal	10	frontend
Mihir Kaskhedikar	10	backend