**Project Documentation: Piramal-UI
(https://piramal-ui.netlify.app/)**

Technology stack used
- HTML, Java Script, React,
- CSS, Material Icons, Tailwind Css (For styles)
- Axios (For calling API)
- Sheet DB (for storing data)
- Netlify (For deploy the site)

Steps

I have created separate components to bifurcate the code into reusable
And added validations on the form and loader for good user experience

Integration

For setting up Google Sheet to receive data from the contact form
submissions and configure API integration for writing data to the sheet, I
followed these technical steps

1. Create a Google Sheet:
   - Logged in to Google account.
   - after that Google Sheets (sheets.google.com) and create a new
     spreadsheet.
   - Label the columns to match the data fields you want to capture from
     the contact form.

2. Share Permissions:
   - Click the "Share" button in the top right corner of the sheet.
   - Share the sheet with the Google account associated with the API
     integration.
   - Provide "Edit" permissions to allow writing data to the sheet.

3. Generate Google Sheets API Credentials:
   - for google API I have used Sheet DB.
   - Firstly signed in Sheet DB
   - Click on create
   - Then provide sheet link
   - Then this API link is ready to use

4. To implement the API integration use Post method to update the user data in sheets also the API which was given as per requirements.

```
Requested body structure was  -
{
name: "",
mobile: "",
email: "",
}
```

5 . For calling API I have used axios. Further step was API Call and Response Handling and Error Handling for handling the errors I have use tost massages using react-hot-toast

6. After successful post request of contact form submissions and data verification is successfully done to both Google Sheets (https://docs.google.com/spreadsheets/d/1SmH3b6kD-uuljlOUAVVWWIeKXyUScGc3QVtsr2mwEpQ/edit#gid=0)
and the specified API data will get stored.

Challenges faced
- I wanted to make the UI exact same to which was given to me as reference so I faced some difficulties replicating the circular text design UI but eventually I was able to to make it.
- I was a little confused about which tech stack I should go about but due to time limitations I chose react because I was more comfortable in that.

Installation and Setup Instructions

Clone down this repository. You will need `node` and `npm` installed globally on your machine.

Installation: `npm install`

To Run Test Suite: `npm test`

To Start Server: `npm start`

To Visit App: `localhost:3000`