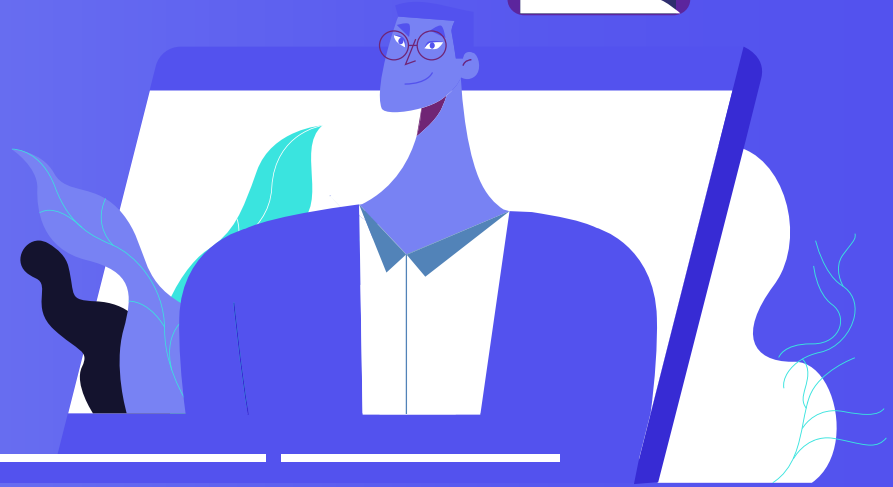


# MREC MEET

## VISHESH PROJECT EXPO

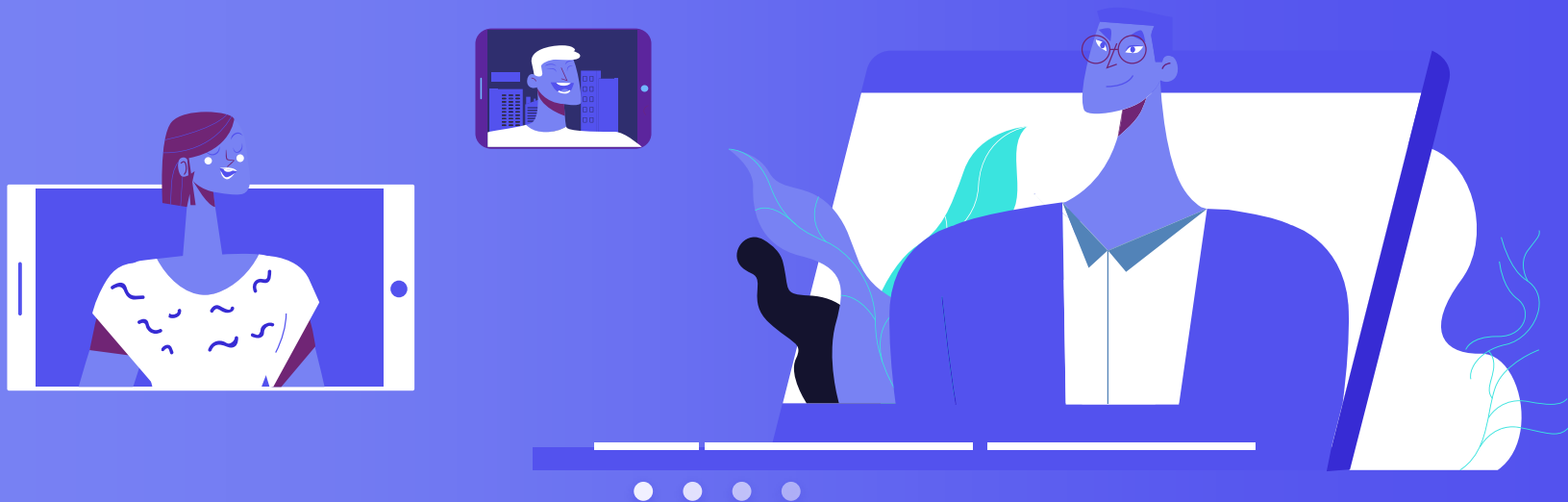
Team  
Members  
Rohan  
Nithin  
Jhansi  
Vijay  
Rakesh





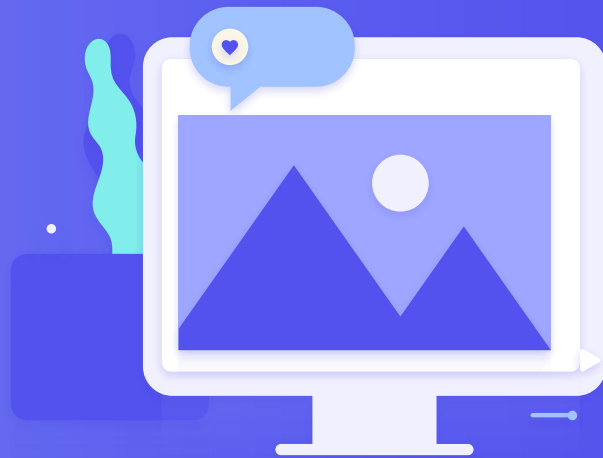
# What is **MREC MEET?**

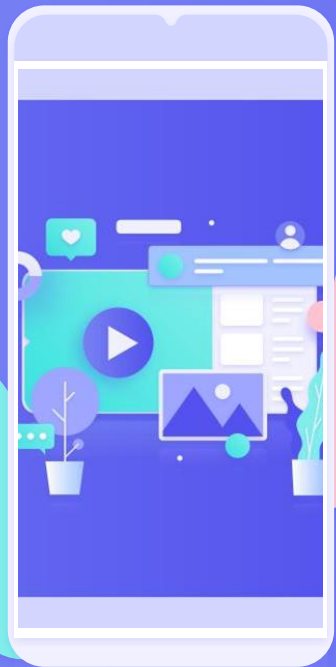
# A Light Weight Video Call Application.



It can be a

# Web Application



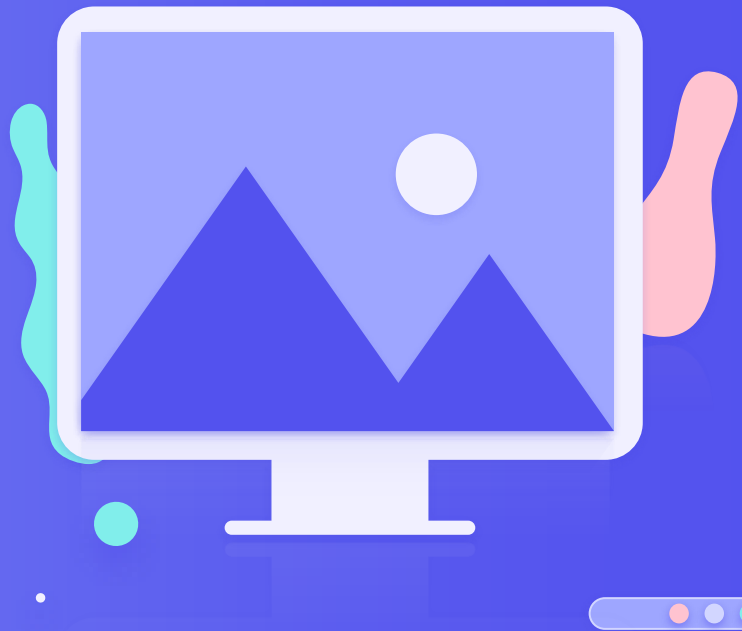


It can be a

# Mobile Application



It can be a  
**Desktop  
Software**





# Agile.

**Implementing Agile Methodology in Our Project**

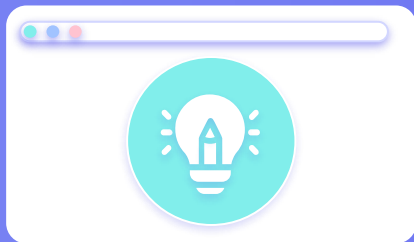
A blue rounded rectangle with a window-like header and a vertical menu on the right. The header has three colored dots (green, grey, orange) on the left. The vertical menu has four white dots on the right. The text "Meet & Plan." is in the center, and "01" is in the bottom right corner.

**Meet & Plan.**

**01**

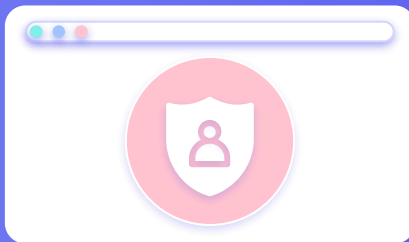


# What we are working on



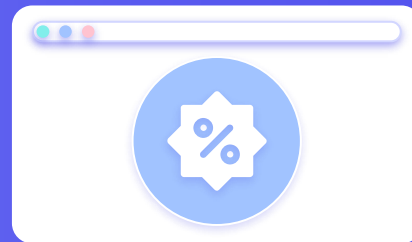
## Design

The Design should be light weight and user friendly with many useful features.



## Secure

Since it's a light weight application it's mandatory to have high security.

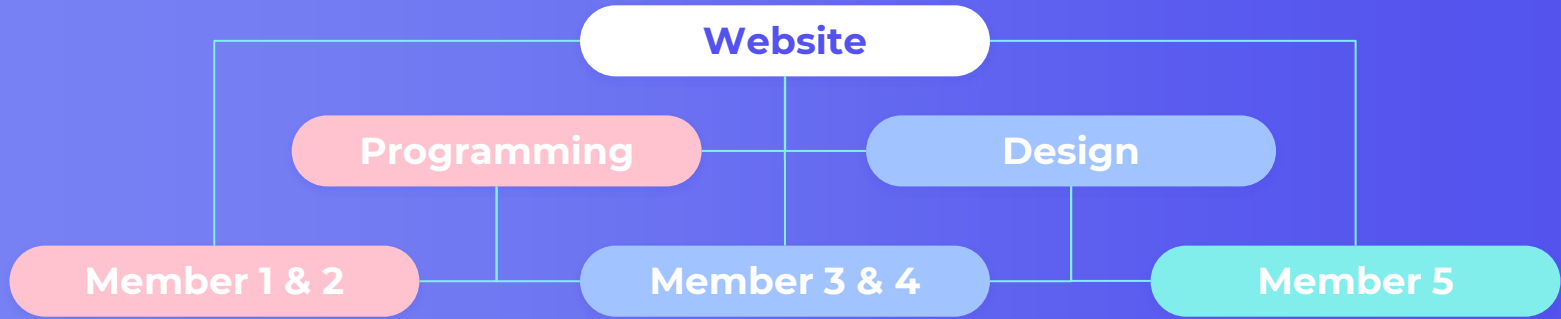


## Accessible

The Application should be accessible from every device to anyone.

# Process Flow Chart

Assigning tasks to each member in the team






**Design & Develop.**

**02**




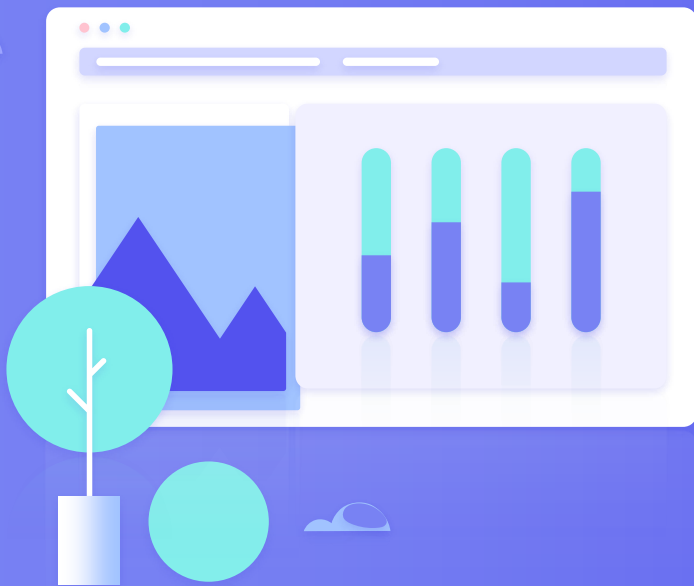
# DESIGN

Designing the UI/UX with the help of Figma



**Making it simple and user friendly wasn't a hard task but adding each feature to the site and making it look useable was defiantly inflexible.**





# First concept

Initially planned on using webRTC for the peer to peer video connection but later came across peer.js which helped in implementation of the video call and using socket.io for the call part and chat engine for the chat part

# Second concept

After building the first concept we realized that there are too many limitations and issues like latency, etc. We decided to use Twillo like popular tools like Uber, Airbnb etc. But ended up using something different.

Using TypeScript, JavaScript for the design and layout of the frontend and slowly building features like screensharing, polls, private dm, authentication, etc.



# Tools we used to build

## Frontend

JavaScript  
TypeScript

Go  
NodeJs

## Backend



## Database

Docket  
SQL

Vercel  
Heroku

## Hosting



The image features a solid blue background with a rounded rectangular frame. At the top left of the frame, there is a horizontal bar containing three small circles in green, white, and red, resembling a window's title bar. On the top right, there is a vertical ellipsis of four small white circles. Centered in the frame is the text "Test & Deploy." in a large, white, sans-serif font. In the bottom right corner, the number "03" is displayed in a very large, white, sans-serif font.

# Test & Deploy.

03



# Testing the Site

## Kanban Board

We used a Kanban board to organize my tasks by dividing them into 4 categories:

**To do, In Progress, Done** and **Bugs**.

We also set an In progress limit of 3, meaning if the In Progress column had 3 tasks, We had to finish them before we move on to start new tasks.



# Deploying the Website



## Backend

Deploying Backend to **Heroku** using Docker build from source.



## Frontend

Deploying Frontend to **Vercel** and connecting the Heroku backend.



**<https://mrec.vercel.app>**

**Publishing the Site**

# All the URL's



[Click here](#) - To access the git repo



[Click here](#) - To check the website without auth



[Click here](#) - To check the website with auth



**Thank You.**