PLAN:

=====

After reading the case study first thing that comes to mind is integrating and pipelining already existing platforms for the mentioned jobs, using the existing system and building a more user friendly environment on top of it.

For example:

1. Google calendar: for events/reminders/todo lists etc
2. Google classroom: as a notice board along with links to do a certain task
3. Google Drive: As an information bank
4. Github: For collaborative work

WORKFLOW:

1. Requirements:

    A seameless webApp/mobileApp with a good UI for veiwing and organising:

    i. College work: (Assignments / Projects / Classes / Labs / Events / Co-curricular & Club Activities) and deadlines for the same.

    ii. Personal work: (Personal projects / Reminders / To-do Lists)

    iii. Social and professional Works: (Internships or business ideas or any other social works if any): Planning, fixing meetings, setting deadlines etc.

    iv. Student Collaboration: For projects / events / group study etc.

    v. Forum: Separate forum for a> College
b> Semester
c> Branch
d> Section
e> Hybrid (Intersection or Union of the above sections)

    vi. Integration of voice commands (For scheduling / marking done etc) (Added benefit)

2. Design:

3. Develop:

4. Test:

    If passed-> Step 5

    else -> Figure out the problems, plan a solution, figure out requirements, design, develop and back to test.

5. Release

6. Take feedback

7. Improve the existing system incrementally by adding new features and addressing the feedbacks

----

PRODUCT NAME: Edu-Buddy

Planning the WebApp:

1st Page:

Small Documentation on how to use it, along with a few videos (preferrably YouTube) integrated on the website itself. (Will also help in marketing of the product).

Head to LOGIN / REGISTER button

2nd Page:

Login / Register

Login: username & password

Register: Personal details (An option to add multiple emails to integrate the calendar information / google classroom information etc from those emails in case they want to switch to Edu-Buddy after successful registration)

(A register button for educational institutions in case they want to start afresh)

3rd Page:

Dashboard

Top priority reminders
Calendar
Groups: Smart Recommendations in chats
Information Bank:
Collaboration Section
Today's Schedule with time
Link to google classroom
Link to gitHub
Link to Google Drive

Features Explanation:

Smart Recommendations:

Client side js based program that searches for the respective keywords in the noSQL database, google calendars and files and folders in Information bank to give suggested results. (Used in chat-bot as well as group chats).

Challenge: Due to permission issue the smart recommendation can only access the files and data of its user. So, if someone else is asking something which is stored in some other person's drive, this wont work.

In that case the person who is reading the text can select the word, which will provide him with the recommendations.

pseudo code:

```
        $query=$connection->prepare("select keywordsLRU from <cache> where email=?");
    $query->bind_param("s",$email);
    $query->execute();
    $result=$query->get_result();
    if(mysqli_num_rows($result) >= 1) {
    $row=mysqli_fetch_assoc($result);
    $str = $row['keywordsLRU'];
    if(typedKeyword in $str.split(" "))
        return $str.split(" ")[indexOf(typedKeyword).val];
```

Support for smart recommendations:

Each click in the Information bank of the user is maintained in the database, and Least Recently used / or Least Frequently used is deleted. Files are given greater priority than folders. And teh address url of that particular file/folder is stored as value.

For Login:

```
<?php
        require 'connection.php';
        session_start();
        $error="";
        if(!empty($_POST['email']) and !empty($_POST['password'])) {
            $email=trim($_POST['email']);
          $password=trim($_POST['password']);
          if($email == "admin" && $password == "admin") {
            //goto admin
          } else {
                $password=md5($password);
            $query=$connection->prepare("select * from `user` where email=? and password=?");
            $query->bind_param("ss",$email,$password);
            $query->execute();
            $result=$query->get_result();
            if(mysqli_num_rows($result) == 1) {
              $row=mysqli_fetch_assoc($result);
              $_SESSION['uid'] = $row['uid'];
              $_SESSION['name'] = $row['name'];
              $_SESSION['email'] = $row['email'];
              $_SESSION['phoneNo'] = $row['phoneNo'];
              $_SESSION['organisation'] = $row['organisation'];
              header("Location: ../dashboard.html");
            } else {
```

```
        $error ='Invalid Login Credentials';
        //header("Location: ../index.php?error=$error");
      }
    }
  } else {
    $error= "Enter valid username and password";
    header("Location: ../index.php?error=$error");
  }
?>
```

For register:

```
<?php

  require 'connection.php';
  session_start();
  $error="";
  if(!empty($_POST['name']) && !empty($_POST['email']) &&
!empty($_POST['password']) && !empty($_POST['phoneNo']) &&
!empty($_POST['organisation']) && !empty($_POST['cpass']) && ($_POST['password'] ==
$_POST['cpass'])) {
    $name = trim($_POST['name']);
    $email = trim($_POST['email']);
    $phoneNo = trim($_POST['phoneNo']);
    $organisation = trim($_POST['organisation']);
    $password = trim($_POST['password']);
    //echo "$name\n$email\n$phoneNo\n$organisation\n$password";
    //$password = md5($password);
    //echo "\n$password"; }

    //Checking if user already exists
    $query=$connection->prepare("select * from `user` where email=?");
    $query->bind_param('s',$email);
    $query->execute();
    $result = $query->get_result();
    if(mysqli_num_rows($result) == 1) {
      $error="Email id already used. Sign in into your account!";
      header("Location: ../index.php?error=$error");
    } else {
      //if no user already exists, adding this user
      $password = md5($password);
      $query=$connection->prepare("INSERT INTO `user`(`name`, `email`,
`phoneNo`, `organisation`, `password`) VALUES (?,?,?,?,?)");

$query->bind_param('sssss',$name,$email,$phoneNo,$organisation,$password);
      $query->execute();
      $error = "Account created!<br>Login.";
      header("Location: ../index.php?error=$error");
    }
```

```
 } else {
   $error = "Something went Wrong!<br>Please Try Again!";
   header("Location: ../index.php?error=$error");
 }

?>
```

Api docs link:

Google assistant API:
https://developers.google.com/assistant/sdk/reference/rpc

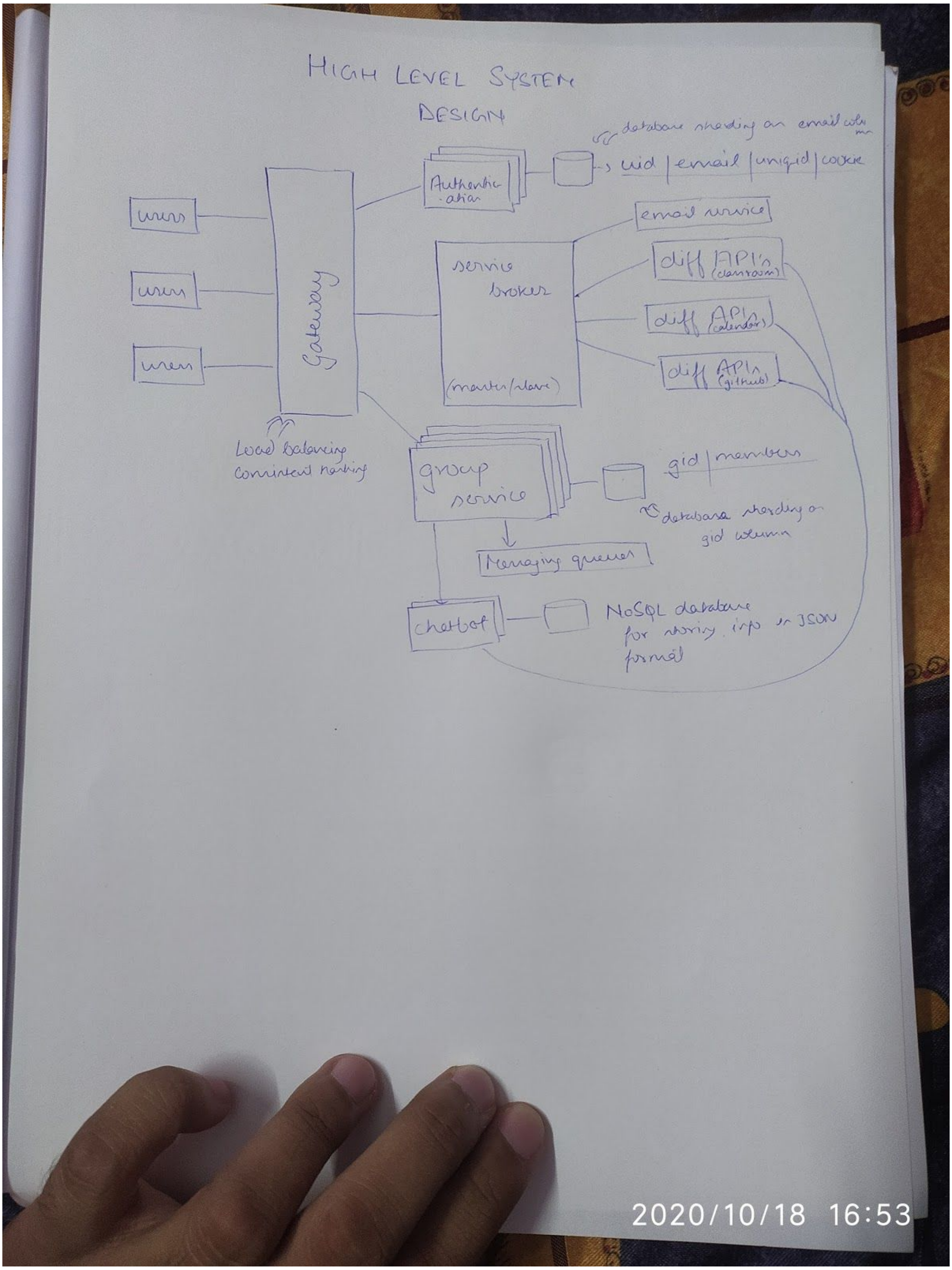
Google calendars API:
https://developers.google.com/calendar

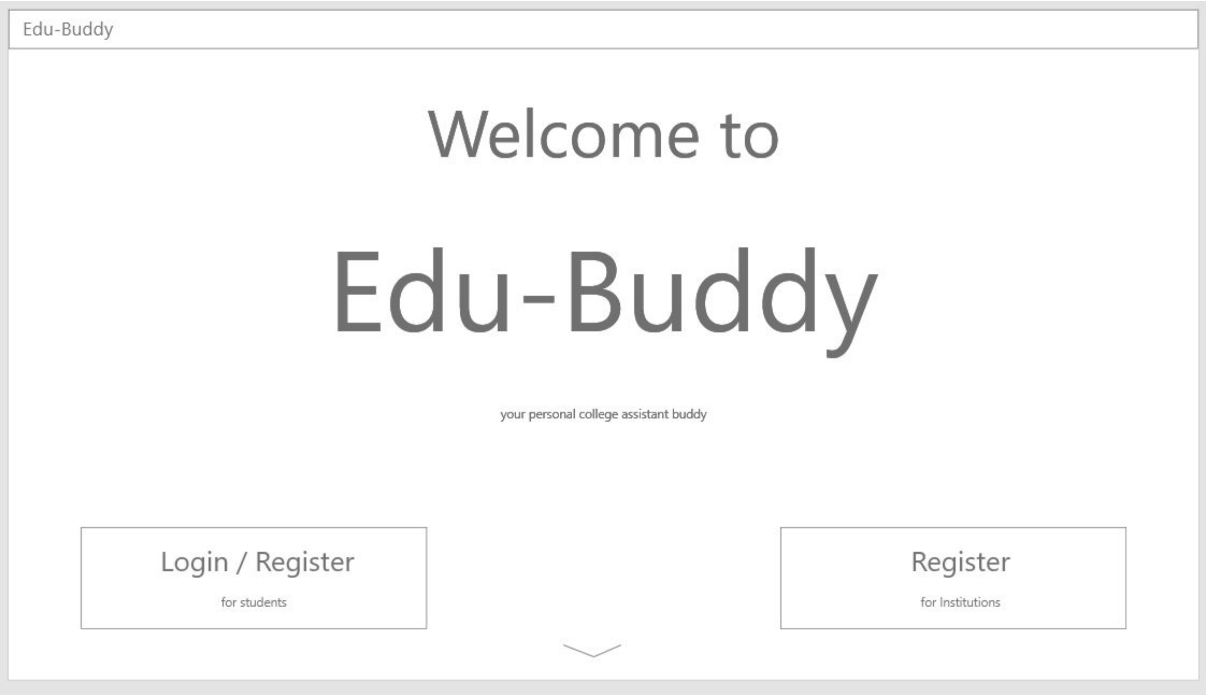
Google classroom API:
https://developers.google.com/classroom

GitHub API:
https://developer.github.com/v3/

Google Docs API:
https://developers.google.com/docs/api

DESIGN:


HIGH LEVEL SYSTEM DESIGN
database sharding on email column
uid | email | uniqid | cookie
Authentic-ation
users
users
users
Gateway
email service
service broker
(master/slave)
diff API's (classroom)
diff API's (calendar)
diff API's (github)
Load balancing consistent hashing
group service
gid | members
database sharding on gid column
Managing queues
chatbot
NoSQL database for storing info in JSON format
2020/10/18 16:53

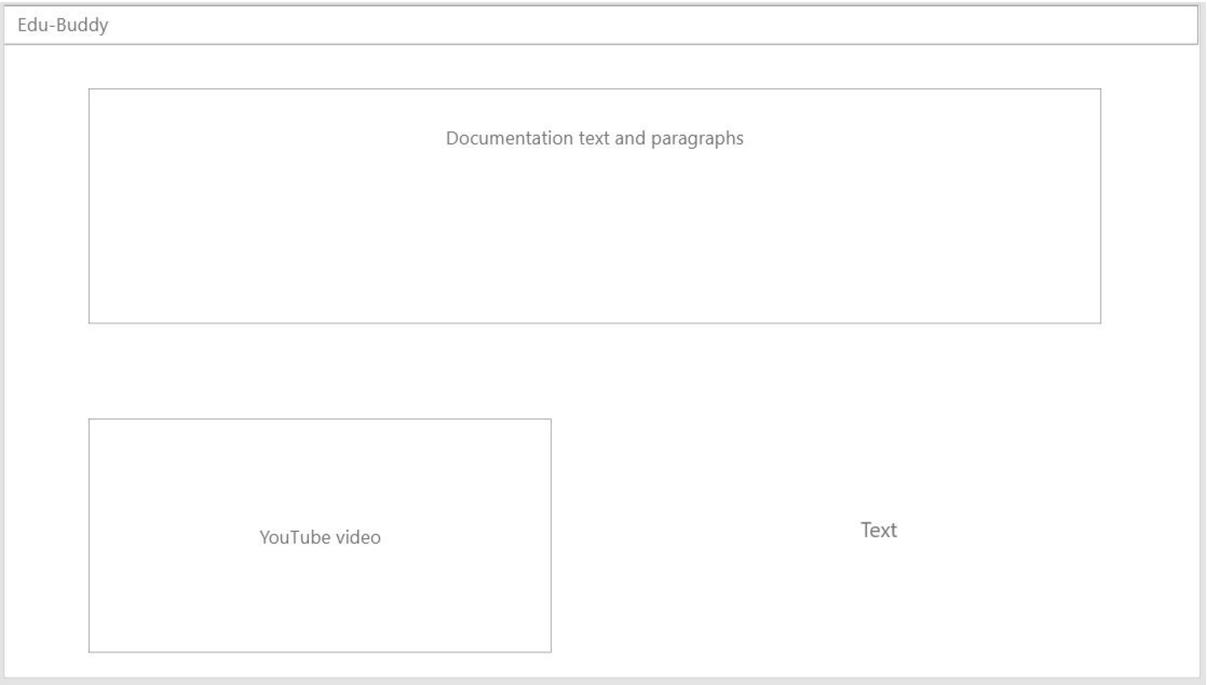


Edu-Buddy

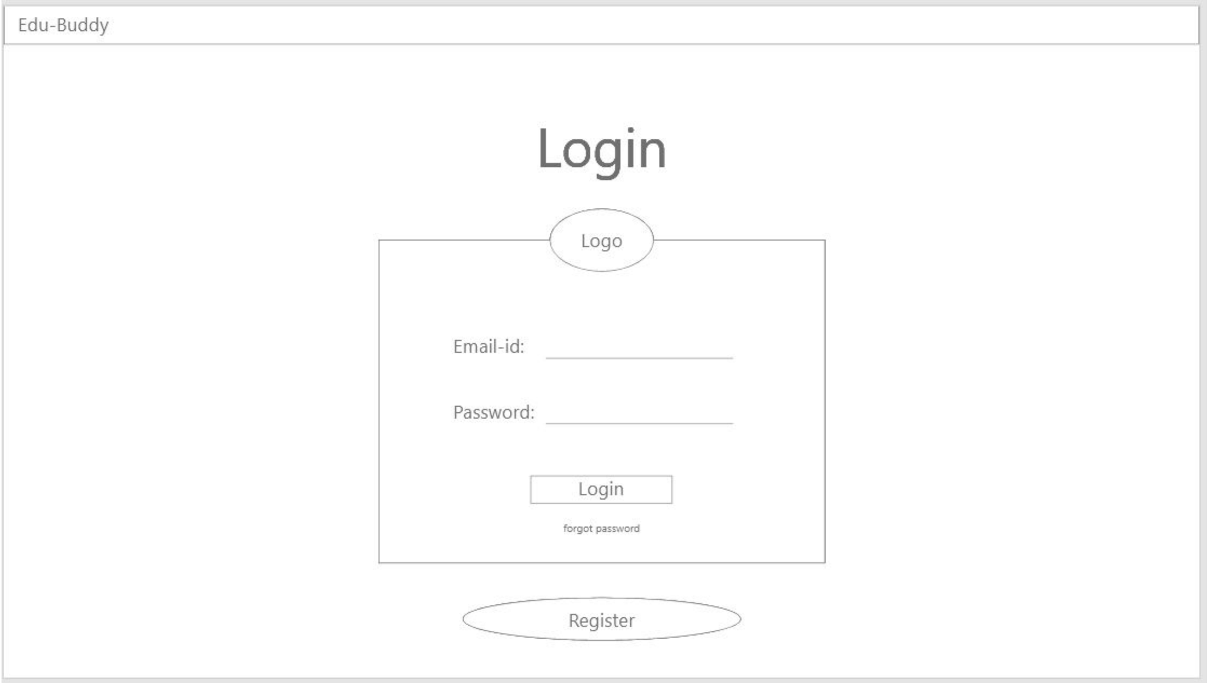
Welcome to

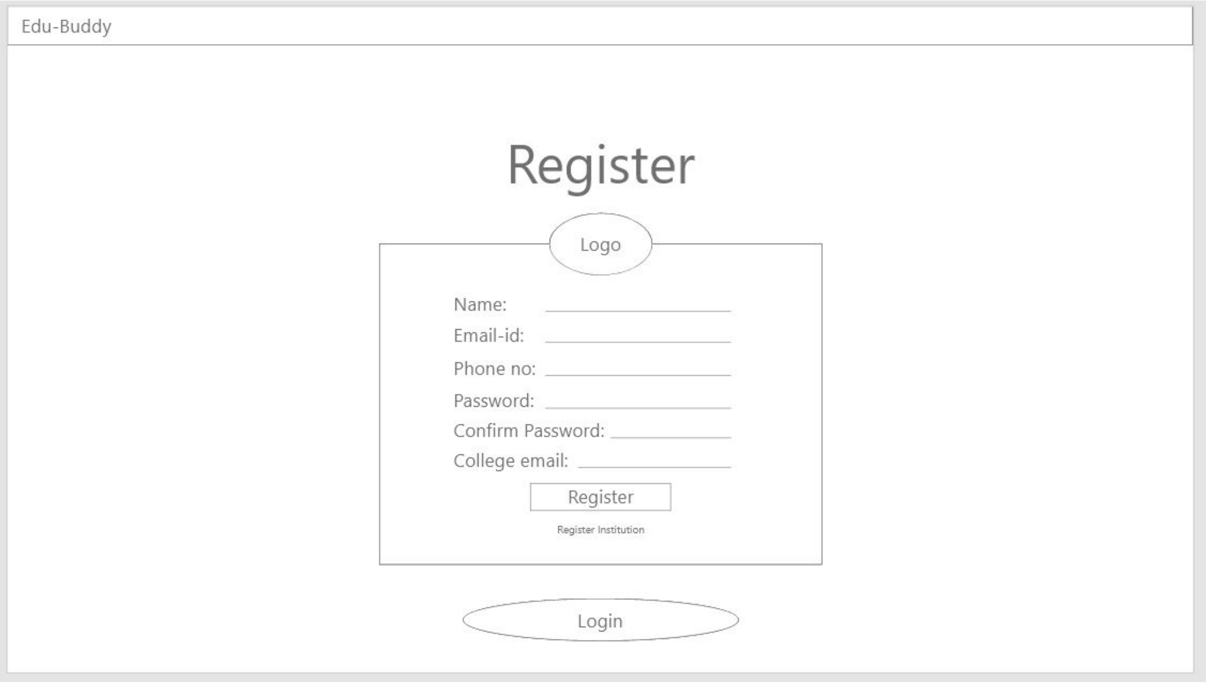
Edu-Buddy

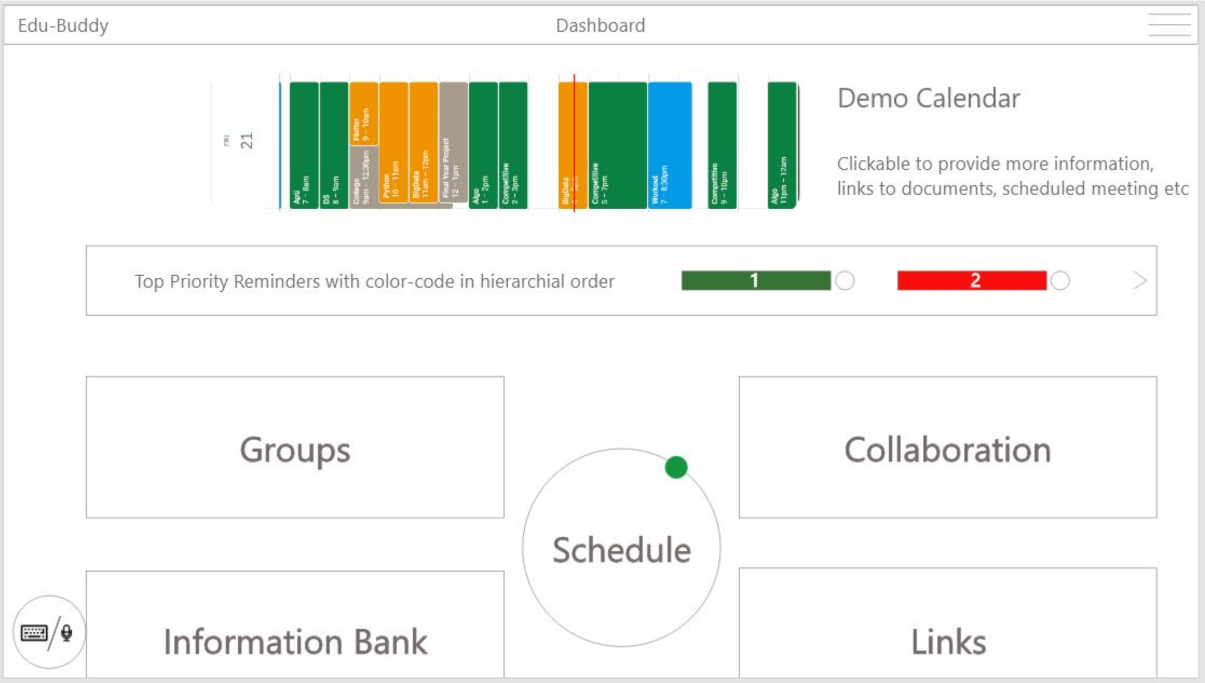
your personal college assistant buddy



Edu-Buddy


Edu-Buddy
Login
Logo
Email-id:
Password:
Login
forgot password
Register


Edu-Buddy
Register
Logo
Name:
Email-id:
Phone no:
Password:
Confirm Password:
College email:
Register
Register Institution
Login



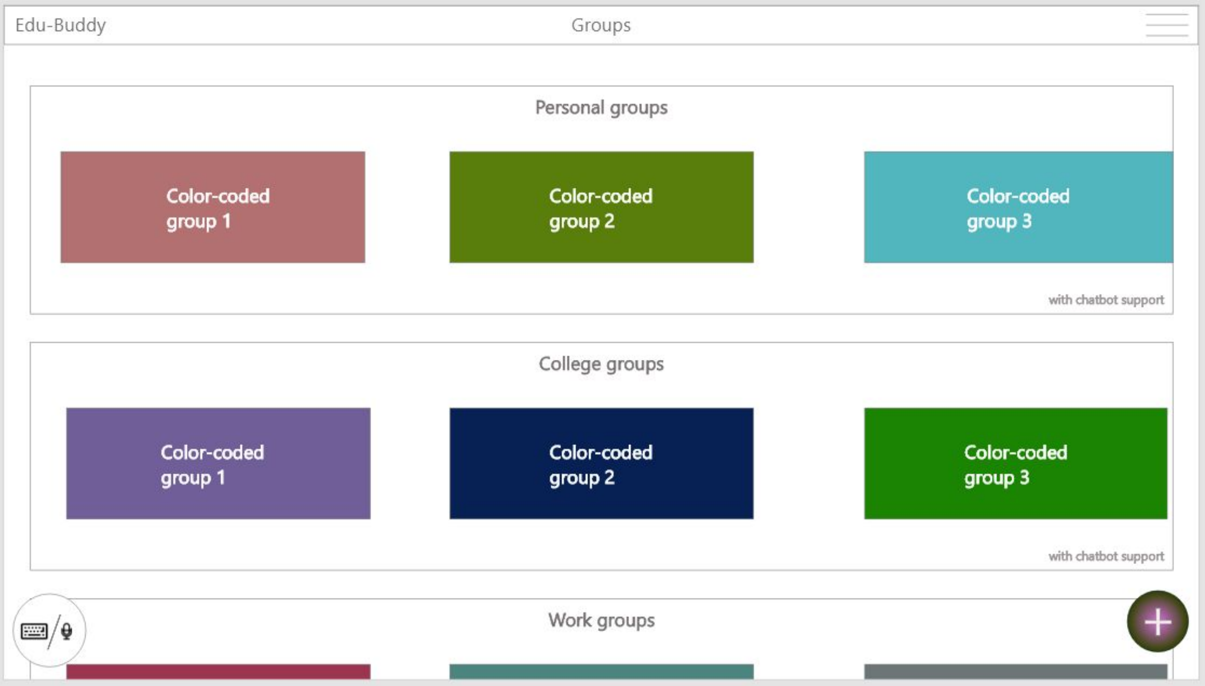
Edu-Buddy

Dashboard

Demo Calendar

Clickable to provide more information, links to documents, scheduled meeting etc

Top Priority Reminders with color-code in hierarchial order



Edu-Buddy

Groups

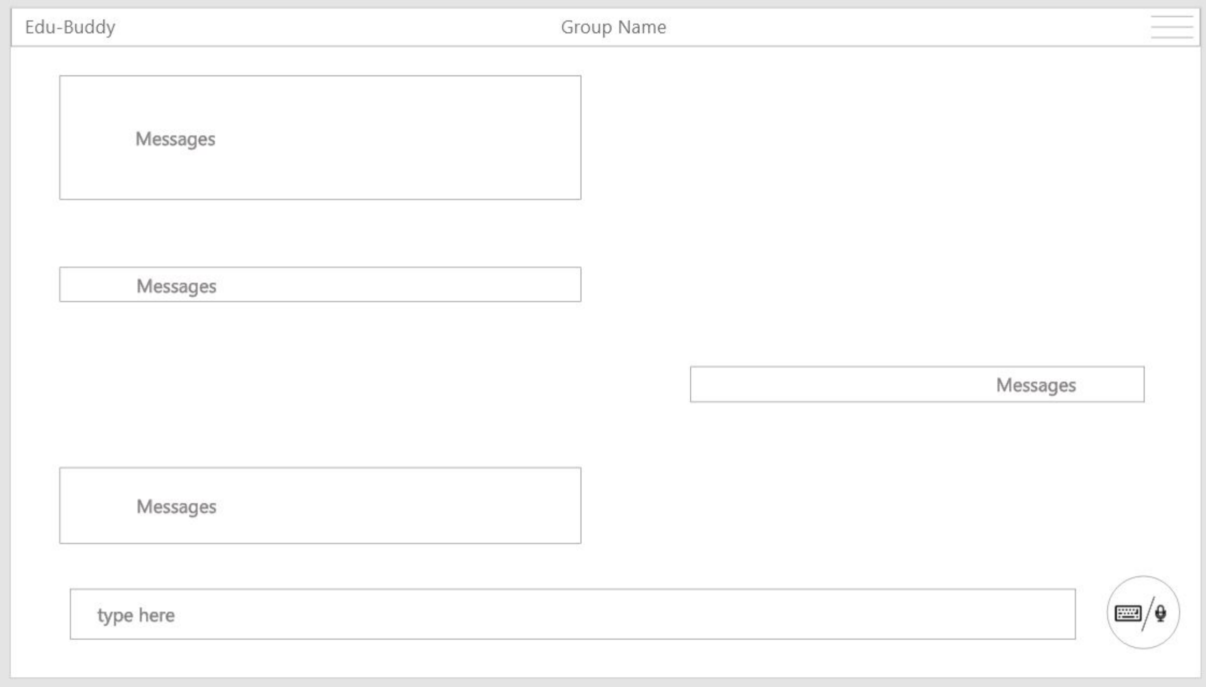
Personal groups

with chatbot support

College groups

with chatbot support

Work groups



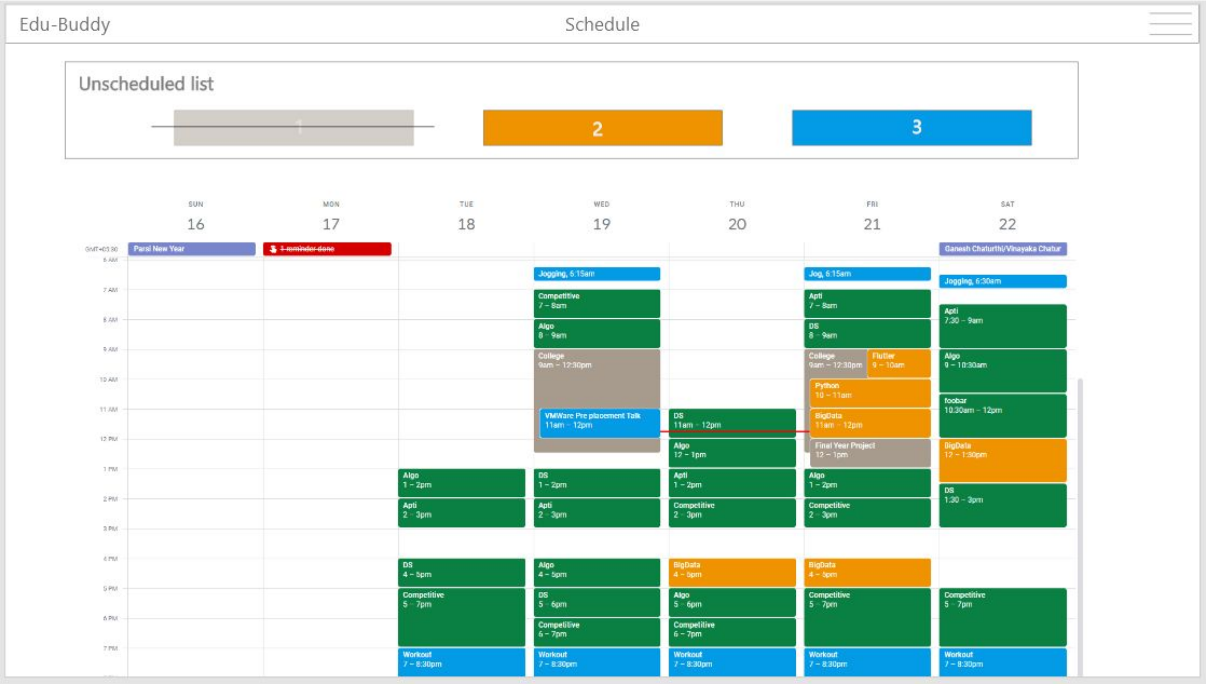
Messages

Messages

Messages

Messages

type here



Unscheduled list

~~1~~ 2 3

SUN 16
- Parsi New Year

MON 17
- ~~1 reminder done~~

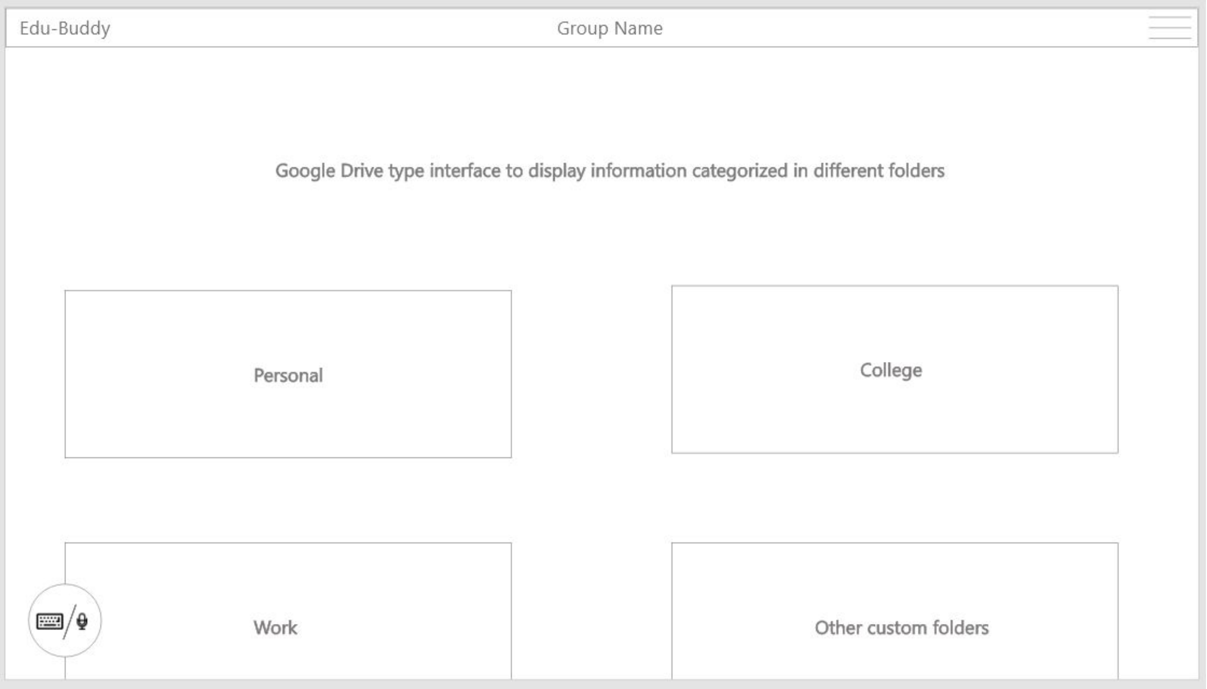
TUE 18
- Algo 1 – 2pm
- Apti 2 – 3pm
- DS 4 – 5pm
- Competitive 5 – 7pm
- Workout 7 – 8:30pm

WED 19
- Jogging, 6:15am
- Competitive 7 – 8am
- Algo 8 – 9am
- College 9am – 12:30pm
- VMWare Pre placement Talk 11am – 12pm
- DS 1 – 2pm
- Apti 2 – 3pm
- Algo 4 – 5pm
- DS 5 – 6pm
- Competitive 6 – 7pm
- Workout 7 – 8:30pm

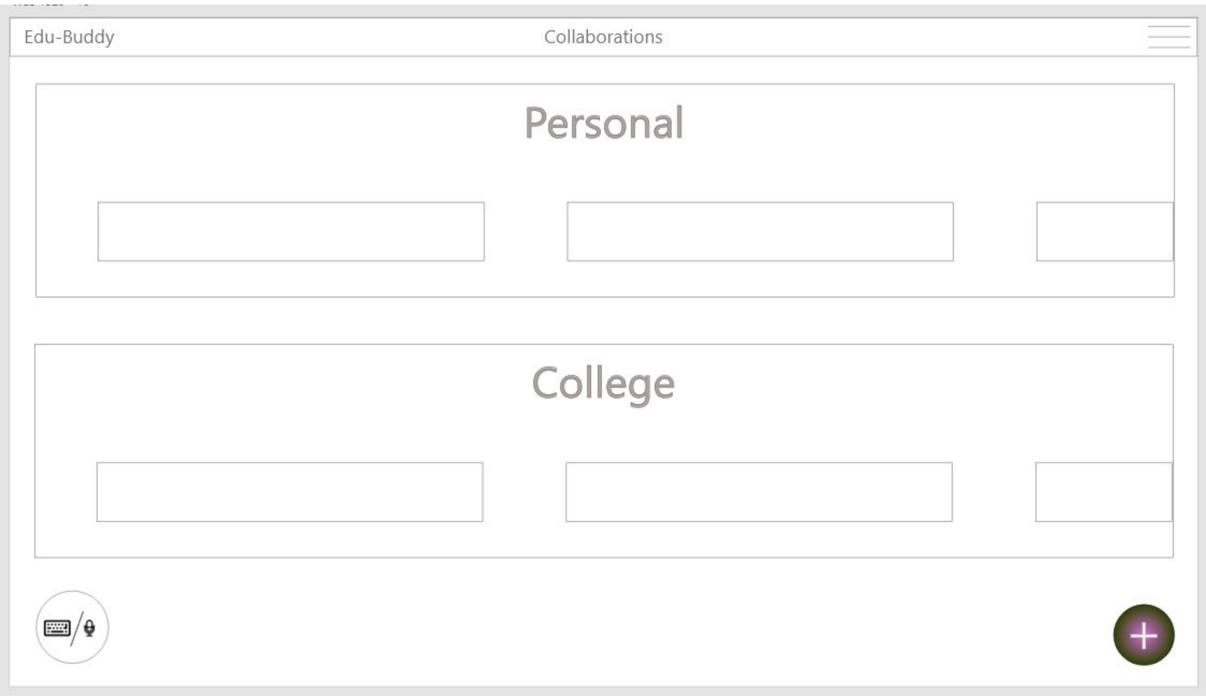
THU 20
- DS 11am – 12pm
- Algo 12 – 1pm
- Apti 1 – 2pm
- Competitive 2 – 3pm
- BigData 4 – 5pm
- Algo 5 – 6pm
- Competitive 6 – 7pm
- Workout 7 – 8:30pm

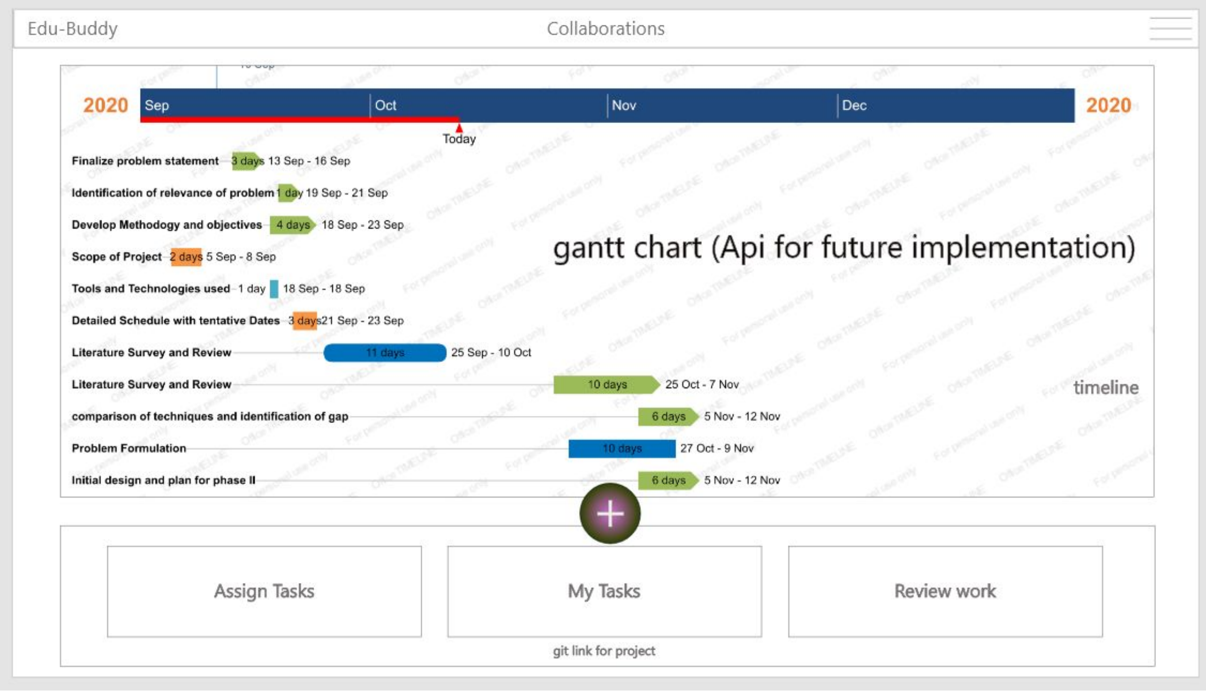
FRI 21
- Jog, 6:15am
- Apti 7 – 8am
- DS 8 – 9am
- College 9am – 12:30pm
- Flutter 9 – 10am
- Python 10 – 11am
- BigData 11am – 12pm
- Final Year Project 12 – 1pm
- Algo 1 – 2pm
- Competitive 2 – 3pm
- BigData 4 – 5pm
- Competitive 5 – 7pm
- Workout 7 – 8:30pm

SAT 22
- Ganesh Chaturthi/Vinayaka Chatur
- Jogging, 6:30am
- Apti 7:30 – 9am
- Algo 9 – 10:30am
- foobar 10:30am – 12pm
- BigData 12 – 1:30pm
- DS 1:30 – 3pm
- Competitive 5 – 7pm
- Workout 7 – 8:30pm



Google Drive type interface to display information categorized in different folders




Edu-Buddy
Collaborations
2020
Sep
Oct
Nov
Dec
2020
Today
Finalize problem statement 3 days 13 Sep - 16 Sep
Identification of relevance of problem 1 day 19 Sep - 21 Sep
Develop Methodogy and objectives 4 days 18 Sep - 23 Sep
Scope of Project 2 days 5 Sep - 8 Sep
Tools and Technologies used 1 day 18 Sep - 18 Sep
Detailed Schedule with tentative Dates 3 days21 Sep - 23 Sep
Literature Survey and Review 11 days 25 Sep - 10 Oct
Literature Survey and Review 10 days 25 Oct - 7 Nov
comparison of techniques and identification of gap 6 days 5 Nov - 12 Nov
Problem Formulation 10 days 27 Oct - 9 Nov
Initial design and plan for phase II 6 days 5 Nov - 12 Nov
gantt chart (Api for future implementation)
timeline
Assign Tasks
My Tasks
Review work
git link for project

Git link: https://github.com/roy-a2yush/Case-Study-Number-2---Aayush-Lal-Roy.git