Introduction

This document details the project plan for the development of "Edu Hub". This plan will include a summary of:

- how the system will function
- the scope of the project from the development viewpoint
- the technology used to develop the project, and
- Overall Description

A lot of times, we students end up searching/browsing multiple websites for webinars, hackathons etc. and it's time consuming. So here we propose a one stop solution Edu Hub where you can find various events hosted by colleges and individuals across the globe.

Objectives

- Help the users publicize their contests, hackathons, quizzes, webinars, workshops, and conferences so that interested users of the application can find and register to them.
  - Users can open the event to a particular group of people like
    - All, Students, Working professionals, etc.
    - All, Male, Female, etc.
  - Users can accept registrations for single/team participation.

- Help the users find the webinars/workshops, contests/hackathons which interest them. So that they won't have to go through the websites of each institute and company to know about the courses they offer.

Description

Customers:

The customers will be college students/scholars who are looking for webinars, hackathons etc. and also the event creators like college clubs.

Functionality:

- Users should be able to sign up with their email accounts.

- Users should be able to search for the events on the platform.

- Users should be able to publish and register for the events.

- Users should be able to keep track of the events they have registered for/ and publisher in the past.

- Google calendar integration to send reminders before the event.

Platform:

The application will be developed in Flutter to enable the creation of an iOS mobile app, and an Android mobile app and a web app.

The application will connect to a SQL server in the backend.

Development Responsibilities

The developers on the "Edu Hub" team will be responsible for writing all the code for the application, developing the database, and managing releases.
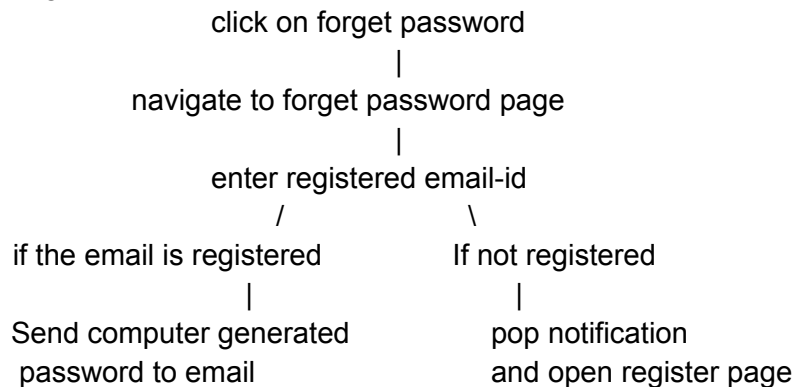
System Features:

Functional Requirements:

**Users can publish and register for the events on the portal.
Events can be webinars, workshops, hackathons.**

- Login/Signup for the portal. (Security question for Forgot password)

1. (path 1)Registered user -
   - Login page has entry fields email-id and password
   - If user forgets his password, follow path-3
   - User enters his registered email-id and password. Clicks submit.
   - If the login credentials match, the user will be navigated to the home page.
   - Else a notification of invalid email-id or password pops up in the login page. Here the user again follows path1 from the beginning.
2. (path 2)Unregistered user -
   - Click on register
   - Enter email-id, password, confirm password
   - Click submit -> navigate to home page
3. (path 3)If user forgets his password

<pre>
                        click on forget password
                                  |
                      navigate to forget password page
                                  |
                          enter registered email-id
                        /   \
        if the email is registered   If not registered
                   | |
        Send computer generated   pop notification
        password to email   and open register page
</pre>

- Search for the events
  - Users should be able to search for a specific event. A search box serves the purpose.
  - One should also be able to get all events under a category or filter out events. Example: ecommerce websites like Amazon allows users to get all products under a category like Clothes, cosmetics etc.
- Register for the events
  1. If the user is already logged in:
     - The user clicks on the 'Register' button.
     - The user details page(like name, email-id, etc) with email-id in non-editable form is displayed.
     - If any fields are blank, the user fills them.
     - Then the user clicks submit for successful registration.

  2. If the user is not logged in:
     - The user clicks on the 'Register' button
     - The login page opens and lets the user log in. User fills the log-in details.
     - If the login credentials are matched, the user details page (like name, email-id, etc) with email-id in non-editable form is displayed.
     - If any fields are blank, the user fills them.
     - Then click on submit for successful registration.

- Publish the events

After the user logs in:
- The user clicks on the 'Publish' button.
- Selects an event from dropdown list (Workshops/Seminars/Hackathons)
- Enters main description about the event (like Date,Platform,**Tag/Category**,Eligibility{Gender,Age,Qualification},Team Size,Google Calendar Event link)
- Adds a Google Form for the users to gain their credentials for future use(Or takes data from our User Registration database)
- Click the 'Publish Event' button.

- Google calendar integration
  - While registering for an event users will be given the option to add the event reminder to Google Calendar on their personal device.
  - Click the 'Add to Calendar' button.
  - The button will redirect the user to Google Calendar App where the user can add the event.
- Should be able to view all the registered events
  - When the user clicks 'Registered events', display all the events user registered with time date ordering. If user didn't log in we redirect to login page
  - We also need to have a cancellation option so that before deadline one can cancel the event if user change his mind
- Should be able to see published events and statistics (no:of people registered for the events)

- Ask user to select some preferences
  - The user clicks on 'Profile', then gets navigated to the profile page.
  - The Profile page has some fields like WebDev, AppDev, etc. We ask users to choose around 3-5 interests among them.
  - So when the user logins next, the new events related to selected preferences appear on the top of the events list.

Non Functional Requirements

- User friendly UI
- Good speed (Low latency)
- Availability

User Interfaces

- Front-end software: Flutter

- Database software: Firebase

Hardware Interfaces

- Android 4.1(APIlevel 16) or Higher.

- iPhone

- Both Mac and Windows operating systems through their default web browser

- Non-Functional Requirement

DataFlow Diagrams:

https://drive.google.com/file/d/1dXduCueWH1sfSIP-ZNMS3vORWkdJkMze/view?usp=sharing

Schema:

https://drive.google.com/file/d/1MT8blsAdiCIulyUa-Jig4SRN52Y5xf2O/view?usp=sharing