Shape

Description automatically generated with medium confidence

**ITECH 3209 – Project 2**

**“Agora”**

**Date: 12/06/2022**

**Shape

Description automatically generated with medium confidence**

A picture containing night sky

Description automatically generated

**User Manual and Product Installation**

**Team Agora:**

**Rijan Bahadur Pradhan: 30375525 (Full Stack Developer)**

**Nischal Tamrakar: 30372492 (UI/UX Designer)**

**Smriti Maharjan: 30373787 (Documentation)**

**Salonika Burlakoti: 30374076 (Documentation)**

**Siranta Koirala: 30375253 (Project Manager)**

|  |
| --- |
| **Revision**    **S.N. Date approved Approved by Description**  1.0 2022-04-26 Team Start User Manual  Members    1.1 2022-05-03 Team Added preface and glossary  Members    1.2 2022-05-12 Team Added Overview  Members    1.3 2022-05-19 Team Added technical specification  Members    2.0 2022-05-22 Team Added Installation guide  Members    2.1 2022-05-28 Team Added User Guide for customer  Members    2.2 2022-06-01 Team Added User Guide for Staff  Members    2.3 2022-06-02 Team Added User Guide for admin  members    2.4 2022-06-03 Team Final formatting of document  members |
| **Preface**  The main aim of this document is to give an overview of the project which is to create a QnA website for IIBIT from where students and teachers can log in with their account limiting the feature by looking at the type of account. The audience that we aim to attract is planning to have their questions answered ready and attempt the quiz that is set up by the teachers. The websites also should be able to be used by three different accounts. One is a student whose account is limited to solving the quiz and creating a discussion topic and answering them. The other one is a teacher whose account lets them create a new subject or quiz and vote on the best answers. Then, there is an admin whose account lets them remove, disable, or delete all the existing accounts. This handbook clearly states SCRUM. This user manual cites the project introduction, objectives, and project deliverables. The essential process model is also quoted in this project handbook in the project life cycle, together with its borders, interfaces, and project responsibilities. It also contains management, technical processes, and other non-functional requirements necessary for the future development of the project.  The user manual serves to provide information about various stages taken during the project lifecycle. With the assistance of clients, the project is developed by developers. The development team then works on the website. If this project requires a technological problem, the project team will examine and remedy the problem when necessary. |

Contents

[Overview 4](#_Toc106055217)

[What is AGORA? 5](#_Toc106055218)

[*Technical Specification and Minimum Platform Requirement* 5](#_Toc106055219)

[Installation Instructions 5](#_Toc106055220)

[1. How to Sign-Up 9](#_Toc106055221)

[*a)* *Steps to be followed by teachers:* 9](#_Toc106055222)

[*b)* *Steps to be followed by students* 11](#_Toc106055223)

[2. Admin’s role and functionality (What an admin should do!) 13](#_Toc106055224)

[3. How to Login 19](#_Toc106055225)

[4. What can student do in AGORA forum? 20](#_Toc106055226)

[5. What can teacher do in Agora? 24](#_Toc106055227)

# Overview

IIBIT is one of Australia’s progressive educational institutions providing quality education which is located in Sydney and Adelaide. Through IIBIT, Federation University delivers undergraduate and postgraduate programs. Students do a lot of research while doing their assignments and preparing for exams. There might be a lot of questions in the student mindset, so they happen to land on websites such as Quora which is basically a AGORA forum and find answers. That is why client IIBIT wants their own AGORA forum for their students which would be similar to Quora or other AGORA forums. For this, the first step for us is to build a design or wireframe of the website. As this is an AGORA forum both students and teachers will be using this website, so logins are required for these two user groups.  After login, they shall get access to the website, but the functionalities would be different. Admin (Teacher) shall have special functionalities which are designing quizzes, removing questions, and banning or removing students. Other functionalities such as sign-up, login/logout, updating personal information, posting questions, replying, adding media to post, and voting for the best answers will be the same but these two user groups will have two different databases to store their information.  Through this project we will be able to make direct communication between students and teachers at the same time IIBIT will help students to learn more about various subjects.

What is AGORA?

AGORA is a Q&A forum for both students and teachers as they will be using this website, so logins are required for these two user groups.  After login, they shall get access to the website, but the functionalities would be different. Admin (Teacher) shall have special functionalities which are designing quizzes, removing questions, and banning or removing students. Other functionalities such as sign-up, login/logout, updating personal information, posting questions, replying, adding media to post, and voting for the best answers will be the same but these two user groups will have two different databases to store their information.  Through this project we will be able to make direct communication between students and teachers at the same time IIBIT will help students to learn more about various subjects.

# *Technical* **Specification and Minimum Platform Requirement**

Application has been built for web browsers and needs **Windows 7,8 or the latest version** of Windows.

# Installation Instructions

* Open Visual Studio code in the code editor.

A screenshot of a computer

Description automatically generated with medium confidence

Fig 1.1: Visual studio App

* + - Click file>open Folder>Choose Agora>Click Select Folder
    - Go on to google and download Node.js

A screenshot of a computer

Description automatically generated with medium confidence

Fig 1.2: Node js installation process

* + - After loading the given file in vs studio, go to the terminal type, “npm install” and after that type the word “npm run start”

A screenshot of a computer screen

Description automatically generated with medium confidence

Fig 1.3: how to start the website

* After running “npm run start” on terminal, the website will automatically pop up on the google chrome in the port “localhost3000”. It will go straight to the login page.

**Graphical user interface, application

Description automatically generated**

Fig 1.4: Login page of Agora

# How to Sign-Up

A login page would be the first thing we will see. When the signup link is clicked the below picture would pop out.

**Graphical user interface, application

Description automatically generated**

Figure 2.1: how to sign-up

## *Steps to be followed by teachers:*

* Type all the details that are required in this sign-up page form. The fields include name, email address, your phone number, your address, date of birth and password which should be very strong.
* Teacher must tick the teacher checkbox otherwise teacher will be logged in as a student. An administrator would verify the given role and give access to the teachers.
* After submitting the form, teacher will have to send an email to admin regarding the role. Agora does not have the automatic feature for this process, so admin must manually change the role of teacher upon getting the email from teachers and confirming with HR.

Graphical user interface, application

Description automatically generated

Fig 2.2: Teacher sign-up

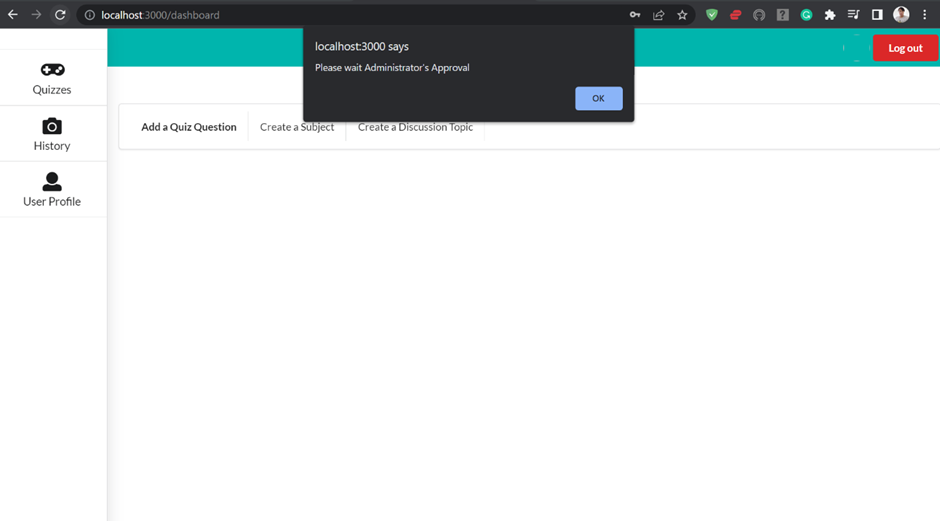


Fig 2.3: Admins approval message pop-up 1

## *Steps to be followed by students*

* Type all the details that are required in this sign-up page form. The fields include name, email address, your phone number, your address, date of birth and password which should be very strong. that
* Students must not tick the teacher checkbox as it requires an administrative approval. By not ticking a teacher checkbox, students can automatically log in with limited functionality.

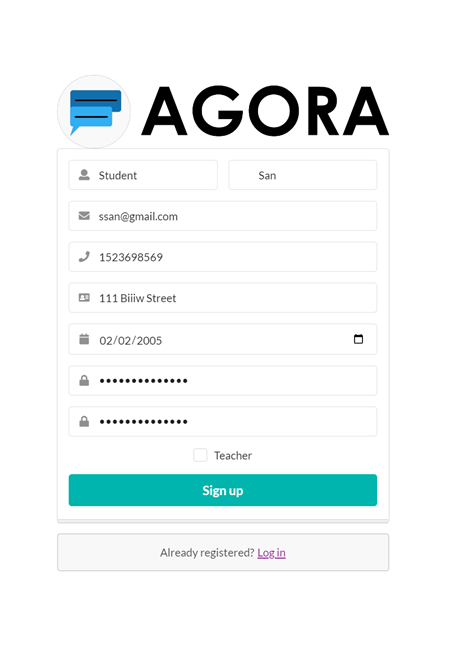


Fig 2.4: Student’s Sign-up

*Note: It is required for both teachers and students meet all the requirements of password given by the website, otherwise, it would not register the user and outputs a message saying, “Your password doesn’t meet the requirement.”*

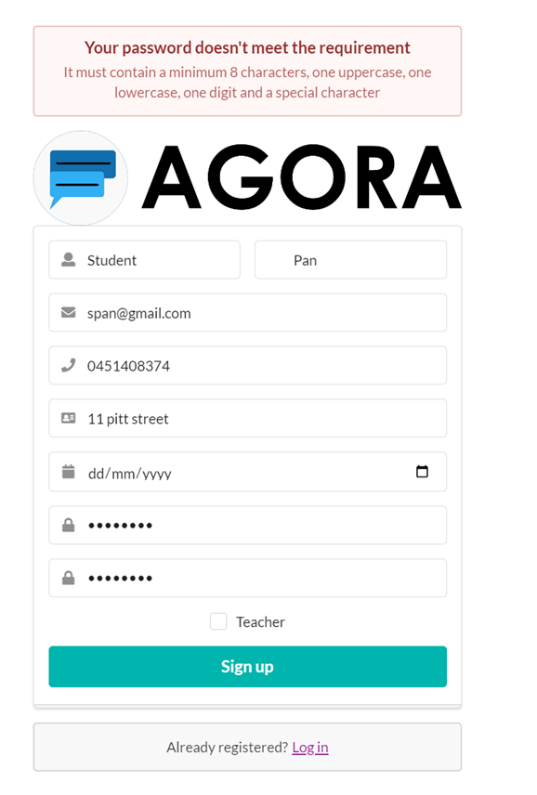


Fig 2.5: Password requirement not met

# Admin’s role and functionality (What an admin should do!)

We use firebase console as our admin panel. You will need a google account to do this. Agora team has already given access to the client in firebase google account. So, client just needs to login with their special privileges account to access the database.

1. To login to fire base:

* Admin should search for Firebase in a seach engine by typing: [firebase.google.com](https://firebase.google.com/)

Graphical user interface

Description automatically generated

Fig 3.1: Firebase webpage

b.) To grant access to teacher and manage database

* After that admin can go to project settings/users and permissions. Administrators will be able to see their accounts and their roles.

**Graphical user interface, chart

Description automatically generated**

Fig 3.2: Firebase dashboard

**A screenshot of a computer

Description automatically generated**

Fig 3.3: Project Dashboard

* Admin will be able to see the database after clicking the authentication.

**Graphical user interface, application

Description automatically generated**

Fig 3.4: Database through firebase

* From here admin will be able to reset, disable and delete account.

**Graphical user interface, text, application

Description automatically generated**

Fig 3.5: admin feature

* After clicking firestore database, click users and three buttons over the top just as shown in the picture and write a keyword “email” in it and hit apply.

**Graphical user interface, text, application

Description automatically generated**

Fig 3.6: How to give permission 1

**Graphical user interface, application

Description automatically generated**

Fig 3.7: How to give permission 2

* A list of emails can also be seen by admin who have signed up would come.

**Graphical user interface, text, application

Description automatically generated**

Fig 3.8: How to give permission 3

* As an Administrator, you can see who has logged in as a student and teacher. An administrator can change the role anytime, from “students to “teachers” and vice-versa. Also, from the picture, approval can only be given i.e “Yes” or “No” by the administrator to access the teacher’s account. All the post, quizzers, subject, topics, users are created in a website and gets saved in this cloud firestore database which only admin have the control of. An administrator can change the things that are arrowed by red circles.

**Graphical user interface, application

Description automatically generated**

Fig 3.9: User database

# How to Login

* For both teachers and students, the process is same. Users basically must enter their email address and password they have used during the sign-up of their account to login into the system.

****

# What can student do in AGORA forum?

1. Students can create a discussion topic to discuss in the AGORA forum. To do so, click “create a discussion topic” and a pop-up window will appear where students have:

* Add Topic title
* Chose topic subject from drop down list
* Add topic description
* And if they have any medias, they can choose by clicking the choose file button

After completing the above steps students will be able to add discussion topic to the forum.

**Graphical user interface, text, application

Description automatically generated**

Fig 4.1: Add discussion topic(student)

1. Students can see the topics created on the website and will be able to engage in them. They can go to a particular topic they like. They can add, like, or dislike to the post.

**Graphical user interface, application

Description automatically generated**

Fig 4.2: Like and dislike (Student)

1. Students can visit a particular topic they like and create a post about that topic. Students can also like and dislike posts inside the topic.

**Graphical user interface, text, application, email

Description automatically generated**

Fig 4.3: Student reply

1. Students can click the “quizzes” button and attempt the quiz. The quiz can only be created by teachers and students can only attempt the given quiz.

**Graphical user interface, application

Description automatically generated**

Fig 4.4: Quiz Dashboard

1. Students can go inside that quiz and try to attempt the quiz.

**Graphical user interface, text, application, email

Description automatically generated**

Fig 4.5: Attempt Quiz (Student)

1. Students can also see what they’ve done in their account. Anything that they do will come up in their history.

**Graphical user interface, table

Description automatically generated**

Fig 4.6: History (student)

1. Students can click the link of “User Profile” and update their username, phone number, address, and DOB. They can also change their picture. Students can click the update button to update their profiles.

**Graphical user interface, text, application, email

Description automatically generated**

Fig 4.7: Update Profile (Student)

# What can teacher do in Agora?

* After teachers logs in, Teacher will be able to see the posts from students and other teachers as well.
* Teachers have two extra functionalities. Add a quiz question and create a subject.

**Graphical user interface

Description automatically generated**

Fig 5.1: Teacher Dashboard

* Teacher can create a subject name and a subject for a student to choose and learn when they are attempting quiz or discussing topics.

**Graphical user interface, application

Description automatically generated**

Fig 5.2: Create Subject

* Teachers can also create quiz questions. They can choose from a vast subject that they’ve created and post a quiz with one correct answer

**Graphical user interface, application

Description automatically generated**

Fig 5.3: Create quiz (teacher)

* Teachers can also pin students’ comments so that the students know which is the official answer or not.

**Graphical user interface, text, application, email

Description automatically generated**Fig 5.4: Pin answers and like/dislike the answers

* Also, a green tick mark appears on a topic that teachers have pinned.

**Graphical user interface

Description automatically generated**

* Teachers can also add extra quiz questions to the quizzes they’ve already created. This functionality is only limited to teachers.

**Graphical user interface, application

Description automatically generated**

Fig 5.5: Add extra question to quiz

* Besides, all the above-mentioned functions Teachers have the same functionalities aligned with students. Teachers can see their history of what they’ve done in their account.

**Graphical user interface, table

Description automatically generated**

Fig 5.6: History (teacher)

* Teacher can see their user profile and update it by clicking to the Update profile from navigation bar.

**Graphical user interface, text, application

Description automatically generated**

Fig 5.7: Update Profile

* Teacher can also create a discussion topic. It can be seen by anyone with an account in Agora Website. They can attach anything image related to the discussion topic.

**Graphical user interface, text, application

Description automatically generated**

Fig 5.8: Add discussion (Teacher)

* Teachers can also post the reply to the question. They can attach any related file as well.

**Graphical user interface, application

Description automatically generated**

Fig 5.9: Reply post