



PROJECT PROPOSAL, DEPLOYMENT AND DELIVERY MODEL CRITICAL ANALYSIS AND RESPONSE

CSCI 5409 – Cloud Computing

Submitted To: Prof. Robert Hawkey

Submitted On: Feb 13, 2022

Submitted By: Group 31(Greencloud)

Group Members:

Sumit Singh [B00882103]

Rama Mohan Vishnu Guturu [B00871849]

Aditya Deepak Mahale [B00867619]

Web Application Name: Education4every1

1. Overview

During COVID times, the world shifted to online mediums to educate its audience [1]. Whether it's YouTube or other video streaming services, every platform witnessed a surge in subscribers/users for their educational content [1]. However, communication is limited to a few users due to language constraints.

Accessible education is indispensable in today's world. It is imperative to make the audio available in different languages to reach out to wider audience. The purpose of this project is to build a web application that translates English language text to a different language (Speech).

There are plenty of video streaming services online providing transcripts in different languages. However, the audio is not available in other languages.

2. Features Overview

Web:

1. Sign Up (Authentication and Authorization)
2. Login
3. Profile page (Edit profile page)
4. List of language translation options
5. Audio player

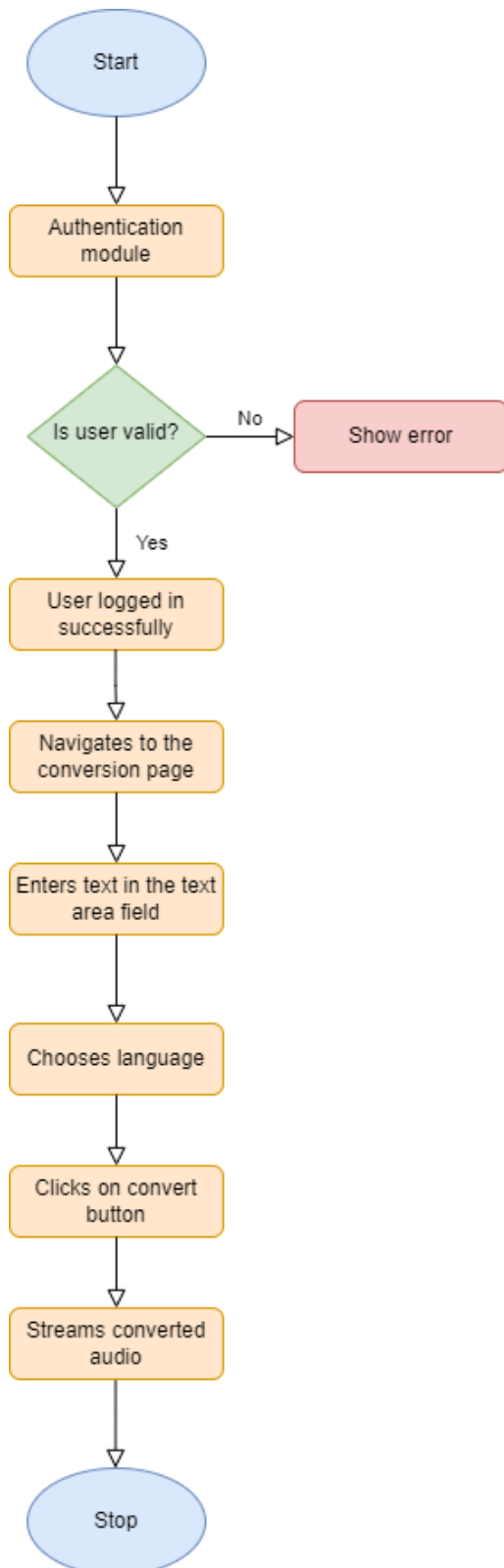
Cloud:

1. Static file (Image/Audio) storage using S3
2. Network and Security using VPC
3. Resilience using redundant VMs.

3. Tech Stack Planned for the Implementation

Since we have the freedom to choose the technology stack in this project, we decided to go for the stack in which we are all comfortable as a team. For the front-end, we have chosen ReactJS. For the back end, we decided to go for Node.js/Express.js.

4. Description (Website Flow)



Conversion

A student should log in to be able to access the content on the website.

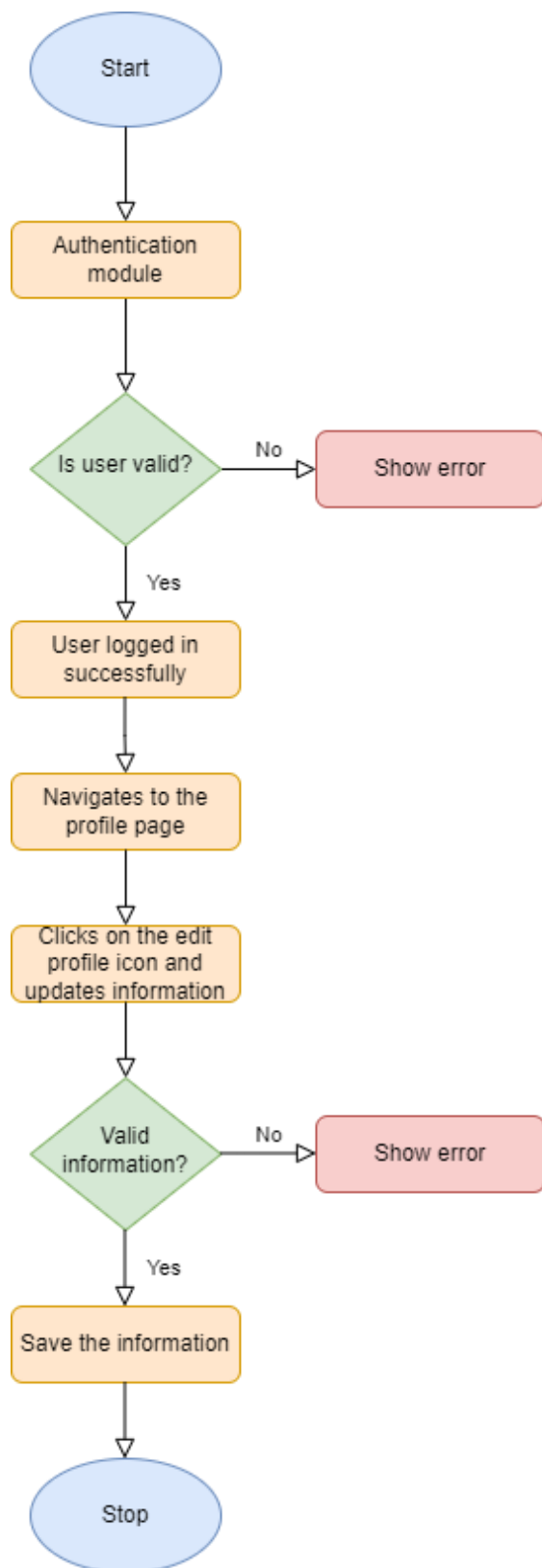
A registration page would allow the users to get authenticated on the website.

A page will appear with a text field for entering audio transcripts.

There will be an option for the user to choose the language.

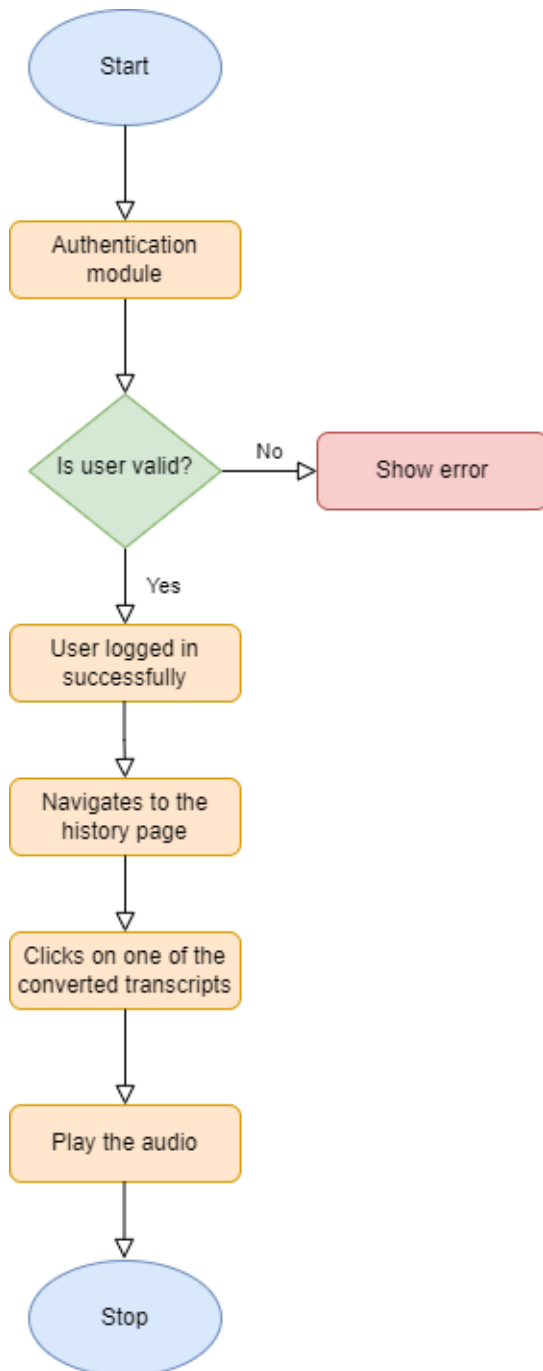
Clicking on the "Convert" button will start the translation and conversion process.

An audio player will pick up the audio file from S3 to stream the translated content.



Edit profile

Users log in using their credentials on the website.
Navigates to the edit profile page.
Click on the edit button and get a modal for editing details.



Conversion history:

A logged-in user can view the history of conversions.
Clicking on the older conversions should stream previously converted files.

Cloud Deployment Model: Public cloud (AWS)
Cloud Delivery Model: IaaS + FaaS

5. AWS Services

Table 1

Mechanism	Services	Description
Compute	EC2, Lambda	<p>We will be using EC2 instances (4) for multiple purposes:</p> <ul style="list-style-type: none">• Host our frontend ReactJS project on an EC2 instance.• Use an EC2 instance for hosting our backend REST API project.• Installing a database (MongoDB/PostgreSQL) on an EC2 instance.• Use bastion host for accessing VMs on a private subnet [2].• Use the lambda service as a trigger to store the audio file in the s3 bucket.• Use lambda to stream audio files from the S3 bucket to the user.
Storage	S3	<ul style="list-style-type: none">• Using S3 for storing converted audio files.• Utilize S3 to host images for our front end.
Network	VPC	<ul style="list-style-type: none">• VPC to provide security to our entire setup.• Use private subnets to host sensitive data in the database.• Configure security groups and Network ACLs to add a layer of security.• Use NAT gateway for getting connectivity.
General	Polly, Translate	<ul style="list-style-type: none">• Use Polly to convert text to speech• Use AWS Translate to translate text to a different language.

6. Project Plan

We'll be conducting two-week sprints to manage our project tasks. Before the sprint, we'll do backlog refinement using an online planning poker game.

In the sprint planning, we'll decide what to include in the sprints.

We'll use the GitLab board to keep track of our agile stories.

Timeline:

14 Feb – 21 Feb

- Discuss the basic deployment architecture.
- Explore AWS Polly and Translate.
- Start studying AWS CloudFormation.
- Practice provisioning EC2 instances, subnets, S3 on AWS Academy.
- Practice interaction with the AWS using APIs (boto client).

22 Feb – 28 Feb

- Prepare design for the frontend.
- Initialize the frontend repository.
- Initialize the backend repository.
- Brainstorm backend design and architecture.
- Design database for the backend part by preparing Entity – Relationship Diagram for the project
- Continue studying AWS CloudFormation.
- Prepare “Mechanism Critical Analysis and Response” report.

1 Mar – 7 Mar

- Build a VPC from scratch with public and private subnets.
- Deploy static files on S3.
- Start building front end for the project using dummy services using ReactJS.
- Deploy a simple application with the backend in the private subnet and frontend in the public subnet.
- Continue studying AWS CloudFormation.

8 Mar – 14 Mar

- Work on the backend code.
- Integrate AWS Polly and Translate services.
- Deploy a simple application using cloud formation.

15 Mar – 21 Mar

- Prepare “Architecture Critical Analysis and Response” report.
- Integrate front-end and back-end.
- Continue working on the backend code.
- Deploy the application on the test environment using AWS console.

22 Mar – 6 Apr

- Prepare “Security & Business Considerations Critical Analysis and Response” report.
- Start working on the final report.
- Start working on the video presentation.
- Perform final integration testing.
- Prepare CloudFormation template for the entire project deployment.

7. Critical Analysis

1. Describe the deployment model for the cloud-based software system you intend to build.

We'll be only using AWS cloud to deploy our project on the cloud. Hence, we'll be using public cloud model for our deployment.

2. Why did your group pick this deployment model?

AWS offers a plethora of services for building a wide variety of websites. All the resources that we need in this project are available on AWS. Hence, we decided to go for the public cloud deployment model (AWS).

3. What are the benefits of your deployment model?

Interaction between services in a hybrid-cloud/multi-cloud model could get highly complicated. On AWS, all the services are well integrated. Hence, various complexities and issues involved in the project could be resolved easily with the model.

4. What are the drawbacks or risks of your deployment model?

Since the project is on the public cloud, there's a concern about privacy security due to multitenancy.

5. Describe the delivery model for the cloud-based software system you intend to build.

We'll be using IaaS + FaaS delivery model for our project. To get a better understanding of how the cloud works, we are planning to build our project using infrastructure services provided by AWS. E.g. We'll build VPC from scratch. We'll install the database cluster on EC2.

6.Explain the tradeoffs of this model in comparison to other delivery models (IaaS, FaaS, PaaS, SaaS). For example, how much control do you have over your cloud provided infrastructure?

IaaS provides complete control over services. We can configure the resources with flexibility. However, it is time-consuming as compared to other delivery models such as PaaS, FaaS, and SaaS.

8. References

- [1] B. Agencies, "YouTube sees surge in subscriber base, views due to Covid-19 lockdown", *Business-standard.com*, 2022. [Online]. Available: https://www.business-standard.com/article/technology/youtube-sees-surge-in-subscriber-base-views-due-to-covid-19-lockdown-120042100710_1.html. [Accessed: 09- Feb- 2022]
- [2] "Linux Bastion Hosts on AWS - Quick Start", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/quickstart/architecture/linux-bastion/>. [Accessed: 10- Feb- 2022]