INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

SSD PROJECT REPORT

SURVEY BUILDER AND TRANSCRIPT GENERATOR

FOR

IIIT EMPIRICAL RESEARCHERS

TEAM 7 - BRAINSTORMERS

LINKS

Project GitHub Repository URL: https://github.com/Rito2323/SSD-project-Team7

Demo Video Link : https://drive.google.com/file/d/10pXTQWATW440Avz6lKIogEShnfUwnmbc/view

Guided by:

Saianirudh Karri

Arvind Narayanan(TA)

Submitted by:

Shaon Dasgupta (2021201068)

Vishal Mani Mishra (2021201050)

Yaswanth Haridasu (2021201032)

Gargi Dwivedi (2021201076)

Ritabrata Podder (2019900031)

INTRODUCTION

About The Project: -

This Project consists of two parts, survey builder and Transcript Generator. In the Survey Builder, the following tasks can be performed: -

1) Researchers can log in to create, modify and upload a survey form. This survey form link can be shared among participants.
2) Participants can log in and answer the questions asked in the survey form.
3) Responses of participants can be seen by researchers whenever he/she wants.

In the Transcript Generation part, after login, the researcher can upload the video or audio file in the respective field and generate the transcript file and download it in the end.

REQUIREMENTS

1. Stakeholders

- Ph.D. Students of IIIT, Hyderabad.
- MS Students of IIIT, Hyderabad.
- Research Faculties of IIIT, Hyderabad.
- Focussed Interns
- Other survey participants.

2. Roles and Responsibilities of Stakeholders

- MS and Ph.D. students and faculties (Survey Developers)

  They will have the role of the survey developers and will use the survey builder to create and publish new surveys. Also, they may use the Transcript generator to generate the transcript of an interview. After publishing the form, they can view the responses in the response section.

- Focussed Interns and other survey participants

  They will fill the published survey and submit their responses which would be recorded and viewed by the survey developer.

WORKFLOW

Survey Developer:

Workflow 1: Survey Builder

The developer will be able to see the Entry Dashboard UI. On selecting the Survey Builder option in the Dashboard, the user would be navigated to the survey builder page.

In the survey builder page, the developer would be able to select the type of question, based on the response type and add the question to the survey question list.

After the questions are selected, the user could preview his questions.

If the preview is satisfactory the user can save and publish the survey. The survey form data will be converted to JSON format and stored in the DB. Also, publishing the survey would generate a survey link that the developer can share with the participants.

Also, developers can delete existing surveys or update existing surveys.

Workflow 2: Response Viewer

After creating and publishing the survey, the developer will be able to view and analyze the responses by navigating to the Responses UI from the dashboard.

Under the Responses UI, all the participants' responses in each survey created and published by the developer could be viewed.

Workflow 3: Transcript Generator

From the dashboard, the developer should be able to navigate to the Transcript generator to generate a transcript of an interview recording.

In the Transcript generator, the developer should be able to upload an audio/video file of any format, containing an interview recording and then generate and download the transcript file.

Participant:

Workflow: Survey fill up and submission

The participant who has access to the survey link should be able to fill it up and submit the responses.

The responses submitted should be stored in the DB.

TECHNICAL DETAILS

Survey Builder Frontend

Technology Used

React JS, Html, CSS, JS, Bootstrap

Description

Using ReactJS and other technologies mentioned above, we have built the front end of the survey builder as well as the front end of the transcript generator. This will simplify the user to create a good survey and look at the responses.

Functionalities Implemented:

1) Create New Survey
2) Add questions to the survey. The questions can be of types:
    a) Text based
    b) Single Select
    c) Multi Select
    d) Matrix Likert
3) Edit / Delete / Preview questions in existing surveys.
4) View / Delete existing surveys.
5) Get links for surveys that are created. These links can be shared with the participants of the survey.
6) View Responses made by the participants for each of the surveys.

Survey Builder Backend

Technology used

Mongo DB, Node js, Rest service

Description

-With node js, we are making a connection to MongoDB; to insert /fetch/update/delete documents from Frontend
-For the survey, we are using survey Schema
-For response, we are using response schema

Survey Schema

Survey schema will be in the form:

```
{
    "SurveyNo" : "<>",          // to uniquely identify a survey
    "SurveyTitle" : "<>",       //any title of survey
    "CreatedBy" : "<>",         //logged in user who is creating the survey
    "Questions" : [{
                    "QuestionType" : <>,    //question type is 1 for text based, 2 for single
                                            //option correct, 3 for multi option correct, 4 for
                                            //matrix likert
                    "QuesionNo" :<>,
                    "QuestionText" : "<>", //question
                    "Options" :[                //options is list of option ; list be null if text based
                                            question, but multiple objects in case of question type as 2
                                            3 4
                            {
                                "name":"<>",    //name will store the option ,and levels
                                                will store scale only for matrix likert
                                                question type else will be null
                                "levels" : []           }       ]       }       ]       }
```

Survey schema has various rest endpoints to interact with Mongo DB:-

POST surveys from frontend to backend
/add_survey

GET all surveys from backend to frontend
/surveys

GET surveys by logged in user and survey number from backend to frontend

/surveys/:devmail/:SurveyNo

GET surveys by logged in user from backend to frontend
/surveys/:devmail

DELETE survey
/delete/survey/:SurveyNo

UPDATE a survey
/update/survey/:SurveyNo

Response Schema:

Response schema will be in the form:

```
{"SurveyNo" : "<>",       // to uniquely identify a survey
        "CreatedBy" : "<>",         //logged in user who is creating the survey
        "Participant":  "<>",         //particpant who is actually giving the survey
        "Response" : [     //Response will be list depending on no of questions in the  survey
        {         "QuestionNo" :<>,
                  "QuestionText" : "<>",
                  "Ans" :{}           //key value pairs of question no. and answers.

 }       ]         }
```

Response schema has various rest endpoints to interact with Mongo DB:-

POST responses from fronted to backend
/add_response

GET all responses from backend to frontend
/responses

Transcript Generator Backend:

1. In the Transcript generator, the developer can upload an audio/video file of any format, containing an interview recording and then generate and download the transcript of the recording in a text file.

Implementation:

1. The main code for converting the audio file to the text is written in python language.
2. The transcript generator is implemented as follows:
   - 2.1. Front end:
     - 2.1.1. Developer uploads a video/audio file from the browser.
     - 2.1.2. The video/audio file will be sent to the back-end(express js server) in the form of a Buffer Object.
   - 2.2. Back end:
     - 2.2.1. In the back-end code, the Buffer Object is converted back to the video/audio file.
     - 2.2.2. The back-end server makes use of the spawn() method and runs the python code (internally on the terminal) on the server.

2.2.3. The python code converts the video file into an audio file using the moviepy library and then divides the audio file into chunks.
2.2.4. Then, the text for each chunk is generated using the SpeechRecogniton library available in python. It internally uses the recognize_google() method to convert the audio chunk into text.
2.2.5. Finally, the text generated for each chunk is combined and sent back to the browser.

2.3. In the front-end, the text response is stored in a .txt file and made available to the user for downloading.

2.4. The accuracy and quality of text generated depend on the input audio/video file and the SpeechRecognition library.

Server requirements:

1) Python requirements:
   a) Install SpeechRecogntion and moviepy packages.
      i) command: pip install SpeechRecogintion
      ii) command: pip install moviepy

2) Node JS requirements:
   a) Install express-fileupload package.
      i) command: npm install --save express-fileupload

Contribution:

Shaon Dasgupta:

Survey builder Front End React js;
Various navigations (preview,save,edit,delete)
Matrix-Likert and multiselect question types;
Integrating with backend;
Design of backend schema;
Report

Ritabrata Podder:

Survey builder Backend MongoDB integrating with node js:
routes for rest services for various operation from UI
(delete/edit/create/find) ,design of backend schema
Report

Vishal Mani Mishra:

Responses UI
Transcript generator CSS
Transcript generator UI
Report

Gargi Dwivedi:

Survey Builder front end react js
Single select
Text-based question types
CSS
Integration
Report

Yaswanth Haridasu:

Transcript generator python scripting
Node JS
Report