ASHESI UNIVERSITY

DEVELOPING A DATA VISUALISATION TOOL FOR THE CANVAS GRADEBOOK

APPLIED PROJECT

B.Sc. Computer Science

Nana Afua Osimpo Kesewaah Amo

2022

ASHESI UNIVERSITY

DEVELOPING A DATA VISUALISATION TOOL FOR THE CANVAS GRADEBOOK

APPLIED PROJECT

Applied Project to be submitted to the Department of Computer Science, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

Nana Afua Osimpo Kesewaah Amo

2022

DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgement

A big thank you to all who helped me out in various ways to work on this project. And, to my parents for their continuous encouragement.

Abstract

This work encapsulates the stages involved in developing a data visualiser web application, CanSea. It focuses on providing a platform that would aid in an easier way to interpret data. For the development of the proposed system, requirements were gathered to ensure that user needs would be met, and that the system was achieving a purpose. Data visualisation has become an important part of human lives because, from different research papers, it is being used by different fields, including education to cut down time and increase productivity by making data easier to understand.

This system was designed to be used by the Ashesi University lecturers and faculty interns, where they can upload csv files exported from Canvas, a learning management system used by the school. The results from this project suggest that the system is feasible and would make it easier to interpret raw data.

Table of Contents

DECLARATION i
Acknowledgement ii
Abstract iii
Chapter 1: Introduction 1
1.1 Background 1
1.2 Justification of Study 2
1.3 Aim, Objectives, and Proposed Solution 2
1.4 Related Works 3
Chapter 2: Requirements and Specifications 6
2.1 User Classes and Use Cases 6
2.2 Requirement Gathering Process 6
2.3 Functional Requirements 7
2.4 Non-functional Requirements 8
2.5 System Constraints 9
Chapter 3: Architecture and Design 10
3.1 Architecture 10
3.2 System Overview 10
3.3 System Diagrams 10
Chapter 4: Technologies and Implementation 15
4.1 Implementation Overview 15
4.2 Frontend Implementation 17
4.3 Database Implementation 20
4.4 Backend Implementation 20
4.5 Tools and Technologies 22
4.6 Software Methodology 24
Chapter 5: Testing and Results 26
5.1 Component Testing 26
5.2 System-Level Testing 27
5.3 User Testing 27
5.4 Analysis of Test Results 29
Chapter 6: Conclusion and Recommendations 30

6.1 Overview 30
6.2 Challenges 30
6.3 Recommendations & Future Works 30
References 32
Appendix 33

Chapter 1: Introduction

Data visualisation can be defined as a way of visualising a data set in a way that can be easily understood. These forms include but are not limited to, pie charts and bar graphs. Data visualisation is important in education because, it helps to make sense of data and help draw correct conclusions. It can clearly draw out patterns and identify outliers from a given data set [1].

For lecturers and faculty interns of Ashesi University, they may easily want to see trends in performance for every assignment, quiz, project, or lab work done. This may not easily be the case when they use the [1]Canvas gradebook. Apart from seeing the average score, lowest and highest score they may need to visualise these scores to see where the class stands in general.

1.1 Background

Ashesi uses a learning management tool known as [2]MyCamu which stores students' grades throughout the semester, also has their daily schedule and allows them to enrol for courses. At the end of every semester, lecturers can generate bar graphs of student performance and use this for their assessment for the school. Apart from MyCamu, the school also makes use of another system known as Canvas Learning Management System, which is also mainly for storing student grades, taking of quizzes, and giving them access to course material. MyCamu and Canvas do not generate graphs for specific assignments, hence this data visualisation system creates the opportunity for lecturers to visualise the various assignments that students do.

[1] https://www.instructure.com/canvas
[2] https://www.mycamu.com/#/

1.2 Justification of Study

According to research by Tableau, every STEM field as well as government, finance, marketing, history, consumer goods, service industries, education and sports among other fields all benefit from understanding data that they may be working with [2]. In a report [3] by Awoonor-Williams of the Ghana Health Service in 2018, he spoke about how data visualisation is employed to visualise data that has been collected and aggregated from health facilities. He spoke of how Microsoft Excel is employed to draw graphs which help to understand the data and improve the use of the data as it has been visualised.

These statistics go on to show how many people use data visualisation tools within Ghana, across the continent, and all over the world, for different purposes. Being able to visualise data is important because it makes it easier to identify trends, patterns, outliers and just makes a whole dataset easier to comprehend, as compared to when it is in its raw form.

Sustainable Development Goal 9 states that, 'Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation' [5]. In explaining this goal, the United Nations (UN) has made it known that information and communication technology are essential to achieving sustainable development. Developing a system which would help amass and make large amounts of data easy to interpret and easier to work with is a step in the right direction towards achieving this goal.

1.3 Aim, Objectives, and Proposed Solution

The aim of this project is to develop a data visualisation system that is built on design principles from Human Computer Interaction and Statistics. It would have a simple form that takes information on the type of data, the audience, the purpose of the presentation, etc., then

runs it through a back-end program to decide what chart and colour scheme to use, then represents the data accordingly.

The system would allow for easier data inputs and easy outflow of the data. The data being viewed should be well visualised so that it is easy to read and matches the type of visual representation the user needs.

The objectives of this project include:

- Cutting down the time it takes to interpret data.

- Reveal patterns and trends in a dataset to easily allow for changes to be made where necessary.

- Making quicker decisions about a dataset as it is easily being visualised.

1.4 Related Works

There are several data visualisation tools that exist and are widely used across the world in different fields such as finance, education, and sports by data analysts. These data visualisation tools have especially made it easier to work with large datasets.

BeSocratic

beSocratic is used for the visualisation of data to improve learning and make student interactions with the learning system much easier. beSocratic also analyses and visualises student data to allow teachers to easily examine student progress and check the overall progress of a class. beSocratic contains several modules that recognize, evaluate, provide feedback, and analyse student drawn structures, including Euclidean graphs, chemistry molecules, computer science graphs, and simple drawings. The system uses a visual, rule-based authoring system which enables the creation of activities for use within science, technology, engineering, and

mathematics classrooms. beSocratic records each action that students make within the system. These visualisations can help teachers quickly identify common strategies and errors for large groups of students [6]. beSocratic system can capture free-form drawings of students into the system and analyse it. This system is used by both students and teachers.

Tableau

Tableau is a data visualisation tool that primarily focuses on making it easier for individuals and business organisations to interpret their data. Tableau allows for people of different fields ranging from businesses to schools to data scientists to have an easier way of viewing trends in their work without much difficulty. Tableau also boasts of having an end-to-end encryption and to be very secure and powerful as well [5]. Tableau has been used as a data visualisation tool to work on a dataset from CSEdResearch.org to prove the effectiveness of data visualisation and has proven to work very well.

The Upshot

The Upshot is a website that uses data visualisation to analyse and produce charts that represent journalistic news. The purpose of Upshot is to provide graphics and some form of interactiveness to help certain categories of people understand the news better. The Upshot system was introduced at New York Times after a long time of their operation, after the need for easier assimilation of data was observed from readers. The development of Upshot shows that people are interested in understanding everything around them, and highlights on the importance of being able to visualise data for an easier and faster understanding.

The proposed data visualisation system, CanSea, is going to take the student scores for different assigned work and be able to generate a graphical representation for either the general class performance or the general performance for a particular student in the course.

Chapter 2: Requirements and Specifications

2.1 User Classes and Use Cases

There are three user classes for this system. The primary users of the system are the lecturers and faculty interns. The lecturers and faculty interns would have the same privilege level to the system as they work closely with each other.

The secondary user of the system is the system administrator who would assign lecturers and faculty interns to the courses that they teach and to also create default accounts for them.

To better understand the requirements of the system, the following scenario is given to show the user story:

Scenario: Dr. Djan is a lecturer at Ashesi University, who is committed to the progress of all his students. He teaches three courses at the school and hence engages with several students on a regular basis. Due to the number of students he deals with, he may not always remember everyone's name and be able to easily have one on one sessions with all of them. Also, because of the learning management systems the school uses, he is not able to get a vivid breakdown of the performance of students for assigned work. With this data visualiser system, Dr. Djan can view student performance for every work assigned and check the general class performance as well as individual's performance. The system would allow Dr. Djan to also view gender distributions. Dr. Djan can comfortably observe the progress of all his courses and individual students and easily see where the problems are.

2.2 Requirement Gathering Process

In gathering the required data necessary for developing the system and getting the most accurate requirements to suit the user needs, in-person interviews were conducted with five (5) faculty interns initially, which was also then followed up with online interviews with two

lecturers. Feedback from these interviews showed an interest and acceptance of working with a data visualisation system such as this one. The faculty interns were willing to use such a system as they believed it would make assessing student performance easier. The research also showed that the identified user classes already have exposure to visualisation charts and know how to view and interpret them without much difficulty.

In getting the sampling population, convenience sampling was used. This sampling method involves getting people who meet the criteria for the stakeholders, based on the convenience of their accessibility, geographic location, and willingness to be a part of the study.

Analysing the Interviews

There were some major areas which were looked at when analysing the system. These included clearly identifying and understanding the user classes so that the charts provided would suit their needs. Also, the amount and type of data was looked at, as well as the level of familiarity with visualisation so that no kind of overly complicated chart was incorporated. The time of the user classes was also considered, so that they could easily use the system and view visualisations, without spending so much time. The type of decisions to be made from the data was also considered.

2.3 Functional Requirements

Faculty Interns & Lecturers

[F001] - Sign In

The system should allow lecturers and faculty interns to sign in.

[F002] – File Uploads

The system should allow users to upload the csv file they want to visualise.

The system should be able to read select columns from uploaded csv files.

[F003] - Chart Generation

The system should allow users to choose the chart they want to generate.

The system should be able to generate the chart that is chosen.

[F004] - Colour Scheme Generation

The system should generate the colour schemes for the generated chart.

[F005] - Understanding Data

The system should be able to read and understand the data that is fed into it.

System Administrator

[F006] – Assign Courses

The system administrator should be able to assign courses to lecturers and faculty interns.

[F007] – Registration of faculty

The system administrator should be able to register lecturers and faculty interns to the system.

2.4 Non-functional Requirements

[NF001] - Scalability

The system should be able to allow for increase or decrease in capacity and functionality when the need be.

[NF002] - Usability

The system should be easy to use and understand.

[NF003] - Reliability

The system should be able to fulfil all its required tasks.

[NF004] - Maintainability

The system should be easily corrected and maintained.

[NF005] - Security

The system should be secure to avoid any data breach.

[NF006] - Response Time

The system should not delay in generating charts.

2.5 System Constraints

1. The application requires the internet to work.
2. This data visualiser tool is only limited to the Canvas Gradebook.

Chapter 3: Architecture and Design

3.1 Architecture

The system is solely designed as a web application hence it is supposed to be supported by various web browsers. The user interacts with the system by logging onto the system and choosing the course they want to view grades for. One important note is that a lecturer or faculty intern can only see courses that they have been assigned to and nothing more.

After choosing the desired course, the lecturer/faculty intern may then upload the csv file for visualization or view already done visualisations from previously uploaded csv files.

3.2 System Overview

The system overview captures the different tiers involved in the interaction for this system. There are three different tiers which are looked at, i.e., the presentation tier, the logical tier, and the database tier.

The presentation tier is where the user interacts with the system. It includes all the clickable parts and parts where the user can input data into the system.

The logical tier is where backend operations are carried out. It is the part connected to the database to carry out logical operations and requests from the user.

The database tier is where the system data is stored. It stores information on the system stakeholders, i.e., user credentials as well as the files that are uploaded to the system.

3.3 System Diagrams

The following diagrams are used to show the structure of how the system is set up, the flow of data through the system, and how the user is expected to interact with the system.

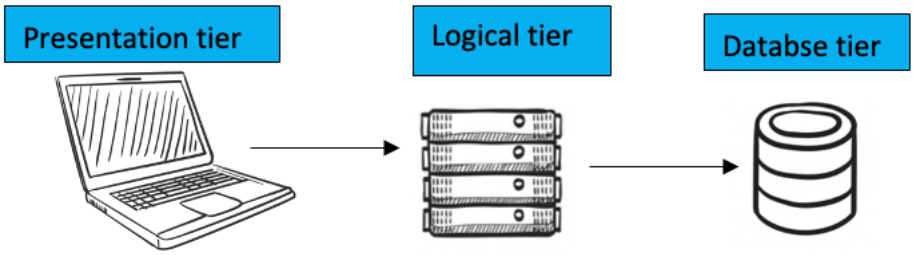


Figure 3.1: 3-tier Architecture Diagram

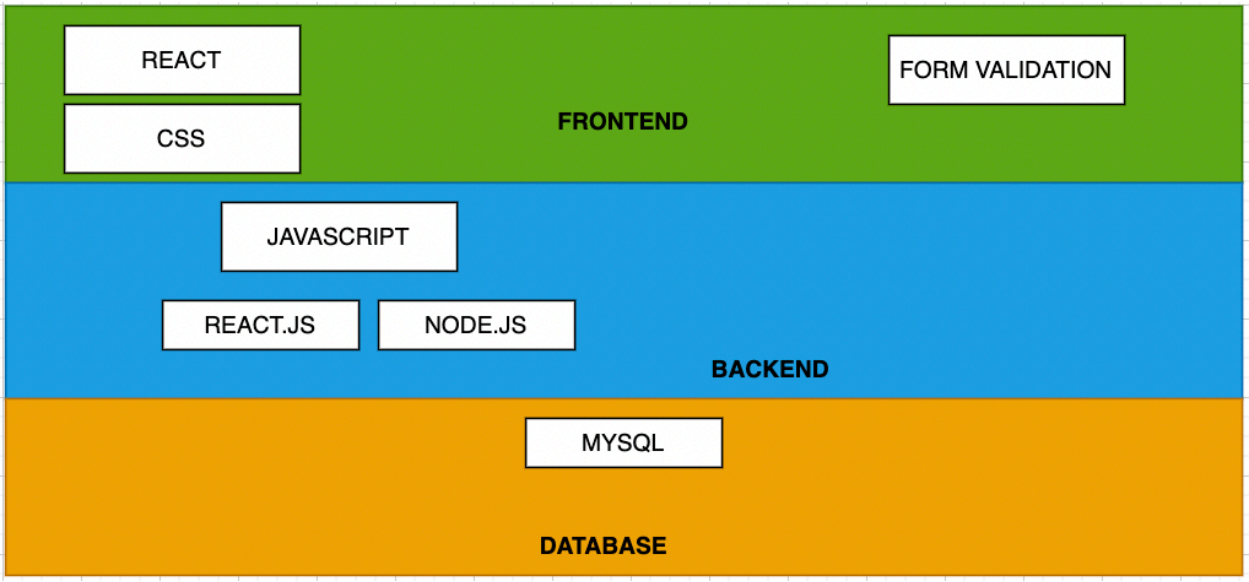


Figure 3.2: High-level system architecture

A use case diagram, as depicted in figure 3.3, is used to highlight on the identified stakeholders and how they would use the system. From the diagram one can identify the lecturers and faculty interns. They can both login, select courses assigned to them, view student, general course or select assignment performance. They can also upload student grade csv files.

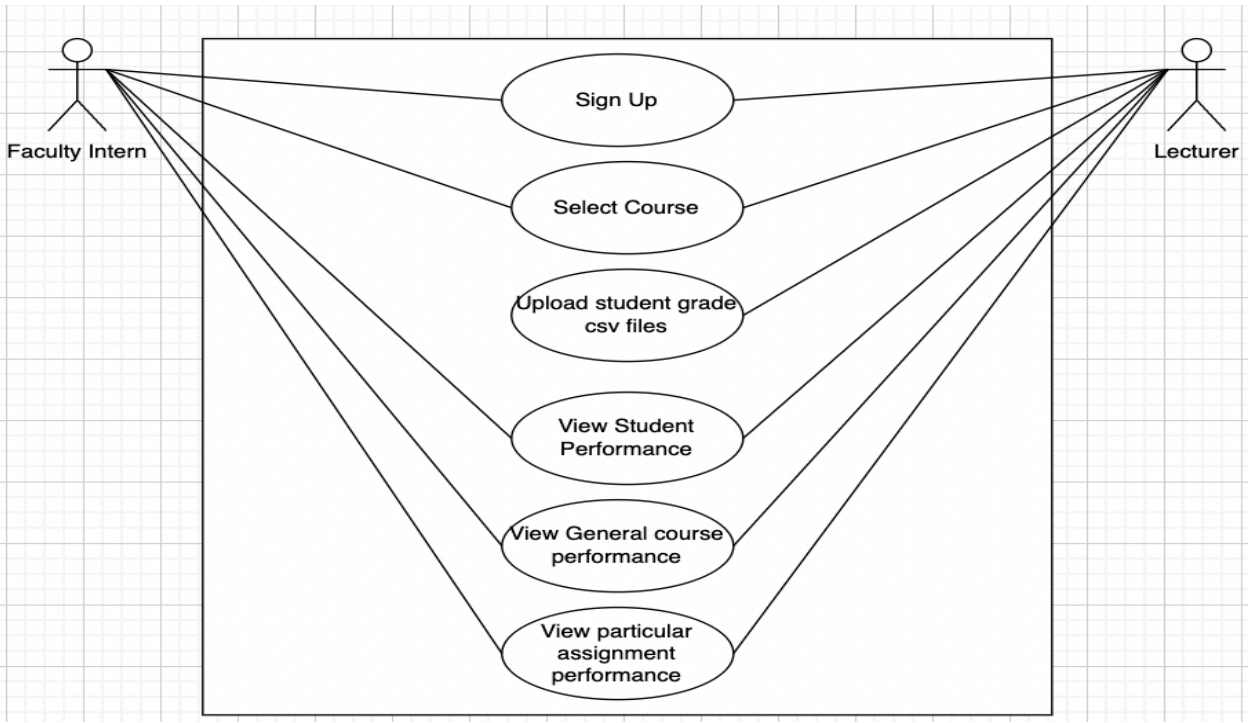


Figure 3.3: Use Case diagram for lecturers and faculty interns

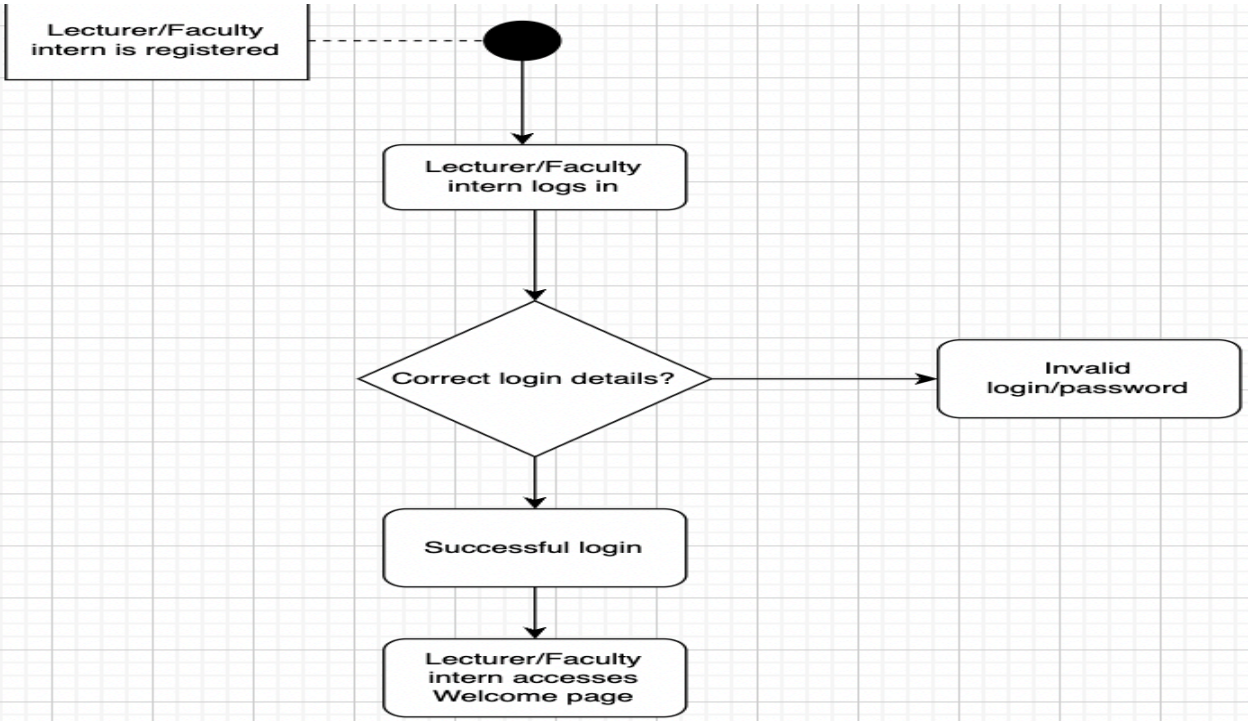


Figure 3.4: Activity Diagram for the login process

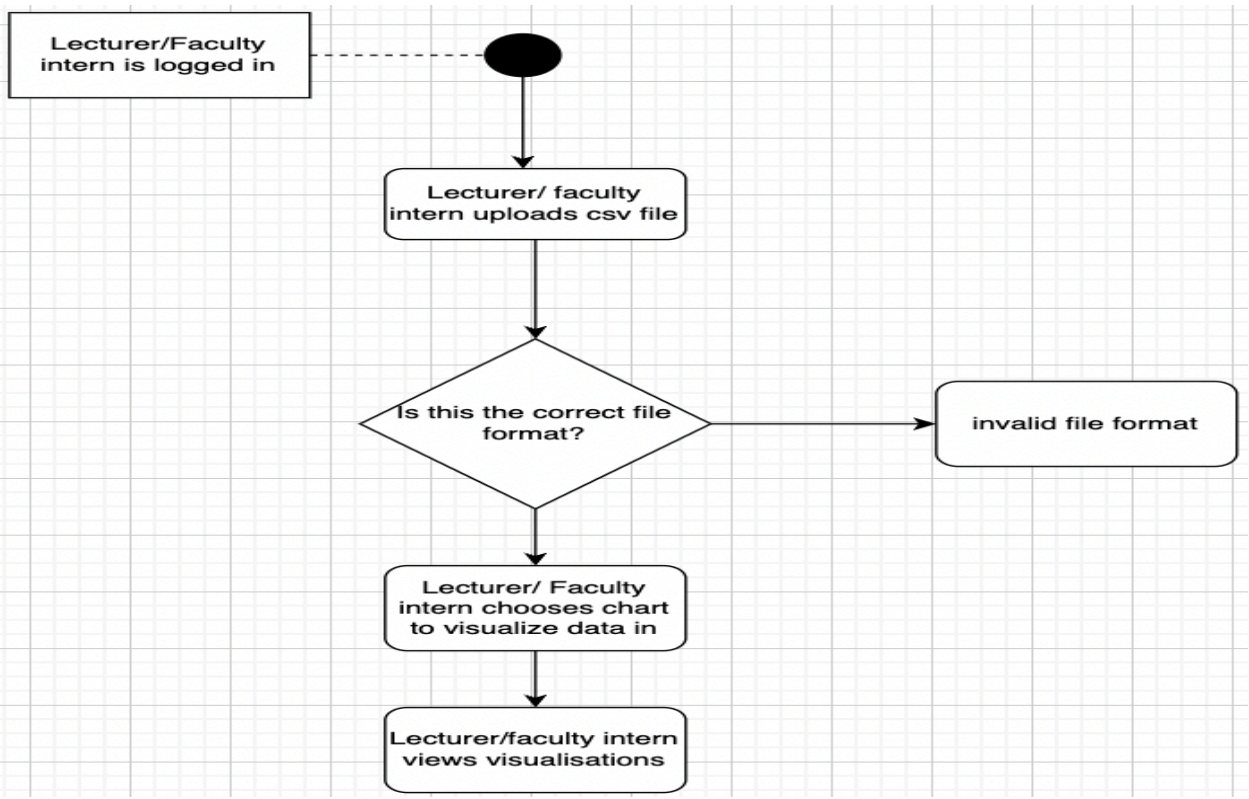


Figure 3.5: Activity Diagram for the csv upload process

The activity diagrams show the flow of activity for different interactions the user classes have with the system. For the login process, in figure 3.4, the lecturer/faculty intern has to login with valid login details. If they are invalid, they cannot access the system. After successful login, they can access the system.

The activity diagram for the csv upload, in figure 3.5, shows the flow of activity when the lecturer/faculty intern is uploading the csv file. The system only accepts csv files, hence if the file format is wrong it gives an invalid file format warning and the upload is not done. If the format is correct, then they can choose the chart they want to visualise in and then continue to view the visualisation charts.

In working with the canvas system towards developing the data visualiser, the following entities shown in figure 3.5 were identified, and this is how they relate to each other. The entities involved are the lecturers, faculty interns, students, courses, cohorts, assignments, grades.

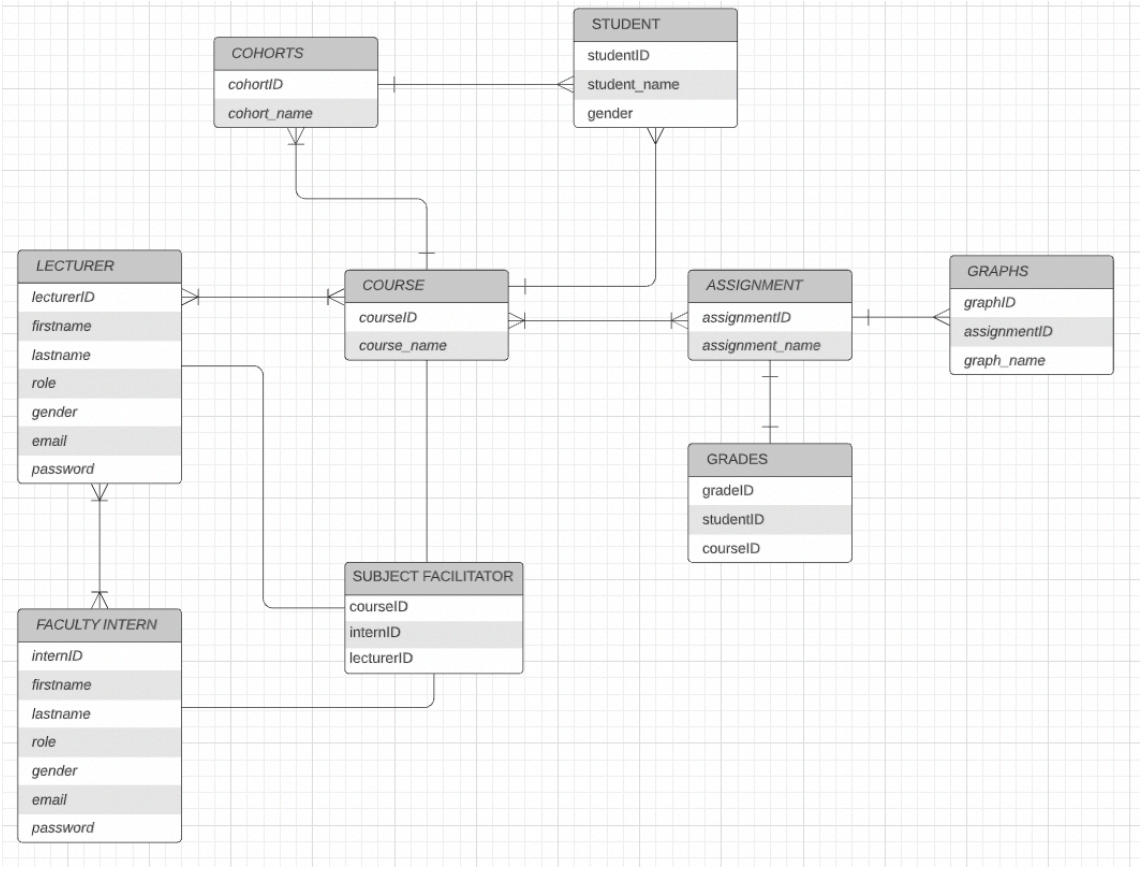


Figure 3.5: ER Diagram

Chapter 4: Technologies and Implementation

This section discusses the processes involved in carrying out the development of the system as well as the technologies involved.

4.1 Implementation Overview

This system is developed in three major phases. The first phase involves the part of the system which can take csv file uploads and read them into the system. The next part of the system is the conversation of the data from the csv file into a visualisation chart. These charts include bar graphs, pie charts and line graphs. The last part of the system involves an algorithm that can read specific columns at a time, without having to read the whole csv file at a go. The other parts of the implementation process of the system are the development of the frontend which is going to be designed based on a Figma prototype.

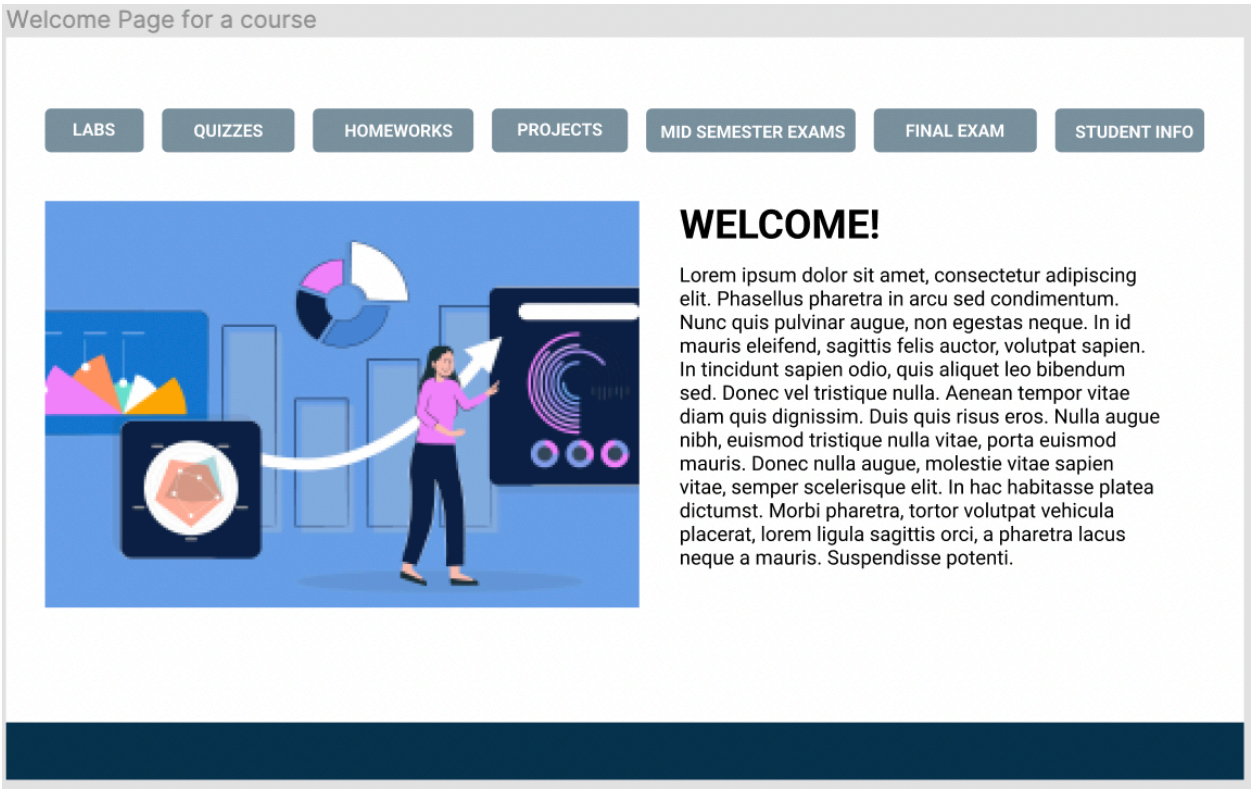


Figure 4.1: The welcome page design on Figma

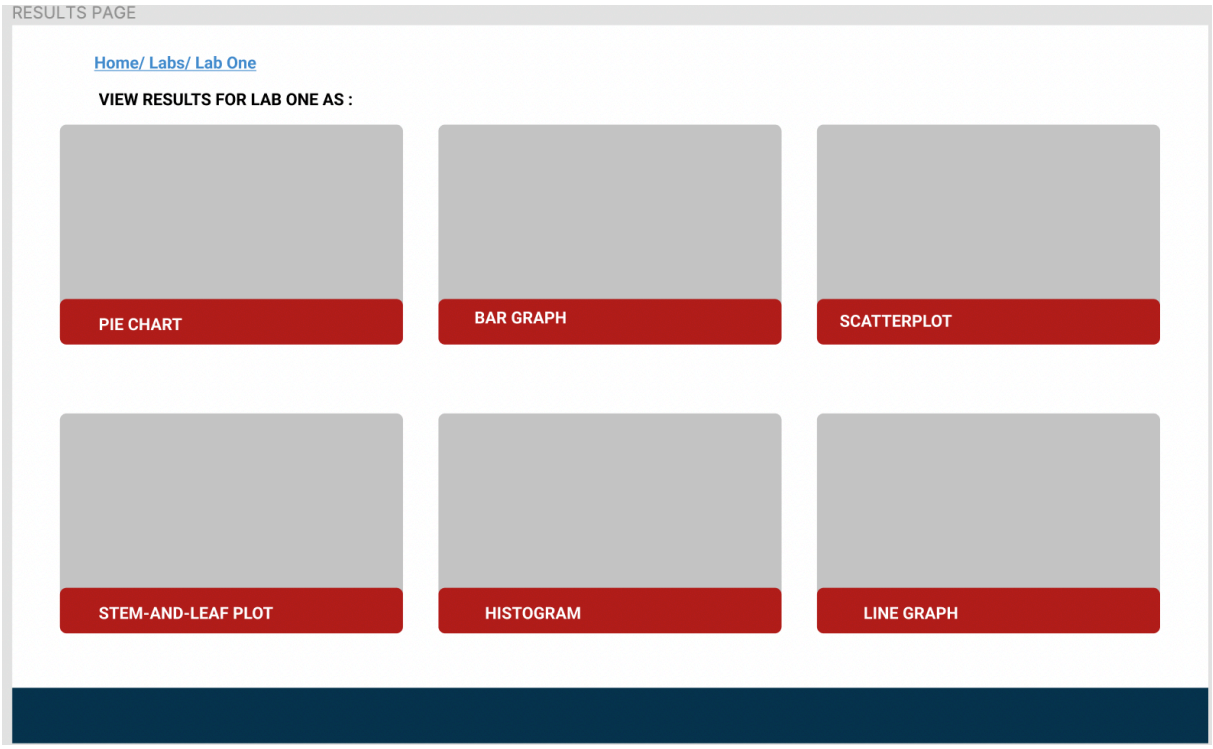


Figure 4.2: Figma design for selection of a particular graph interface

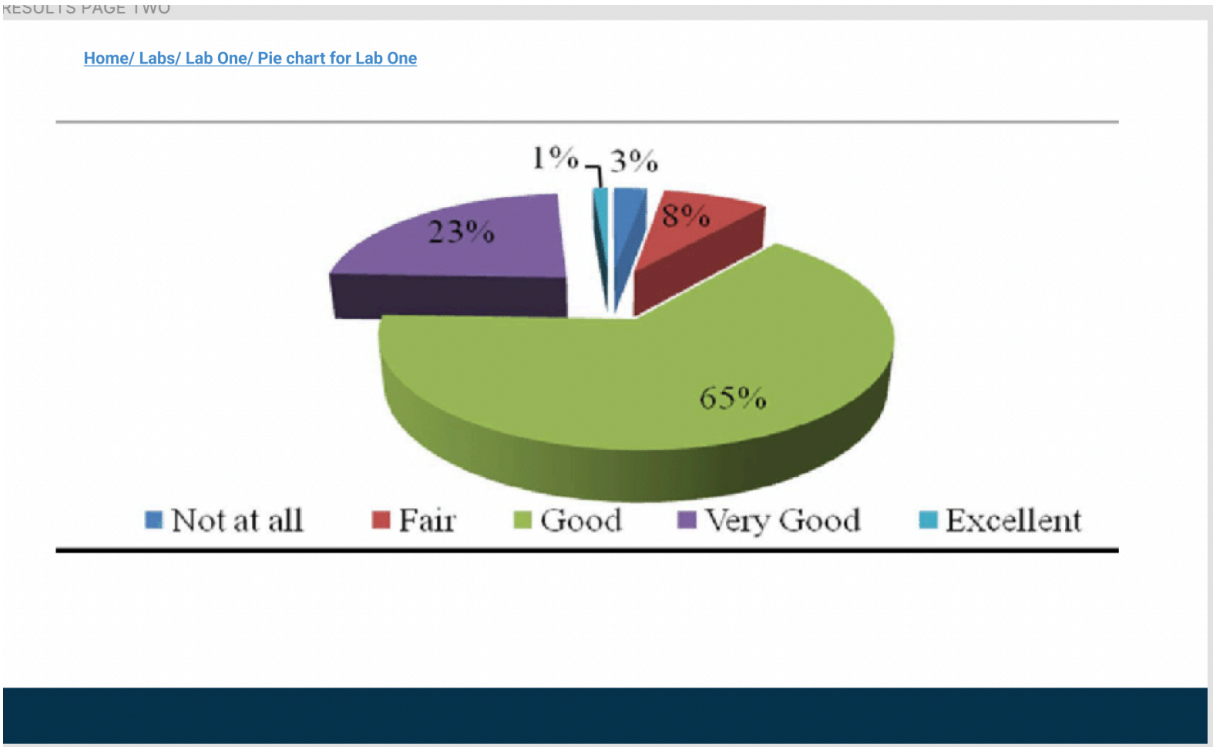


Figure 4.3: Figma interface for the when a particular graph is chosen and generated.

4.2 Frontend Implementation

For the implementation of the frontend ReactJS is adopted to design responsive webpages. The use of components was used to design the system to be as close to the Figma design as possible. Due to a limited timeline, a much simpler user interface was implemented, which still covered all the necessary components to ensure a good user experience.

Figure 4.4: The Landing page of the website

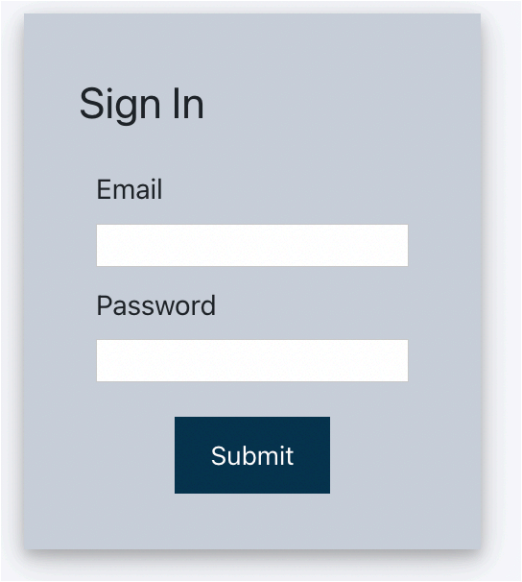


Figure 4.5: Login Interface

Figure 4.5 shows the login interface for lecturers and faculty interns of the data visualisation system. The users need to login into the system to be able to access the main pages to upload the csv files.

Figure 4.5: Interface to allow users to upload files

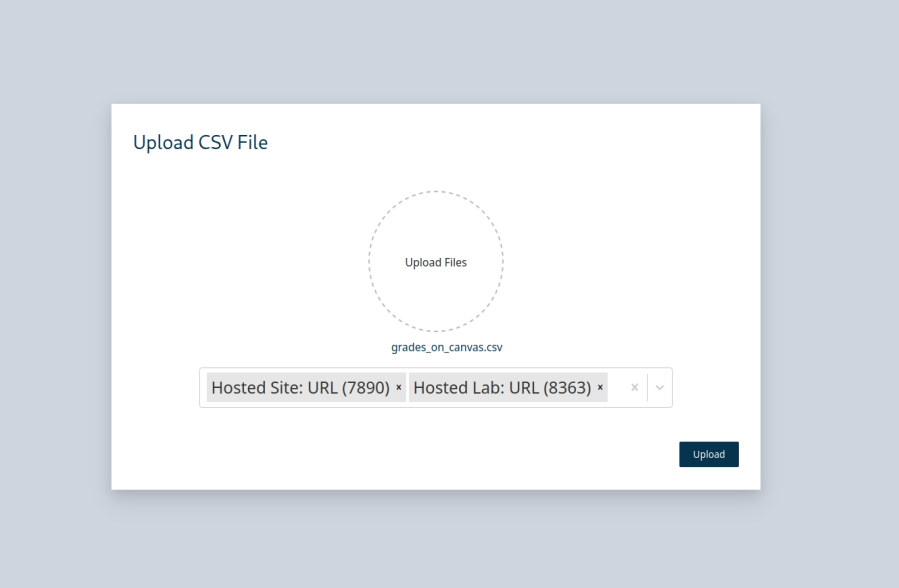


Figure 4.6: The user can see the file they chose and select particular columns.

For the system to access the csv file the user needs to transform, the interface shown in figure 4.5 allows the user to select the 'Upload Files' option and access files on their pc, from which they may choose the preferred file. Users may directly drag and drop files in a folder

directly into the dotted circle shown in figure 4.5 and 4.6. This drop zone was achieved using the react-dropzone component.

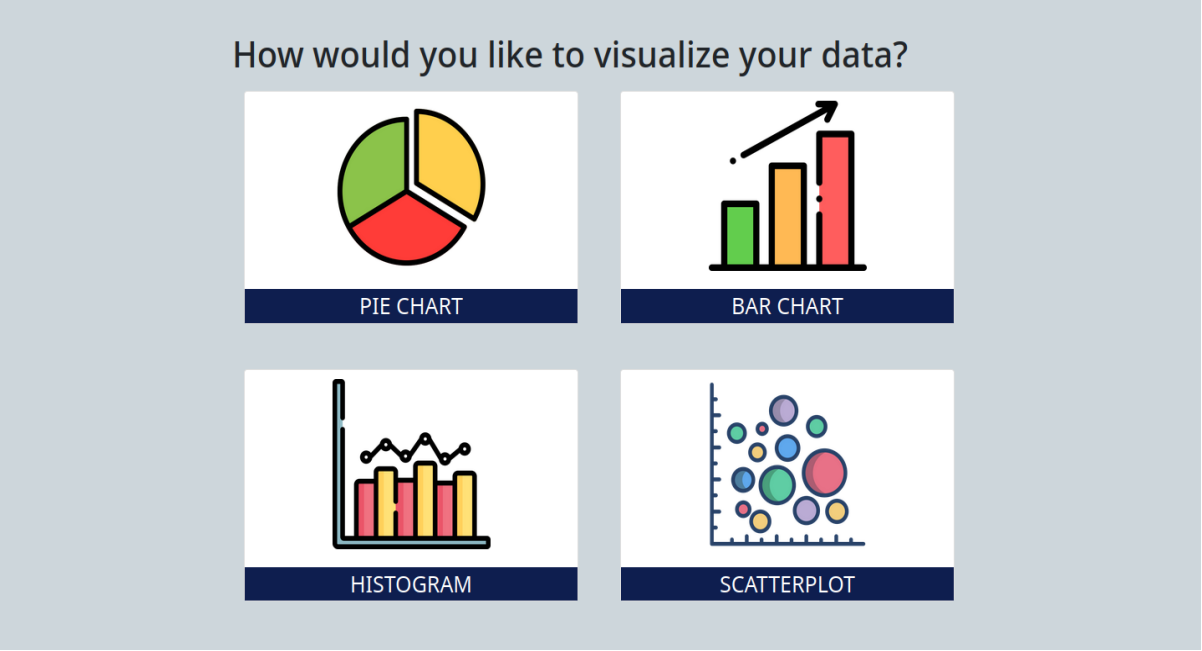


Figure 4.7: The user can select the chart they want to generate.

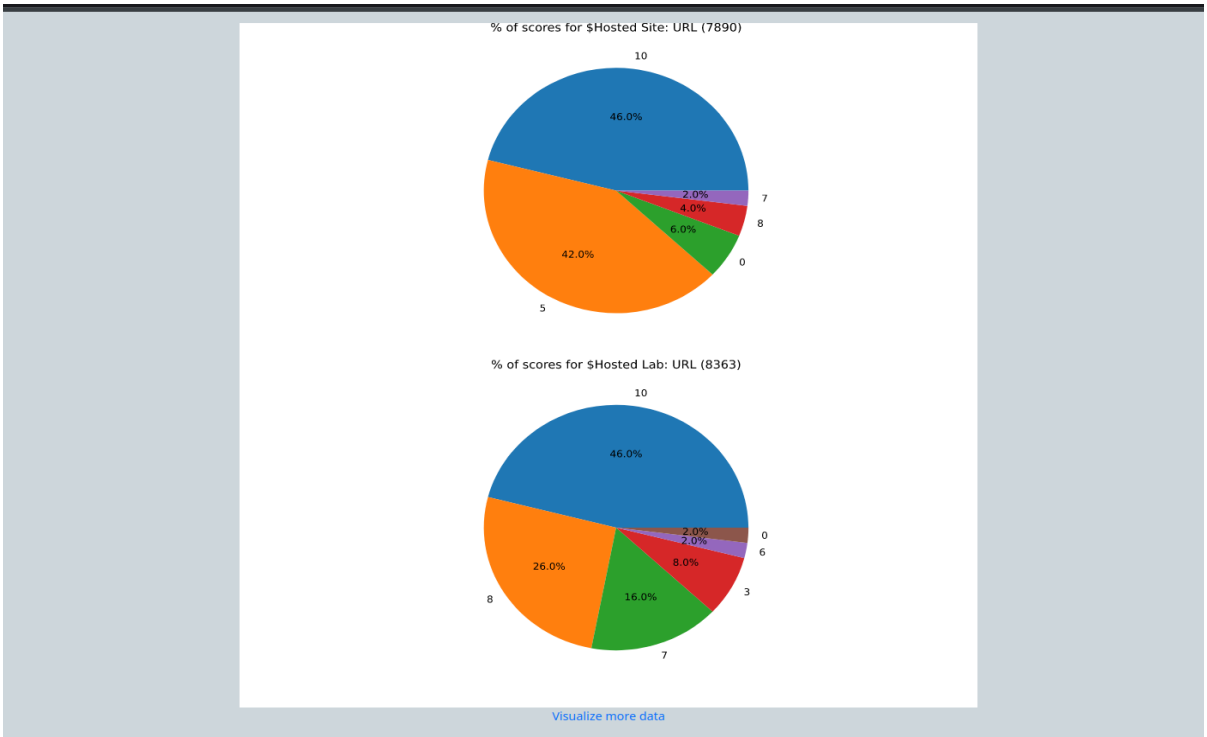


Figure 4.8: Pie chart generated by the system.

4.3 Database Implementation

The input of information and the transformation of the graphs would require a database. MySQL would be used because it is a sturdy and reliable software. It is also easy to use and allows for encryption of user details. The database would store the login credentials, as well as the courses each lecturer and faculty intern teaches. Also, the csv files which would be uploaded would also be stored by the database.

4.4 Backend Implementation

The backend was mainly implemented using Python and Flask. These languages make it easier to implement functions. The system can read successfully read from a csv file and generate bar charts, histograms, our charts and scatterplots.

```python
@app.route("/visualize", methods=["GET", "POST"])
def visualize_csv():
    if request.method == "POST":
        chartType = request.form['chartType']
        chartType = json.loads(chartType)
        f = request.files["file"]
        columns = request.form["columns"]
        columns = json.loads(columns)
        now = datetime.now()
        timestamp = now.strftime("%Y%m%dT%H%M%S")
        if f and len(columns) > 0:
            filename = secure_filename(f.filename)
            split_array = filename.split(".")
            fname = split_array[0] + timestamp + "." + split_array[1]
            destination = os.path.join(app.config["UPLOAD_FOLDER"], fname)
            f.save(destination)
            b64 = perform_visualization(destination, columns, chartType)
            response = make_response(
                jsonify({"data": b64.decode('ascii'), "status": "OK"}), 200)
            return response
        response = make_response(jsonify({"data": "", "status": "ERROR"}), 400)
        return response
```

Figure 4.9: Visualise route

```python
def perform_visualization(csv_file, columns, chartType):
    df = pd.read_csv(
        csv_file, usecols=columns, skiprows=[1, 2], skip_blank_lines=True
    )  # read csv, skip first two rows, skip blank lines
    df.fillna(0, inplace=True)

    if str(chartType) == "1":
        b64 = gen_pie_chart(df, columns)
    elif str(chartType) == "2":
        b64 = gen_bar_chart(df, columns)
    elif str(chartType) == "3":
        b64 = gen_hist(df, columns)
    elif str(chartType) == "4":
        b64 = gen_scatter(df, columns)

    return b64
```

Figure 4.10: Function which decides which chart to print

```javascript
async uploadFiles() {
  this.setState({ uploadProgress: {}, uploading: true });
  const promises = [];
  let columns = this.state.selected.map((e) => {
    return e.value;
  });
  promises.push(this.sendRequest(this.state.files[0], columns));

  try {
    await Promise.all(promises);

    this.setState({ successfullUploaded: true, uploading: false });
  } catch (e) {
    this.setState({ successfullUploaded: true, uploading: false });
  }
}
```

Figure 4.11: Async function to upload csv

```python
def gen_pie_chart(df, columns):
    fig, axes = plt.subplots(len(columns), 1, figsize=(10, 10))
    fig.tight_layout()

    for count, value in enumerate(columns):
        df[value].fillna(0, inplace=True)
        score_count = (
            df[value]
            .value_counts()
            .rename_axis("scores")
            .reset_index(name="stu_scores")
        )

        if len(columns) > 1:
            axes[count].pie(
                score_count.stu_scores, labels=score_count.scores, autopct="%1.1f%%"
            )
            axes[count].set_title(f"% of scores for ${value}")
        else:
            axes.pie(
                score_count.stu_scores, labels=score_count.scores, autopct="%1.1f%%"
            )
            axes.set_title(f"% of scores for ${value}")

    imgdata = BytesIO()
    plt.savefig(imgdata, format="svg")
    imgdata.seek(0)
    base64_img = base64.b64encode(imgdata.read())

    return base64_img
```

Figure 4.12: Pie Chart Generate function

4.5 Tools and Technologies

[3]Python

Python is a high-level programming language that has various functions and can be used to develop websites and software, as well for data analysis. It is not developed for any specific purpose or to solve a particular problem.

React JS

React JS is an open-source JavaScript library that contains various components that are used for developing the user interface of websites. It has much flexibility and allows users to create specific components which changes can be made to as and when needed.

Chakra UI

[3] https://www.python.org/

Chakra UI is a react component library that contains already designed react component blocks that allow for easier development when using React JS.

Flask

Flask is a lightweight Python web framework, that provides certain tools and features which make development of web applications in Python easier. It does not require particular libraries and is very flexible.

Victory.js

Victory.js is a JavaScript library that is used for data visualisation. The Victory charts are employed for the development of bar charts.

Hypertext Markup Language (HTML)

HTML is a markup language that is used to structure contents of a website. It allows for grouping of elements into paragraphs, columns, and lists amongst other structures. It is a widely used and recognised markup language.

Cascading Style Sheet (CSS)

CSS is a language that is used for styling HTML elements. CSS is used to describe how elements should be presented on a page to make it look more organized and pleasing.

Figma

Figma is a prototyping tool that enables users to design wireframes and high-end frontend prototypes for a system they may be developing. The system allows the designer to also simulate user interactions on how client would interact with the system. Figma was employed to develop the different designs that may be preferred for the frontend and the final design and colour scheme that was chosen for the system.

MySQL

MySQL is an open-source relational database management system. It is very efficient for web databases and how rigid the system is.

4.6 Software Methodology

The software methodology employed for the implementation of this system was the agile software development. Agile software development was employed because this method allows for the system to be incrementally developed making sure that it is carefully broken down and it is checked to ensure that the various milestones are being met as the time goes on. A software system, Trello was employed and used to track the progress of the system development. Trello allows the user to create a board where they have cards that they can put their different milestones and group them under “to-do”, “doing”, “done”, and any other categories they may please.

In the development of this system, user stories were also developed on the Trello board. The user stories were to serve as a guide in coming up with the milestones and keep track of how the user is expected to use the system.

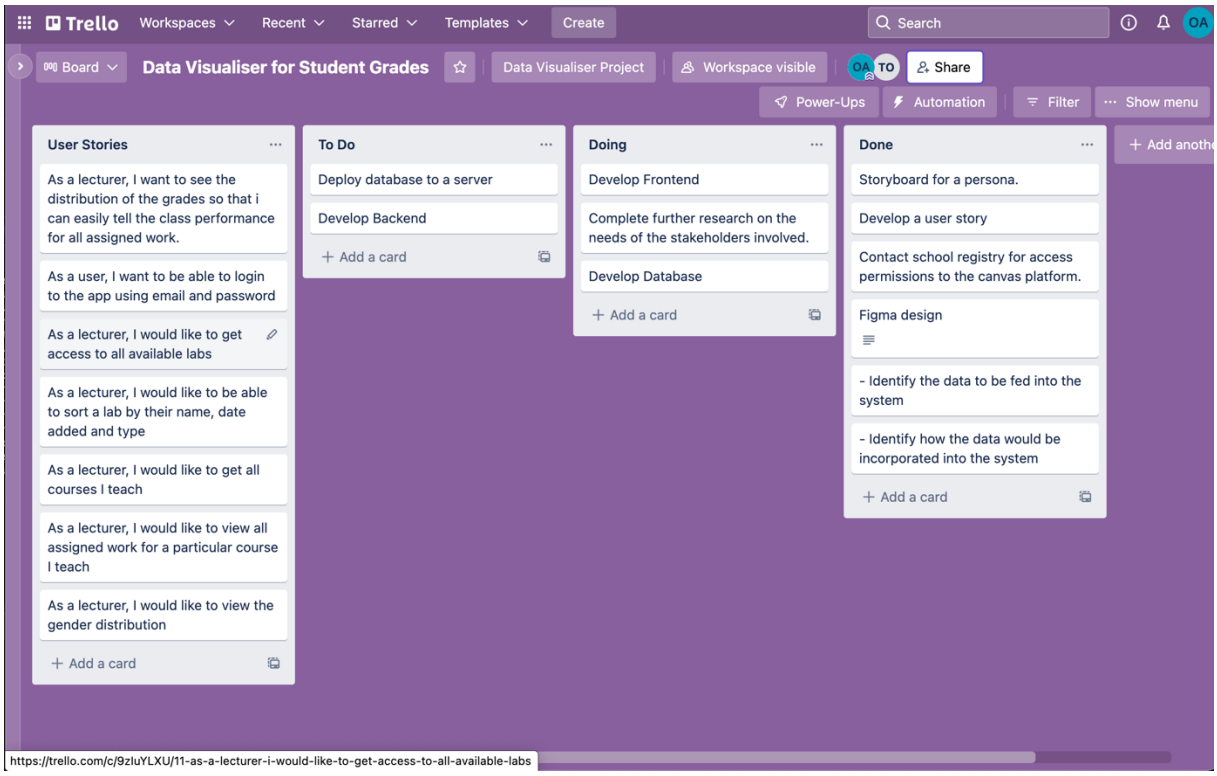


Figure 4.13: Trello Board developed for this data visualiser system.

Chapter 5: Testing and Results

Testing of the system is done in parts to ensure that when the system is running there would be no issues that would be faced. This section talks about the tests have been completed so far.

5.1 Component Testing

This involved testing different components of the system. Over here testing different csv files with two columns for visualisation has been done. The system can successfully read from these csv files and generate bar charts.

An initial issue that came up when testing the rendering of data into bar charts was that the displayed data was squeezed together because of improper planning. The sample size, which has the average class size per course to be fifty (50) was what was used to perform this test.

Student	ID	FinalExamScore
Liam	2371	57
Olivia	3737	80
Noah	3902	55
Oliver	3862	60
Elijah	3389	71
William	2884	70
James	2908	57
Benjamin	3750	59
Lucas	3185	74
Henry	3598	55
Alexander	2928	60

Figure 5.1: Sample of a csv file that was used to test the system.

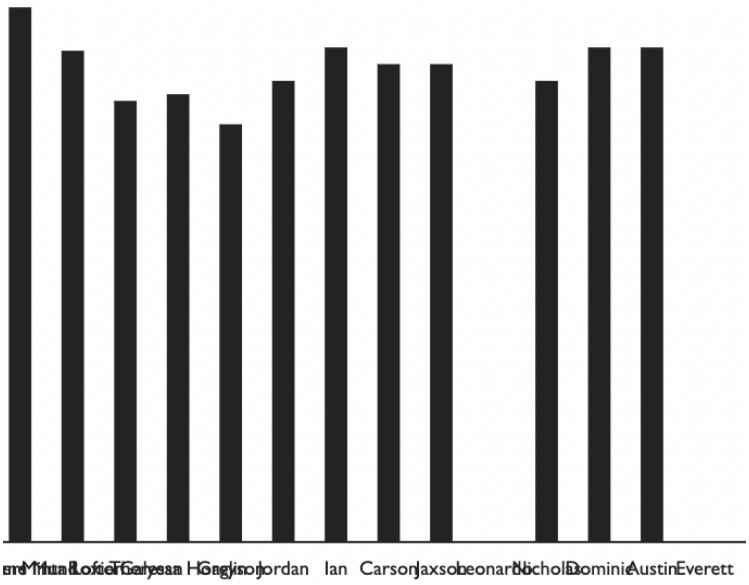


Figure 5.2: Display issue-overlapping names

In testing the login and registration API, the use of Postman was employed and a GET and POST method was used to call for specific usernames and emails from the database. When there is a match, a success message is returned as well as a token.

5.2 System-Level Testing

This testing is done to validate the software and explore if it is well integrated. This involves deploying it to an online server and hosting it there. It also involves running it on other browsers other than the one that was used for development. For this system, the reactJs web application was hosted to netlify and the php backend was deployed to linode.

5.3 User Testing

This testing involves the involvement of the stakeholders and testing the developed system with them. In conducting the user testing for CanSea, a faculty intern tested the system. The faculty intern was able to successfully navigate the system and log onto the system with login credentials provided from the system administrator's part of the system. The faculty intern was able to successfully log in and select a file for upload.

They were able to successfully click the dropzone and get a file dialog open, where they could choose a file from their laptops, and upload a file by dragging and dropping it into the dropzone.

A test sample of exported grades, using the same format as the canvas gradebook was uploaded. On initial testing, the system could not read from particular columns and hence a much simpler csv was used to test for the chart generation. On the second test the system was successfully able to read from select columns to generate charts.

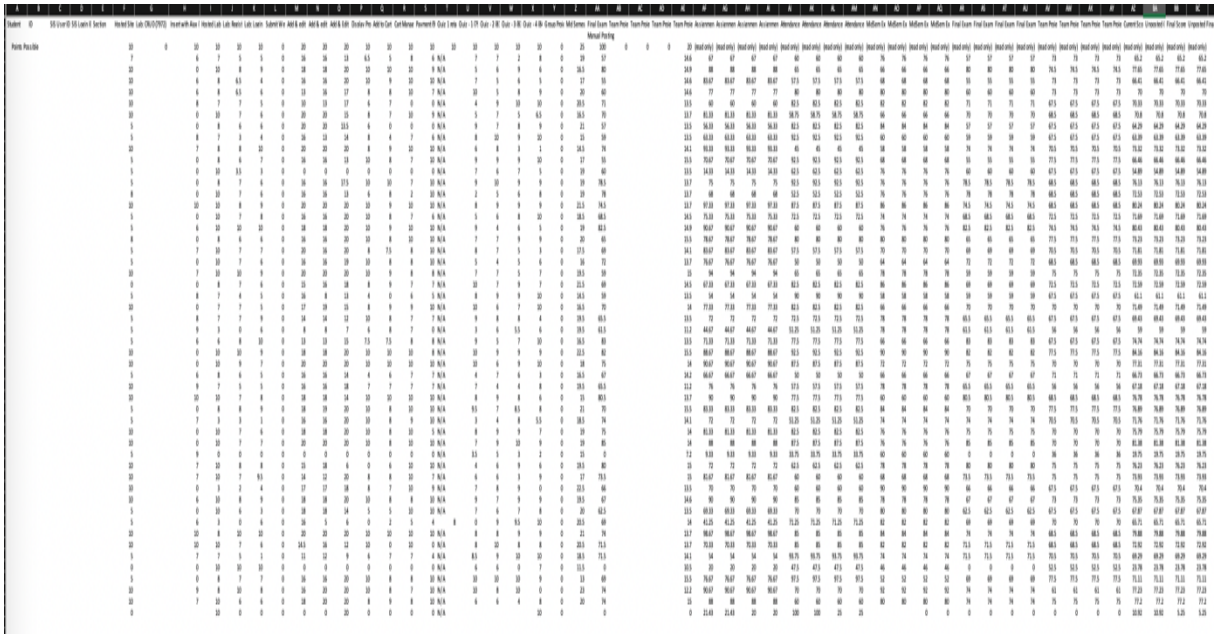


Figure 5.3: Sample of the number of columns when grades are exported from Canvas Gradebook

5.4 Analysis of Test Results

Below is a summary of testing CanSea, against the requirements.

Table 1: Functional Requirement Achievement

Requirement Number	Functional Requirement	Meets Expectation
[F001]	The system should allow lecturers and faculty interns to sign in.	Yes
[F002]	The system should allow users to choose the chart they want to generate.	Yes
[F003]	The system should be able to generate the chart that is chosen.	Yes
[F004]	The system should generate the colour schemes for the generated chart.	Yes
[F005]	The system should be able to read and understand the data that is fed into it.	Yes
[F006]	The system should allow users to upload the csv file they want to visualise.	Yes
[F007]	The system administrator should be able to assign courses to lecturers and faculty interns.	No
[F008]	The system administrator should be able to register lecturers and faculty interns to the system.	Yes
[F009]	The system should be able to read selected columns from the uploaded csv file.	Yes

The results from table 1, show that the system performs most of the functional requirements that is expected of it. The table indicates an issue with the system allowing users to choose the chart they may want to generate. The system also provides a way for the csv file to read by reading from select columns at a time.

Chapter 6: Conclusion and Recommendations

6.1 Overview

This project aims to provide a visualisation that would help Ashesi lecturers and faculty interns easily view charts that give the general class performance as well the performance of individuals. This system is designed as a web application. Currently the system can read a csv file which lecturers would be expected to export from the Canvas learning management system.

6.2 Challenges

The practicalities of implementing this data visualization system to upload data into different parts of the database, via uploading and reading every column in an uploaded csv file uploaded from canvas, proved challenging, and are worth reporting since these are the sorts of obstacles that will beset projects that may require one to develop an algorithm that can read select and read particular parts of a csv at a time, rather than the whole document at a go.

6.3 Recommendations & Future Works

Some recommendations that are important and would help this data visualisation system work more effectively include:

Connect to the Canvas API

When the system is connected via the canvas API, there would be no need to let the system administrator add lecturers and faculty interns to the system. Rather the API would grant access to the login details of the Canvas system and the courses they are already assigned to. Apart from that, the grades from each course would be able to be accessed by the data visualisation

system, without having to let the lecturer go through the process of exporting the grades from canvas before uploading it into the data visualisation system.

One Time Upload

The system should allow the csv file to be stored for a particular course and allow the lecturer to continuously choose different columns and charts without having to upload the csv file every time they want to generate a particular chart.

References

[1] Kirk Andy, 2012. Data Visualisation: A Successful Design Process. Retrieved March 4, 2022, from https://books.google.com.gh/books?hl=en&lr=&id=I4qBVLfD3t4C&oi=fnd&pg=PT6&dq=importance+of+data+visualization&ots=b7YGtQcD7u&sig=IWJ0UaMFCM8u74AFBaUa8-I3msE&redir_esc=y#v=onepage&q=importance%20of%20data%20visualization&f=false.

[2] Splash Business Intelligence Inc. 2021. Importance, Purpose, and Benefit of Data Visualisation Tools! Retrieved October 5, 2021, from https://splashbi.com/importance-purpose-benefit-of-data-visualization-tools/

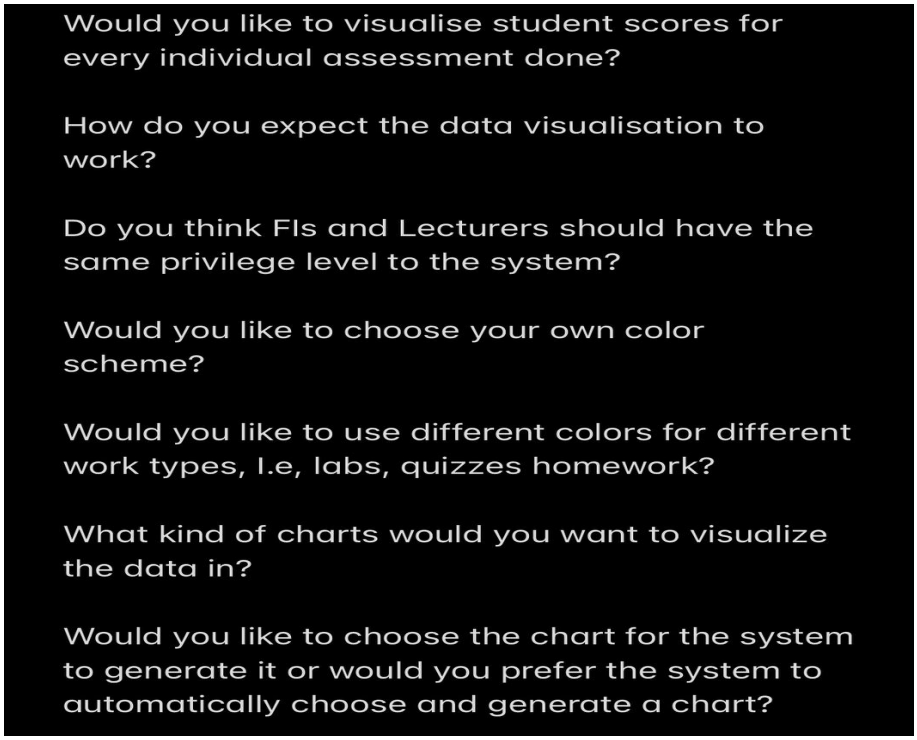
[3] Tableau. 2021. What is Data Visualisation? Retrieved October 5, 2021, from https://www.tableau.com/learn/articles/data-visualization

[4] Koku Awoonor-Williams. 2018. The Power of Data Aggregation and Visualisation Tools for Improved Decision-Making. (April 2018) Retrieved October 10, 2021, from https://www.swisstph.ch/fileadmin/user_upload/SwissTPH/Events/Symposia/SS2018/3.2_Awoonor-Williams_16-9_SS2018.pdf

[5] Samuel Bryfczynski. 2012. *BeSocratic: An Intelligent Tutoring System for the Recognition, Evaluation, and Analysis of Free-form Student Input.* All Dissertations. Clemson University. Clemson, SC.

[6] Tableau. 2021. What is Tableau? Retrieved October 17, 2021, from https://www.tableau.com/why-tableau/what-is-tableau

Appendix



Set of questions used to conduct the interviews with lecturers and faculty interns.