



Exactscore: Result Analysis

Mohammed Mubashir Uddin ^{1*}, Mir Rahimuddin Ahmed Khan ², Syed Haji Ali ³, Asad Hussain Syed ⁴

¹⁻³ BE CSE (AI& DS) MJCET OU Hyderabad, Telangana, India

⁴ Assistant Professor CS & AI department MJCET OU Hyderabad, Telangana, India

* Corresponding Author: Mohammed Mubashir Uddin

Article Info

ISSN (online): 2582-7138

Impact Factor: 5.307 (SJIF)

Volume: 05

Issue: 01

January-February 2024

Received: 12-10-2023;

Accepted: 15-11-2023

Page No: 57-65

Abstract

The analysis of academic performance of students is an important concern for the universities and colleges of higher learning as it is very important for planning and management of the teaching pedagogy. There is a need for the system to examine and assess the result of students in order to understand how effective the existing education system is accessing the factors for academic performance is also important so that students can be assisted to develop their individual academic performance in the university.

Result analysis of universities is the process of evaluating and interpreting data related to the academic performance of students. This can be done for various purposes, such as to identify strengths and weaknesses in the education system, to monitor student progress and support, and to make informed decisions about academic policies and programs. The first step in conducting a result analysis is to gather data. This may include exam scores, grades, attendance records, and demographic information. The data is then organized and processed using statistical methods, such as descriptive statistics and inferential statistics, to generate meaningful insights.

Another important aspect of result analysis is comparing the performance of different groups of students, such as those from different socioeconomic backgrounds or with different learning needs. This can help to identify disparities in the education system and to target resources and support where they are needed most.

Finally, result analysis can be used to evaluate the effectiveness of various academic programs and initiatives. For example, data can be analysed to determine the impact of a new teaching method or the effectiveness of a tutoring program. This information can be used to inform future decisions and to continuously improve the quality of education.

The existing system is done manually where the faculty members of university analyse it which is time consuming and frustrating. This project is intended to focus on the interaction of different factors and the role that they play to the performance of students in academics in their college or university. The proposed system will have a UI interface which asks you put URL of results link of OU website and hallticket number range. After clicking submit, it analyses with different parameters and displays all results in organized format automatically.

DOI: <https://doi.org/10.54660/IJMRGE.2024.5.1.57-65>

Keywords: academic, OU, Result, Analysis

Introduction

Today, in most of the universities and colleges, the performance of the students academically has recently come under analysis for a number of reasons. There is number of factors involved that affects the academic performance of students. These factors may vary from one set of students to next, form one academic background to another and from one state or nation to another. At times, students work hard but their hard work may not be correlated with their result. This may lead to high failure percentage and thus increases the rate of training these graduates. Low pass percentage of students may enforce enormous cost on community in terms of low number of students graduating and reduced intake of probable users due to absence of spaces.

So, in order to confirm that a big portion of labour force is highly trained, all universities and college must ensure and consider the aspects that affect the performance of the students.

To attain high profile position and more than expected salary, the higher education level is must for any organization. Many researches have been conducted to analyse the factors associated with academic performance of students in various universities but less are done for private colleges. The cost of education in private colleges is much high as compared to those of public institutions. There is no guarantee that qualifying for admission to any college will lead to success in the degree course.

The ultimate aim for us to make any project is to find real world problems that we experience and see in our environment and make a solution that eases and simplifies the problem. We came to understand that result analysis in our college is done manually. We aim to automate the process of opening results, analysing the results and present in clean and organized format with just one click!

The manual existing system of result analysis in universities typically involves the manual collection, recording, and calculation of grades and marks obtained by students in various assessments and exams. The data is then analysed to determine the performance of each student, the overall performance of the class, and the performance of the university as a whole. This is typically done using spreadsheets, paper-based record-keeping systems, and manual calculations. The results are then shared with students, teachers, and university administrators through printed reports or manually generated transcripts.

The present existing system is done manually by the faculty. It is done manually by them. It is Usually done by two persons or more, one dictates the individual result of a single student by going to the website, Typing the hall ticket number, waiting for result to come or they wait for memos and the other person puts values in excel. Then it is manually analysed using different parameters such as

- Class wise Analysis
- Subject wise Analysis
- Personal wise Analysis

The process takes a lot of time and can be frustrating

The existing system is done manually where the faculty members of university analyse it which is time consuming and frustrating. This project is intended to focus on the interaction of different factors and the role that they play to the performance of students in academics in their college or university. The proposed system will have a UI interface which asks you put URL of results link of OU website and hall ticket number range. After clicking submit, it analyses with different parameters and displays all results in organized format automatically.

Proposed System

Problem Statement: To Automate the process of Result Analysis.

The proposed system of result analysis in universities often involves the use of technology, such as computer software and databases, to automate the process of collecting, recording, and analysing student results. This can include:

1. Electronic gradebook software that allows teachers to record and store grades electronically, making it easier to track student progress and performance.
2. Student information systems (SIS) that integrate student data from various sources, such as grades, attendance records, and demographic information, into a single database.
3. Analytics tools that provide data visualization and analysis capabilities, allowing administrators and teachers to easily see trends, patterns, and areas for improvement.

The proposed system will

- Have a user interface created with HTML, CSS, JS
- The interface created is simple, interactive, user friendly and efficient.
- A simple UI that has 3 tabs to fill:
 - Enter URL of results link
 - Initial and Final Hallticket Number
- After entering all the inputs, the user clicks the submit button
- The process redirects to the website URL which was given by the user.
- The code automatically opens the result in the browser
- Type respective hall ticket numbers
- Scrape all the necessary data like Name, Hall ticket number, Subjects, Subjects marks, Subjects grades and overall CGPA displayed on the OU site.
- The unnecessary data is discarded for clean dataset.
- The cleaned and necessary data is saved in the database that is connected
- Here, we use MongoDB as our database.
- The dataset is collected for analysis
- Then MongoDB uses query functions which analyse the data based on parameters we set.
- It then redirects to the frontend with the data presented in an organized format.
- All of this is done in the backend, user is only showed the output.

The proposed system can lead to more efficient, accurate, and timely result analysis, reducing the workload for teachers and administrators, and providing students and parents with real-time access to their academic performance information.

Methodology

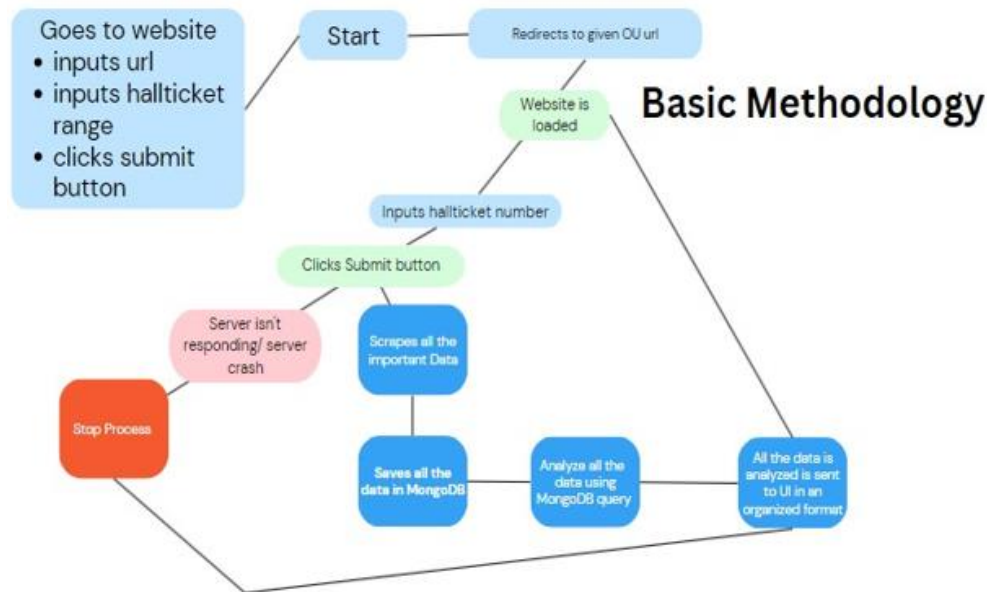


Fig 1: Basic methodology

4.1. Software Components

▪ Puppeteer JS (FRONTEND)

Puppeteer is a Node.js library which provides a high-level API to control Chrome/Chromium over the DevTools Protocol. Puppeteer runs in headless mode by default, but can be configured to run in full (non-headless) Chrome/Chromium.

▪ Express JS (Server Connection)

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js

▪ Node JS (Backend)

Node.js is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more. Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.

▪ Mongo DB (Database)

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License which is deemed non-free by several distributions.

▪ HTML5 (Frontend)

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and final major HTML version that is a World Wide Web Consortium recommendation. The current specification is

known as the HTML Living Standard.

▪ CSS (Frontend)

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

▪ Mongo Query (Backend)

MongoDB Query is a way to get the data from the MongoDB database. MongoDB queries provide the simplicity in process of fetching data from the database, it's similar to SQL queries in SQL Database

▪ Bootstrap (Frontend)

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Implementation

Approach - 1

We initially started off using python. We used Selenium library from python which handles automation. Selenium can easily send standard commands of python to various browsers irrespective of all variation in the design of the browser.

We used Keys library to input hallticket numbers. We used X-path to locate the table and its contents. We used Pandas library to store the data in excel.

Finally, we were able to scrape the data from the website and store it in excel sheet.


```

C: > Users > royal > Desktop > mini project > from selenium import webdriver.py > ...
1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  from selenium.webdriver.common.keys import Keys
4  import options
5  options = webdriver.ChromeOptions()
6  import pandas as pd
7  import time
8  options.add_experimental_option('excludeSwitches', ['enable-logging'])
9  driver = webdriver.Chrome(options=options)
10 result=[]
11 for k in range(160420747001,160420747005):
12     hallticket = k
13     url = "https://www.osmania.ac.in/res07/20221206.jsp"
14     driver=webdriver.Chrome(r"C:\Users\royal\Downloads\chromedriver")
15     driver.get(url)
16     driver.find_element("name", "htno").send_keys(hallticket)
17     driver.find_element("name", "htno").send_keys(Keys.RETURN)
18     res=driver.find_element(By.XPATH,r"//table")
19     print(res.text.split("#"))
20     result.append(res.text)
21     #result=[res.text]
22     # print(result)
23     df=pd.DataFrame(result)
24     df.to_excel("result.xlsx")
25
26
27     #df.merge(df1,how="right")
28     #df.to_excel('resul1.xlsx')
29     time.sleep(7)

```

Fig 2: Approach 1 code

Drawback

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
0		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747001	Gender	FEMALE	Name	FARHA ABDUL WASAY	Father's Name	GULAM MOHAMMED ABDUL WASAY	Course	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade							
1		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747002	Gender	FEMALE	Name	MAHEEN NAAZ	Father's Name	MOHAMMED ZAKRIACourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
2		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747003	Gender	FEMALE	Name	HANIA GHOUSE	Father's Name	GHOUSE BAIG MIRZACourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
3		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747004	Gender	FEMALE	Name	SANA ANJUM	Father's Name	SHAIK HUSSAIN SAHEBCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
4		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747005	Gender	FEMALE	Name	MEHEK FATIMA	Father's Name	BARKAT ALICourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade	Points							
5		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747006	Gender	FEMALE	Name	ARSHI FATIMA	Father's Name	DABEERUDDIN KHAN DANISHCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
6		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747007	Gender	FEMALE	Name	SYEDA ZAINAB FATINA	Father's Name	MIR TARIQ ALI ZAHIDCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
7		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747008	Gender	FEMALE	Name	JUVERIA TANVEEN	Father's Name	MOHAMMED MUSHTAQ AHMEDCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
8		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747009	Gender	FEMALE	Name	AMINAH MASARRATH	Father's Name	MOHD LIYAQAT SHAREEFCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
9		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747010	Gender	FEMALE	Name	AYESHA SIDDIQI	Father's Name	MOHAMMAD MADARCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
10		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747011	Gender	FEMALE	Name	SAFIYA HANI	Father's Name	MOHD AMIADCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade	Points							
11		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747012	Gender	FEMALE	Name	ANAM FATIMA	Father's Name	MOHAMMED NUMAN ATIFCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
12		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747013	Gender	FEMALE	Name	SYEDA AYESHA FATIMA	Father's Name	MIR MUNAWAR ALICourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
13		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747014	Gender	MALE	Name	MOHAMMED JUNAID ADIL	Father's Name	MOHAMMED ABDUL MOJIBCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
14		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747015	Gender	MALE	Name	SYED SARDARUDDIN	Father's Name	SYED NAIMUDDINCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
15		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747016	Gender	MALE	Name	MUSTAFA UMAIR	Father's Name	MD SAFEED UDDINCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade	Points							
16		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747017	Gender	MALE	Name	MOHAMMED JALALUDDIN YAZDANI	Father's Name	MOHAMMED NOORCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
17		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747018	Gender	MALE	Name	MUHAMMAD UMAR KHAN	Father's Name	MOHAMMED VASEEM KHANCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
18		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747019	Gender	MALE	Name	MOHAMMED AHMED HUSSAIN	Father's Name	MOHAMMED ILYAS HUSSAINCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
19		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747020	Gender	MALE	Name	OSMANI MIR ABED ALI	Father's Name	MIR TAREQ ALICourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade	Points							
20		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747021	Gender	MALE	Name	MOHD ZAIN UDDIN JAMAL	Father's Name	MOHD ANSARICourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
21		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747022	Gender	MALE	Name	MOHAMMED ABDUL RAHMAN	Father's Name	MOHAMMED ZAMEERUDDINCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
22		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747023	Gender	MALE	Name	MOHAMMED ABDUL FAROOQUI	Father's Name	MOHD SAJID WALI FAROOQUICourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
23		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747024	Gender	MALE	Name	SYED ABDUL MUBASHIR	Father's Name	SYED ABDUL KHADERCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
24		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747025	Gender	MALE	Name	DESAI SAI PRANEETH REDDY	Father's Name	DESAI NARENDRANATH REDDYCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
25		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747026	Gender	MALE	Name	MOHAMMED ABDUL MUNAF	Father's Name	ABDUL WAHED ABDUL BAQIICourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								
26		BE(CBCS) SEPT-2022 RESULTS07-Dec-2022Personal DetailsHall Ticket No.	160420747027	Gender	MALE	Name	MIR RAHIMUDDIN AHMED KHAN	Father's Name	JAVEED AHMED KHANCourse	BE(CBCS) Medium	Marks	DetailsSub Code	Subject Name	Credits	Grade								

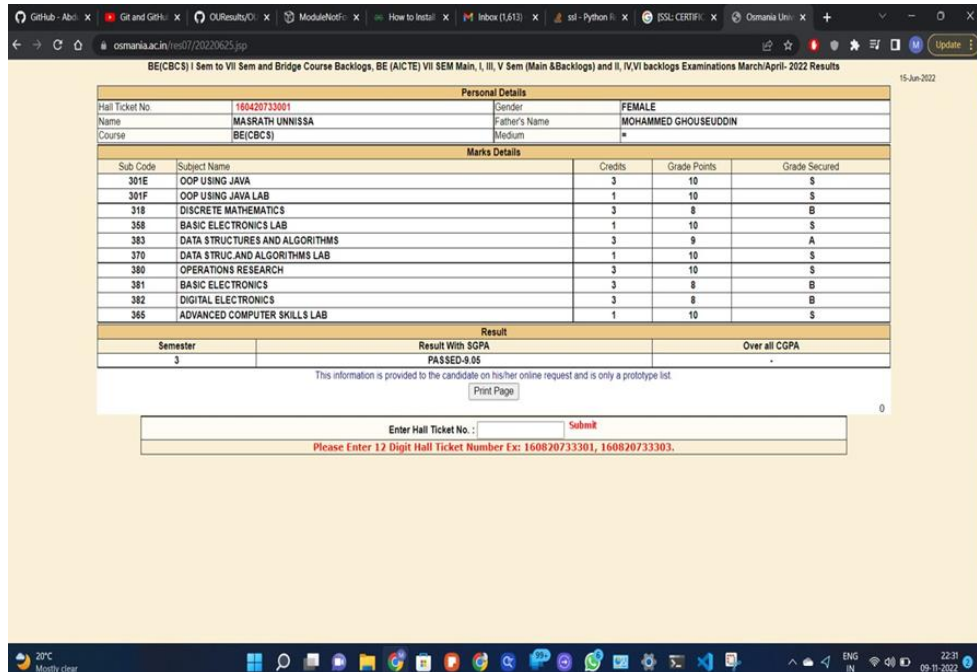
Fig 3: Approach 1 drawback

The major drawback of this approach was

- The data stored in excel was in string format.
- The data which were scraped into excel lacked spaces, commas and new lines.
- It was impossible to analyse as one whole result was in one excel cell.
- We were not able to remove the irrelevant part which is not related to the result analysis. Ex: Father name, Medium etc.
- We were not able to manipulate the excel sheet automatically using python
- We were not able to present this on frontend UI as we were unable to analyse and extract the relevant part of the information.
- Overall, the dataset wasn't clean and this approach failed.

Approach - 2

- Firstly, the OU website link and hallticket range is taken from the user.
- It goes to the backend of the website
- Here it automatically redirects to the link provided
- After the site is loaded, it finds the hall ticket input
- Now, it enters into loop of the hall tickets which the user has provided
- It enters the hallticket no.
- Here, the database is connected.
- It presses the submit button.
- Now, it only scrapes the important information which is required to analyse.
- It stores the important data in the database that is connected (here we use MongoDB).

**Fig 4:** Result URL opens automatically

- The figure 5.3 is a snap of the OU Result portal which is opened automatically through puppeteer library.
- The scraping is done of all the hallticket numbers in loop similarly.

Code Snippets

```

async function loadAndExtractData(hallticketNumber, page) {
  console.log('going to page...');
  await page.goto("https://www.osmania.ac.in/res07/20220711.jsp");
  console.log('page loaded');
  // await page.screenshot({ path: 'example2.png' });
  const ht = await page.waitForXPath("//*[@id='AutoNumber6']/tbody/tr[1]/td/b/font/input[1]");
  await ht.click();
  await ht.type(hallticketNumber);
  const submit = await page.waitForSelector('input[type=image]');
  await Promise.all([
    submit.click(),
    page.waitForNavigation({waitUntil: 'networkidle2'})
  ]);
  // await someTime(5 * 60);
  const trs3 = await page.$$('#AutoNumber3 > tbody > tr'); // student name
  const trs4 = await page.$$('#AutoNumber4 > tbody > tr'); // marks
  const studentName = await extractName(trs3, page);
  // console.log('hall ticket number is ' + hallticketNumber)
  // console.log("student name is " + studentName);
  const codes = await printTrContent(trs4, page);
  // console.log("codes is " + codes);
  const subject= await printSubject(trs4,page)
  // console.log("Subjects are " + subject);
  const subjectMarks= await printSubjectMarks(trs4,page);
  // console.log("Subject Marks are " + subjectMarks);
  const subjectGrades= await printSubjectGrades(trs4,page);
  // console.log("Subject Grades are " + subjectGrades);
}

```

Fig 5: Code Snippet 1

The figure is the part of the code that is responsible for automating the redirection to the result URL, entering the hall

ticket numbers and scraping the useful data.

```
const Score = mongoose.model('Score', {
  hallticketNumber: String,
  studentName: String,
  code: String,
  subject: String,
  marks: String,
  grade: String,
});

async function saveToMongoDb(data) {
  for (let i = 0; i < data.codes.length; i++) {
    const code = data.codes[i];
    const subject = data.subject[i];
    const subjectMarks = data.subjectMarks[i];
    const subjectGrades = data.subjectGrades[i];
    const score = new Score({
      hallticketNumber: data.hallticketNumber.trim(),
      studentName: data.studentName.trim(),
      code: code.trim(),
      subject: subject.trim(),
      marks: subjectMarks.trim(),
      grade: subjectGrades.trim(),
    });
    await score.save();
    console.log("saved");
  }
  console.log(data);
}
```

Fig 6: Code snippet 2

- The figure is the part of the code that establishes the connection to the database
- The data scraped is stored in the database
- MongoDB was used for the database

```
const puppeteer = require('puppeteer');
const fs = require('fs/promises');
const mongoose = require('mongoose');

async function start() {
  const halltickets = ["094205010001", "094205010002", "094205010003", "094205010004", "094205010005", "094205010006"];
  mongoose.connect('mongodb://127.0.0.1:27017/scores');
  const browser = await puppeteer.launch({
    headless: false,
  });
  const page = await browser.newPage();

  for (let i=0;i<halltickets.length;i++){
    const data = await loadAndExtractData(halltickets[i], page);
    await saveToMongoDb(data);
  }

  await browser.close();
}
```

Fig 7: Code snippet 3

The figure is the part of the code that after visiting the URL, goes in loop until all the required results are scraped.

Async function: It simply allows us to write promises-based code as if it was synchronous and it checks that we are not breaking the execution thread. It operates asynchronously via the event loop. Async functions will always return a value. Here Puppeteer JS is asynchronous so we use await function. Puppeteer is a Node library which provides a high-level API to control Chromium or Chrome over the DevTools Protocol. The Puppeteer API is hierarchical and mirrors the browser

structure. NOTE On the following diagram, faded entities are not currently represented in Puppeteer.

The benefit of using Mongoose is that we have a schema to work against in our application code and an explicit relationship between our MongoDB documents and the Mongoose models within our application. The downside is that we can only create blog posts and they have to follow the above defined schema.

The entire data is scraped of all the hall ticket numbers in the loop that was entered in the loop condition.

Results & Output



Fig 8: UI of Exactscore

The figure 6.1 is the user interface of the application EXACTSCORE. It was made using HTML, CSS, JS.

```
{
  hallticketNumber: '094205010004',
  studentName: 'DUPALLY RUSHALI',
  codes: [
    '401', '402',
    '403', '415',
    '405', '406',
    '404'
  ],
  subject: [
    'TECH. MGMT.& ENTREPRENEURSHIP',
    'STRATEGIC MANAGEMENT',
    'INTERNATIONAL BUSINESS',
    'PROJECT(GRADE)',
    'PERFOR. APPRAISAL&COUNSELLING',
    'INDUSTRIAL REL.& LABOUR.LAWS',
    'SUPPLY CHAIN MANAGEMENT'
  ],
  subjectMarks: [
    '56', '39',
    'ABS', 'A+',
    '57', '50',
    '52'
  ],
  subjectGrades: [
    'PASS', 'PASS',
    'ABSENT', 'PASS',
    'PASS', 'PASS',
    'PASS'
  ]
}
```

Fig 9: Scraped data in terminal

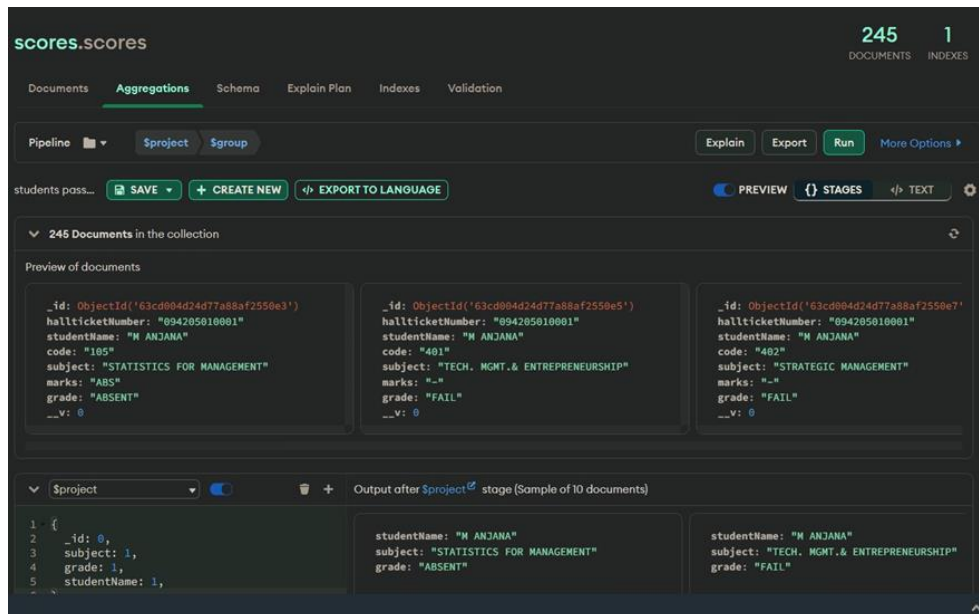


Fig 10: Database

- The figure is the snap of the database.
- All the data that was scraped is stored in this database.
- Only the parts necessary for analysis were scraped
- Each subject scraping is an different entry in the database
- MongoDB was used for the database
- MongoDB is a non-relational document database that provides support for JSON-like storage. The MongoDB database has a flexible data model that enables you to store unstructured data, and it provides full indexing support, and replication with rich and intuitive APIs.
- MongoDB is designed to make data easy to access, and rarely to require joins or transactions, but when you need to do complex querying, it's more than up to the task. The MongoDB Query API allows you to query deep into documents, and even perform complex analytics pipelines with just a few lines of declarative code.

Conclusion and Future Work

In this study, we have analysed the result of students in order to formulate the future teaching pedagogy on the basis of students' performance

Our project aims to ease the process of result analysis and evaluation. We plan to create a platform that all OU affiliated colleges can use.

For the purpose of conducting this analysis we have used Puppeteer which is a framework of Javascript. Along with that we have also used MongoDB for storing the data that was scraped to analyse.

1. Demographic breakdown: A breakdown of the results by demographic factors such as gender, ethnicity, and socio-economic status can provide insights into any disparities in academic achievement.
2. Course-specific analysis: An in-depth analysis of results by course can help identify strengths and weaknesses in the university's academic offerings and highlight areas that may require improvement.
3. Student satisfaction: Analysis of student satisfaction surveys can provide valuable information on the university's performance and help identify areas for improvement.
4. Comparison with peer institutions: Comparison of the

university's results with those of similar institutions can provide a benchmark for performance and highlight areas for improvement.

5. Recommendations for improvement: Based on the analysis, recommendations can be made for actions that the university can take to improve results and support student success.

Future Work: We plan to create a dashboard that would have different tabs showing various options like top10, list of failures in particular subjects etc. which would make student life easier and let them focus more where there is need for.

1. Streamlined data collection and reporting: The proposed system would automate the process of collecting and recording student results, reducing the risk of errors and omissions. This would also make it easier to generate reports, such as transcripts and academic records, for students, parents, and administrators.
2. Improved data accuracy and integrity: By using a centralized database and automated data entry processes, the proposed system would ensure that student results are recorded accurately and consistently. This would reduce the risk of data errors and ensure the integrity of the results.
3. Enhanced data security: The proposed system would use secure servers, firewalls, and data encryption to protect student data from unauthorized access and theft. This would ensure that sensitive information, such as grades and personal details, are kept confidential.
4. Improved decision-making: With access to real-time data, teachers and administrators would be able to quickly identify trends and patterns in student performance. This would allow them to make informed decisions about teaching methods, curriculum development, and student support.
5. Increased efficiency and productivity: The proposed system would automate many of the manual processes involved in result analysis, freeing up time for teachers and administrators to focus on other tasks. This would increase efficiency and productivity and allow universities to allocate their resources more effectively.

The proposed system of result analysis in universities would provide a more efficient, accurate, and secure method for analysing student results, which would help to improve the quality of education and support student success.

Overall, the conclusion should present a clear picture of the university's results and provide recommendations for future improvement

Perhaps the essence of human evolution is to automate manual tasks.

References

1. Here are a few research paper references on the topic of result analysis of university students:
2. A Study of Student Performance Analysis Using Data Mining Techniques by S. Bhatia and V. Bhatia, 2011.
3. An Exploration of Student Performance Analysis and Prediction using Machine Learning Techniques by B. Lee and J. Kim, 2015.
4. Data Mining Techniques for Student Performance Prediction in Higher Education by J.A.K. Suhail and M.A. Al-Jumeily, 2015.
5. An Empirical Study of Student Performance Analysis in Higher Education by K.R. Gouda and S.B. Patil, 2013.
6. Student Performance Analysis: A Review of the Literature by M.S. Chhabra and P. Chhabra, 2012.
7. An academic results analysis of a first year interdisciplinary project approach.