|  |  |
| --- | --- |
|  | **INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY** |
|  | **FACULTY OF INFORMATION TECHNOLOGY** |

**A logo with a red and blue letter and a black background

Description automatically generatedREPORT – BUILDING THE EDUMALL DATA COLLECTION FLOW**

**Member:**

|  |  |  |
| --- | --- | --- |
| **No.** | **Student name** | **Student ID** |
| 1 | Lê Thị Kim Oanh | 22632471 |
| 2 | Nguyễn Duy | 22708301 |
| 3 | Lê Đặng Phương Trinh | 22636871 |
| 4 | Nguyễn Huy Khang | 22710921 |
| 5 | Đào Trung Kiên | 22635961 |
| 6 | Trần Thị Mỹ Uyên | 22718671 |

**Class No.:** DHHTTT18BTT

**Start Date:** 10/11/2024

**End Date:** 22/11/2024

**Name of the Coordinator:** MR. Huỳnh Nam

**Date of Submission:** 22/11/2024

**MENU**

[I. DATA COLLECTION AND STOGARE IN MONGODB 4](#_Toc183128195)

[II. DATA PROCESSING WITH PYSPARK 11](#_Toc183128196)

[III. DATABASE MODELLING 12](#_Toc183128197)

[IV. DESIGN OF THE DATA ANALYSIS PROBLEM 15](#_Toc183128198)

[V. OPTIMIZING THE MODEL WITH A DATA WAREHOUSE 16](#_Toc183128199)

[VI. FINAL PRODUCT 20](#_Toc183128200)

[**1.** **A galaxy schema data warehouse** 20](#_Toc183128201)

[**2.** **Dashboards and analysis** 22](#_Toc183128202)

1. **DATA COLLECTION AND STOGARE IN MONGODB**

* Set up Selenium WebDriver: Launch the Chrome browser using ChromeDriver to automatically access the list of courses on the Edumall.vn website and retrieve detailed information about each course. Use Selenium to bypass restrictions against data collection by the website administrators.
* Access the course list: Collect course links and save them to the file list\_links\_full2.txt using the find\_elements method. This will allow you to extract all the necessary information later.
* Iterate through all course pages: Create a for loop to navigate through all the course pages and collect data. Each page will be fully loaded with its HTML source and all available element XPaths for data extraction.
* Loop through the course list: Create another for loop to iterate through all the courses in the list and collect data for each course.
* Use WebDriverWait: Ensure that data is fully loaded before extraction by using the WebDriverWait function. This avoids errors caused by elements not loading completely.
* Below is the content of the source code file edumall\_selenium.py:

A computer screen with text on it

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generated



A computer screen shot of text

Description automatically generated

* Insert Data into MongoDB: For each piece of data collected from the website, use PyMongo to connect to MongoDB. The data will be inserted into MongoDB in JSON format using the insert\_many () functions for faster and more efficient storage. If an insertion fails, the process will skip to the next data row using a try\_except block.

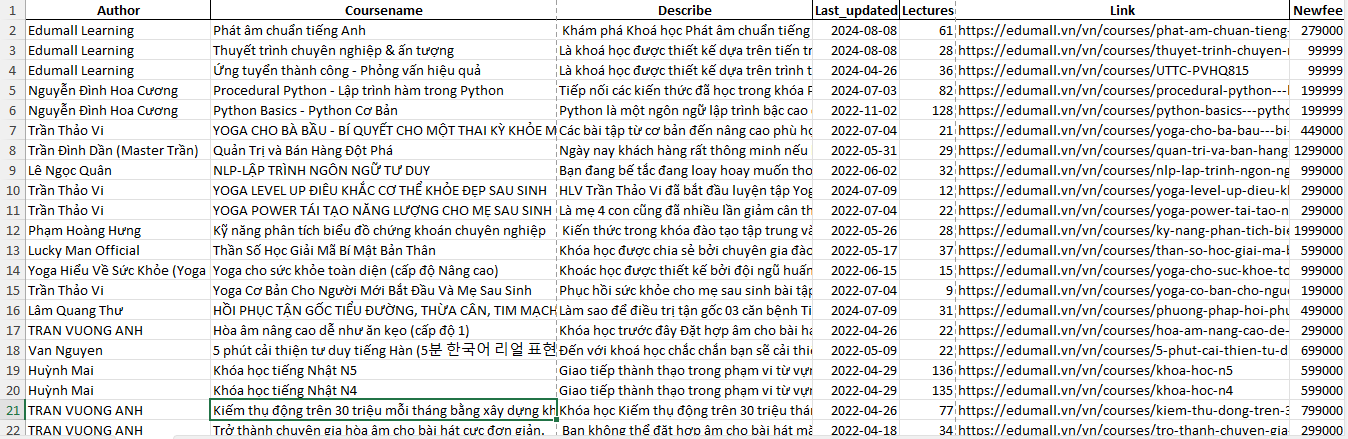
A computer screen shot of a black background

Description automatically generated with medium confidence

A screen shot of a computer code

Description automatically generated

* Results Obtained: The courses are collected with detailed information, including course name, instructor, topic, description, learning outcomes, old price, new price, last updated date, number of sections, number of lectures, duration, and course link. These details are extracted from the course link lists across all pages and saved in both CSV and JSON files. Additionally, the data is stored in MongoDB to enable flexible data retrieval and scalability in storage.
* *File CSV:*



* *File JSON:*

A screenshot of a computer program

Description automatically generated

1. **DATA PROCESSING WITH PYSPARK**

* Initialize a Spark Session: Set up a Spark session to begin processing the data.
* Connect to MongoDB: Establish a connection to read data directly from MongoDB using Spark's MongoDB connector.
* Remove Unwanted Characters: Use the regexp\_replace function to clean the data. Regular expressions are applied to identify and replace special characters, ensuring cleaner data while maintaining the original data type of each column.
* *The source code for the file "edumall\_cleaning.py*

A screen shot of a computer program

Description automatically generated



* *Dockerfile to build the Spark container*

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

1. **DATABASE MODELLING**

**Create a relational model for the tables and then store it in PostgreSQL:**

* Courses: Course Name, Description, Old Price, New Price, Rating, Duration, Last Updated Date, Last Updated Month, Last Updated Year, Number of Sections, Number of Lectures, Learning Outcomes, Instructor ID, Topic ID.
* Topics: Topic ID, Topic Name.
* Instructors: Instructor ID, Instructor Name.
* Source Code for "edumall\_postgresql.py":

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

* *The content of the Dockerfile to build a Postgres container:*

A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

1. **DESIGN OF THE DATA ANALYSIS PROBLEM**

Problem: Factors affecting the course price.

A computer code on a dark background

Description automatically generated

A computer code with text

Description automatically generated with medium confidence

A screen shot of a computer code

Description automatically generated

A computer screen shot of a computer code

Description automatically generated with medium confidence

A computer code on a dark background

Description automatically generated

A computer screen shot of a blue and white text

Description automatically generated with medium confidence



1. **OPTIMIZING THE MODEL WITH A DATA WAREHOUSE**

* *Source code of the file Create\_SQLServer.py, used to initialize tables in the Data Warehouse:*

**A screen shot of a computer

Description automatically generated**

A screen shot of a computer program

Description automatically generated

* *Source code of Postgre\_to\_SQLServer.py, used to transfer data from PostgreSQL to SQL Server.*

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

1. **FINAL PRODUCT**
2. **A galaxy schema data warehouse**

* The data warehouse can provide additional information about the relationships between courses, topics, authors, and update times.

A screenshot of a computer

Description automatically generated

* Use the "Select ... from ..." query to check whether the data has been successfully inserted into the data warehouse.

A screenshot of a computer

Description automatically generated

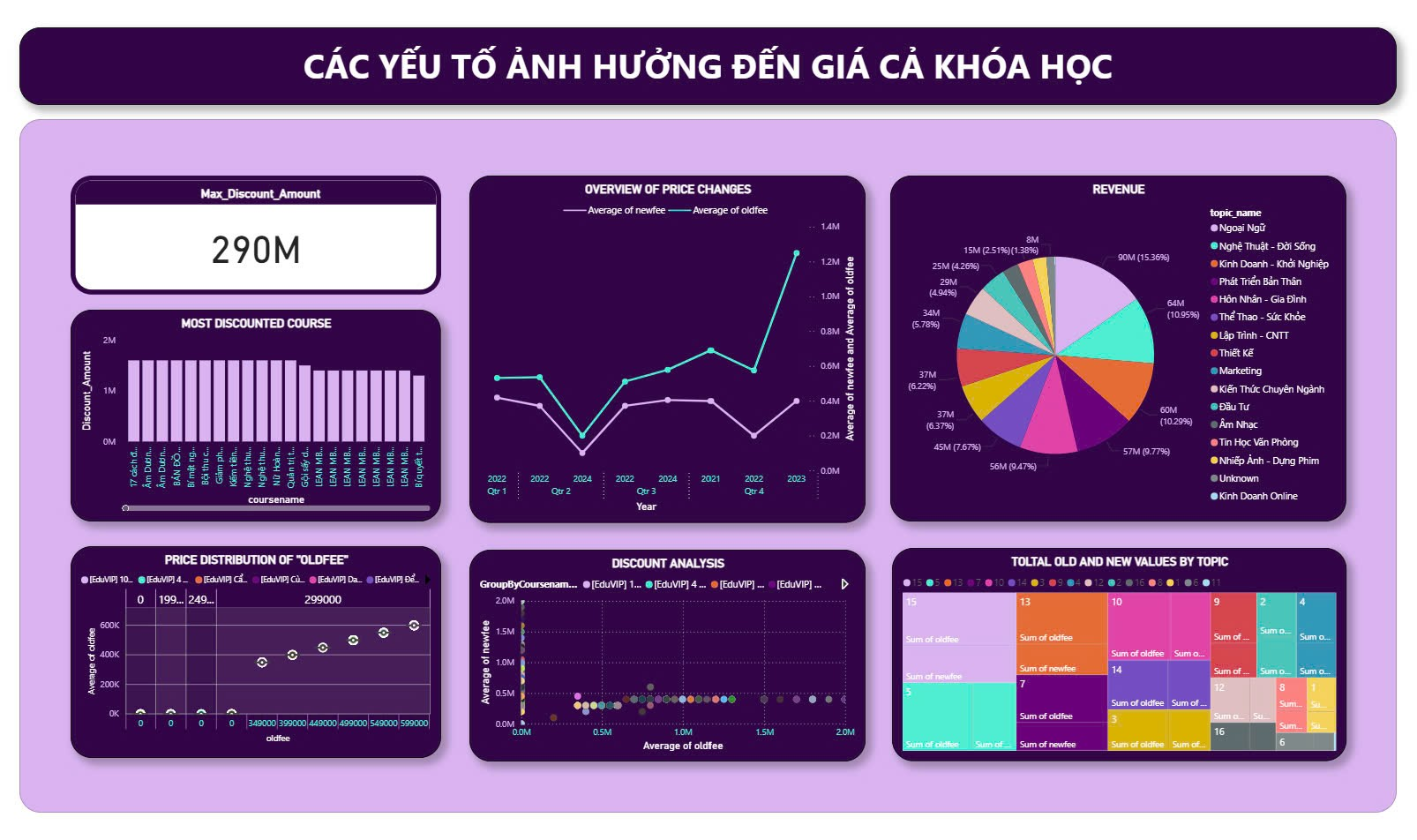
* After that, proceed to the next step to create the data cube database.

A screenshot of a computer

Description automatically generated

1. **Dashboards and analysis**

* *Overview of price changes*



1. **Max\_Discount\_Amount (Card Visual)**

**Meaning:** Display the highest discount value of a course, represented by a specific number (290M). This value is calculated using the Measure Max\_Discount\_Amount.

**Objective:** To help users quickly identify the course with the highest discount.

1. **Overview of Price Changes (Line Chart)**

**Meaning:** Track the average price changes (newfee and oldfee) by quarter and year. The blue line represents the average newfee, while the purple line represents the oldfee.

**Objective:** Compare the trends of old and new prices over time, in order to identify significant price changes.

1. **Revenue (Pie Chart)**

**Meaning:** The pie chart displays revenue from the new price (newfee) distributed by topic (topic\_name). The tooltip can show the total revenue and the percentage share of each topic.

**Objective:** Identify the topics that generate the highest revenue. Analyze the revenue share between different topics.

1. **Most Discounted Course (Column Chart)**

**Meaning:** The bar chart ranks courses by discount value (Discount\_Amount). The tallest bar represents the course with the highest discount.

**Objective:** Find the course with the highest discount. Compare the discount levels between different courses.

1. **Price Distribution of "Oldfee" (Boxplot)**

**Meaning:** The boxplot displays the distribution of the old price (oldfee). Metrics such as Q1, Median, Q3, Min, and Max are calculated and shown to illustrate variations in the data.

**Objective:** Analyze the old prices of courses to identify outliers and the degree of dispersion. Discount Analysis (Scatter Plot)

1. **Total Old and New Values by Topic (Treemap)**

**Meaning:** The treemap displays the total value of oldfee and newfee distributed by topic (topic\_name). The tiles are hierarchically arranged based on the value proportion, and the tooltip can provide details for each topic.

**Objective:** Analyze the difference between old and new prices by topic. Identify the topic with the highest total value.