Comp3111 QSProject report

Lam Pak Lung 20869543

repo link : https://github.com/lunglam/Comp3111Project/tree/master

Additional features for the QS project

1. **University Tier**
2. **website and logo hyperlink**
3. **new tabs for only Tier A University**

1. University Tier

- In response to a user-friendly ranking system, universities are categorized into different tiers based on their scores for a given year.
- The scores are analyzed using normal distribution. The grading system is shown below :

```
*    grading system:
*          >2sd -> A+
*          >1.6sd -> A
*          >1sd -> A-
*          >0.6sd -> B+
*          >0sd -> B
*          >-0.6sd -> B-
*          >-1sd -> C+
*          >-1.6sd -> C
*          >-2sd -> C-
*          below 2sd -> D
```

- The tier is shown in the datatable. Hence users can see how each university’s score deviates from the average, providing insights into its relative performance.

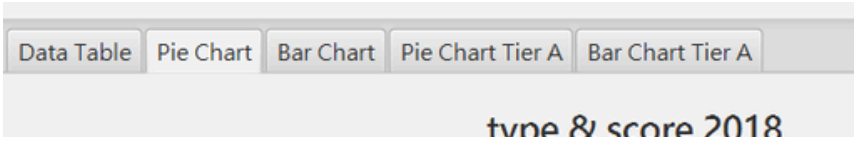
Rank	University	Score	Country	City	Type	Score Tier	Logo	Website
1	Massachusetts Institute of ...	100	United States	Cambridge	Private	A+	logo	link
2	Stanford University	98.6	United States	Stanford	Private	A+	logo	link
3	Harvard University	98.5	United States	Cambridge	Private	A+	logo	link

2. Website and logo hyperlink

- Website and logo can now easily shown by clicking the hyperlink in datatable
- This makes the QS program more user friendly and the user can easily check detail information for that university from their website.

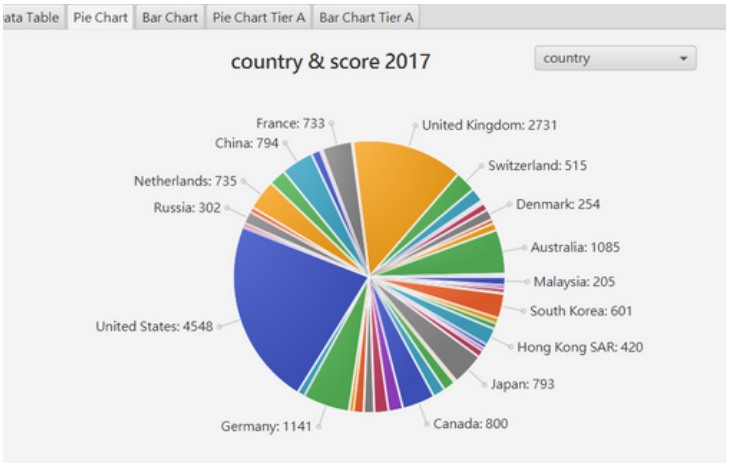
3. new tabs for only Tier A University

- In response to the tier assigned to each university, a specialized feature has been implemented: two distinct tabs dedicated exclusively to Tier A institutions. These tabs serve the purpose of showcasing data related to the world top-ranking universities.
- Tier A includes A+, A, A-

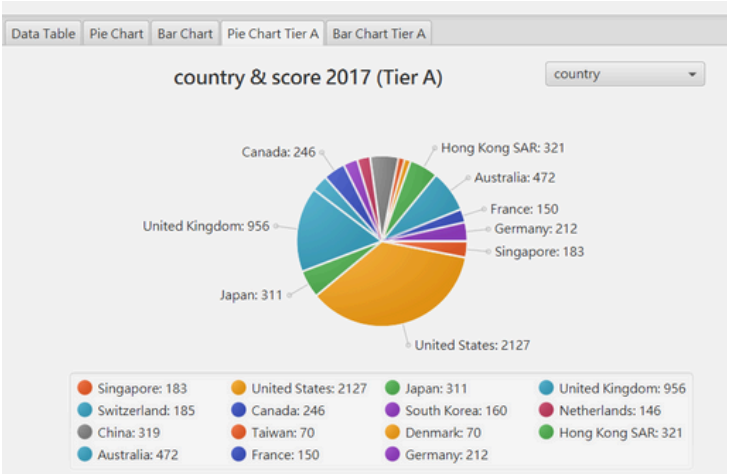


- These designations represent the highest echelon of academic excellence, signifying outstanding performance
- By focusing on Tier A universities, users gain access to a curated selection of institutions renowned for their exceptional achievements.
- Also, users can easily compare the data of Tier A universities against other tiers. This facilitates informed decision-making during the university selection process.
- For example, in the country category, the distribution for original chart and the chart Tier A have great difference

Pie chart for country in 2017

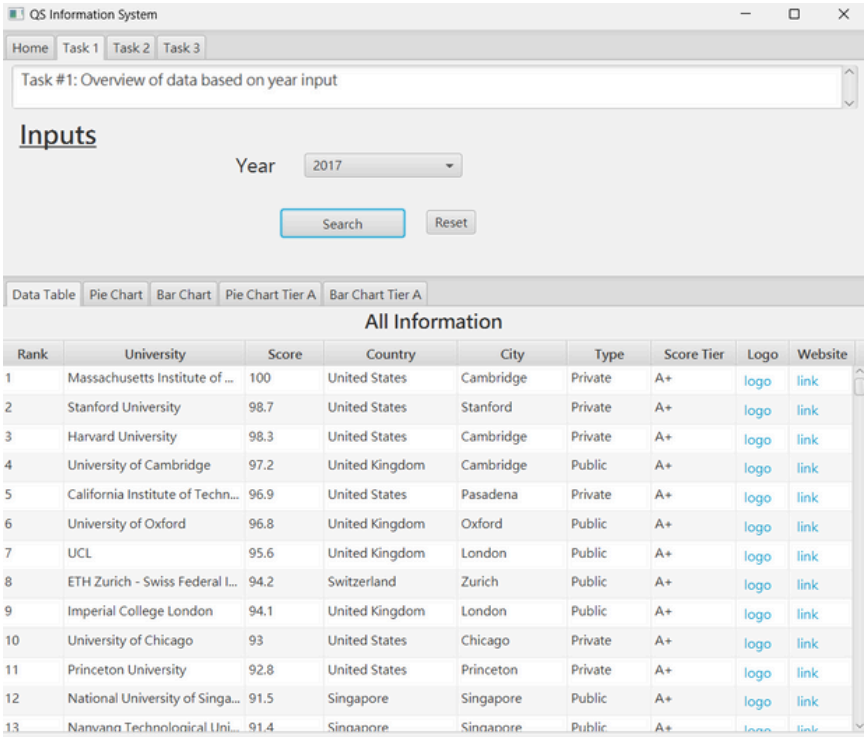


Tier A Pie chart for country in 2017



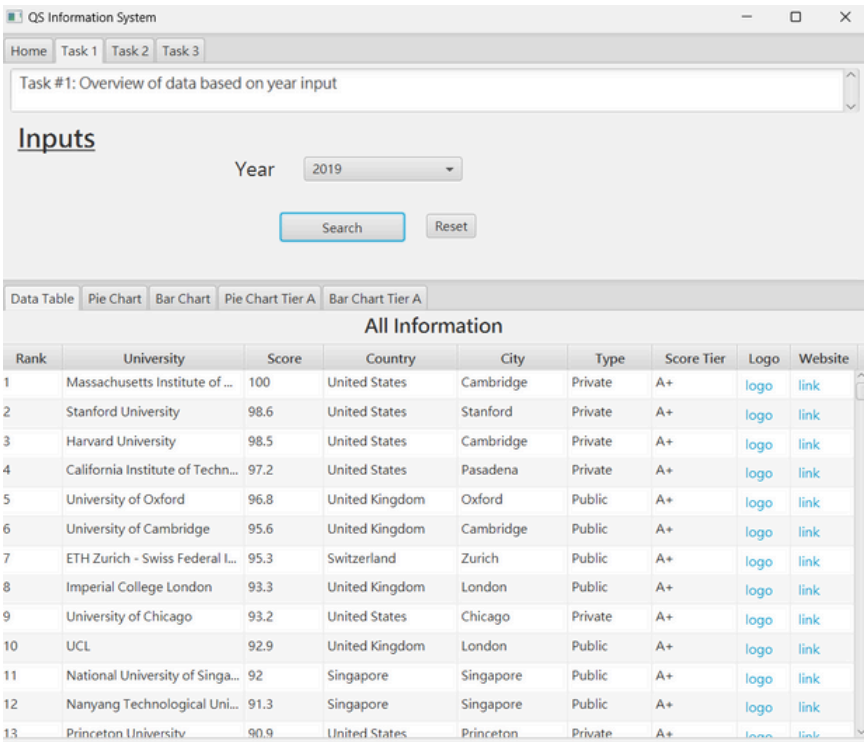
Screenshots of the execution

Datatable : search for 2017



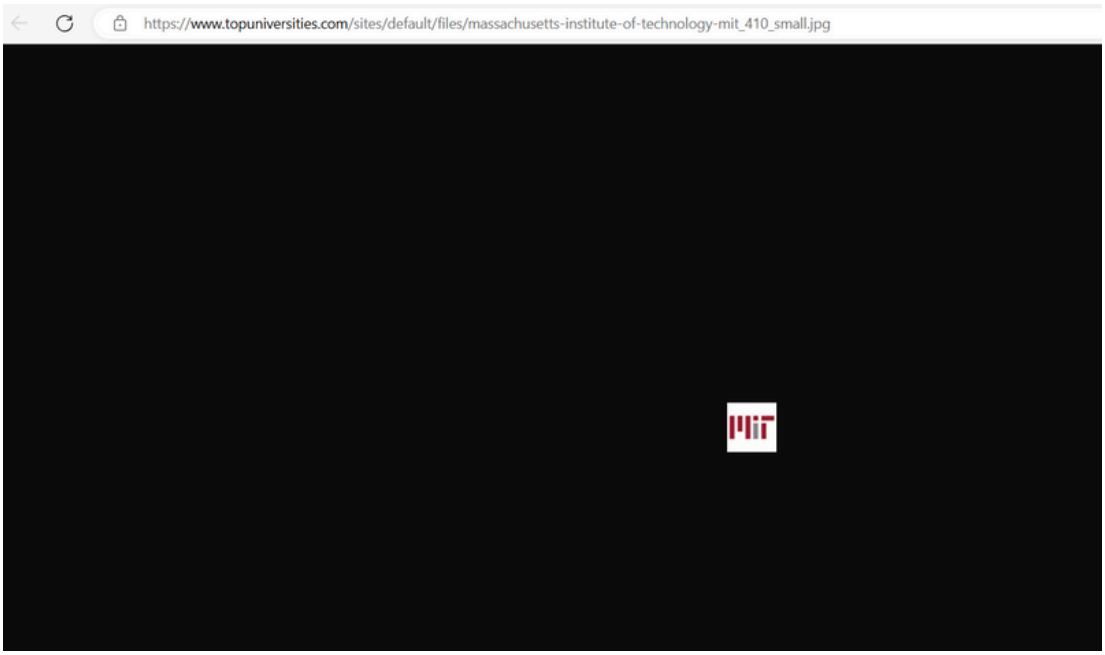
Rank	University	Score	Country	City	Type	Score Tier	Logo	Website
1	Massachusetts Institute of ...	100	United States	Cambridge	Private	A+	logo	link
2	Stanford University	98.7	United States	Stanford	Private	A+	logo	link
3	Harvard University	98.3	United States	Cambridge	Private	A+	logo	link
4	University of Cambridge	97.2	United Kingdom	Cambridge	Public	A+	logo	link
5	California Institute of Techn...	96.9	United States	Pasadena	Private	A+	logo	link
6	University of Oxford	96.8	United Kingdom	Oxford	Public	A+	logo	link
7	UCL	95.6	United Kingdom	London	Public	A+	logo	link
8	ETH Zurich - Swiss Federal I...	94.2	Switzerland	Zurich	Public	A+	logo	link
9	Imperial College London	94.1	United Kingdom	London	Public	A+	logo	link
10	University of Chicago	93	United States	Chicago	Private	A+	logo	link
11	Princeton University	92.8	United States	Princeton	Private	A+	logo	link
12	National University of Singa...	91.5	Singapore	Singapore	Public	A+	logo	link
13	Nanyang Technological Uni...	91.4	Singapore	Singapore	Public	A+	logo	link

Datatable : search for 2019

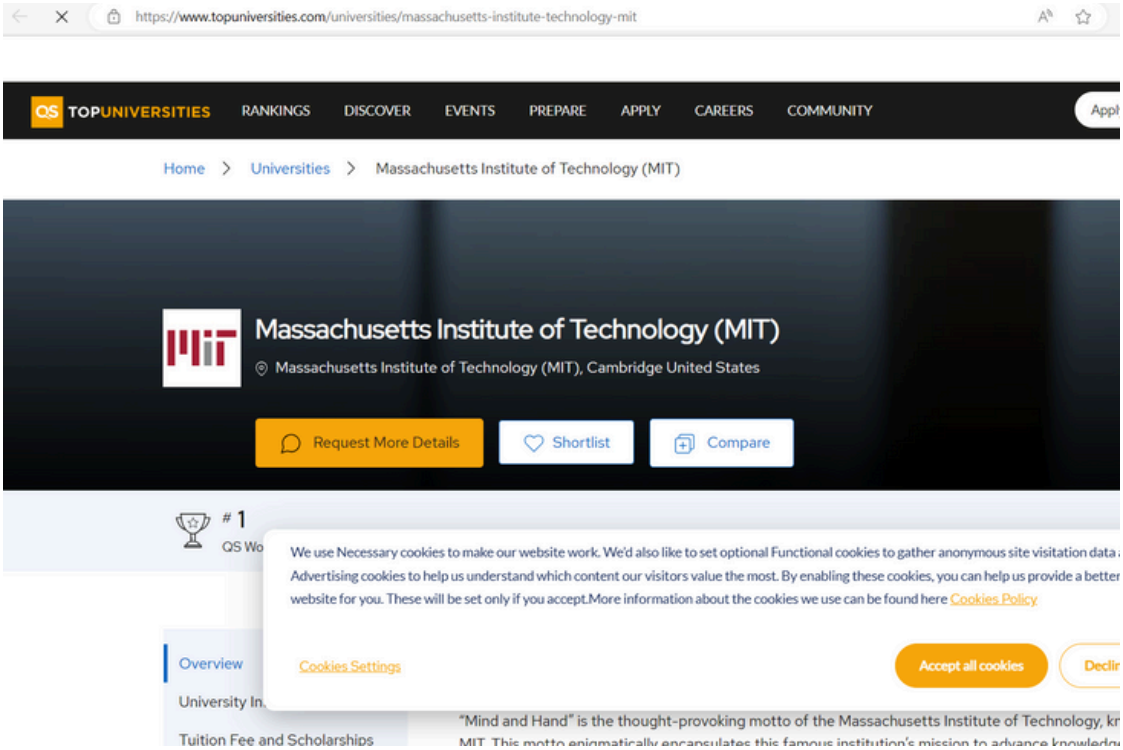


Rank	University	Score	Country	City	Type	Score Tier	Logo	Website
1	Massachusetts Institute of ...	100	United States	Cambridge	Private	A+	logo	link
2	Stanford University	98.6	United States	Stanford	Private	A+	logo	link
3	Harvard University	98.5	United States	Cambridge	Private	A+	logo	link
4	California Institute of Techn...	97.2	United States	Pasadena	Private	A+	logo	link
5	University of Oxford	96.8	United Kingdom	Oxford	Public	A+	logo	link
6	University of Cambridge	95.6	United Kingdom	Cambridge	Public	A+	logo	link
7	ETH Zurich - Swiss Federal I...	95.3	Switzerland	Zurich	Public	A+	logo	link
8	Imperial College London	93.3	United Kingdom	London	Public	A+	logo	link
9	University of Chicago	93.2	United States	Chicago	Private	A+	logo	link
10	UCL	92.9	United Kingdom	London	Public	A+	logo	link
11	National University of Singa...	92	Singapore	Singapore	Public	A+	logo	link
12	Nanyang Technological Uni...	91.3	Singapore	Singapore	Public	A+	logo	link
13	Princeton University	90.9	United States	Princeton	Private	A+	logo	link

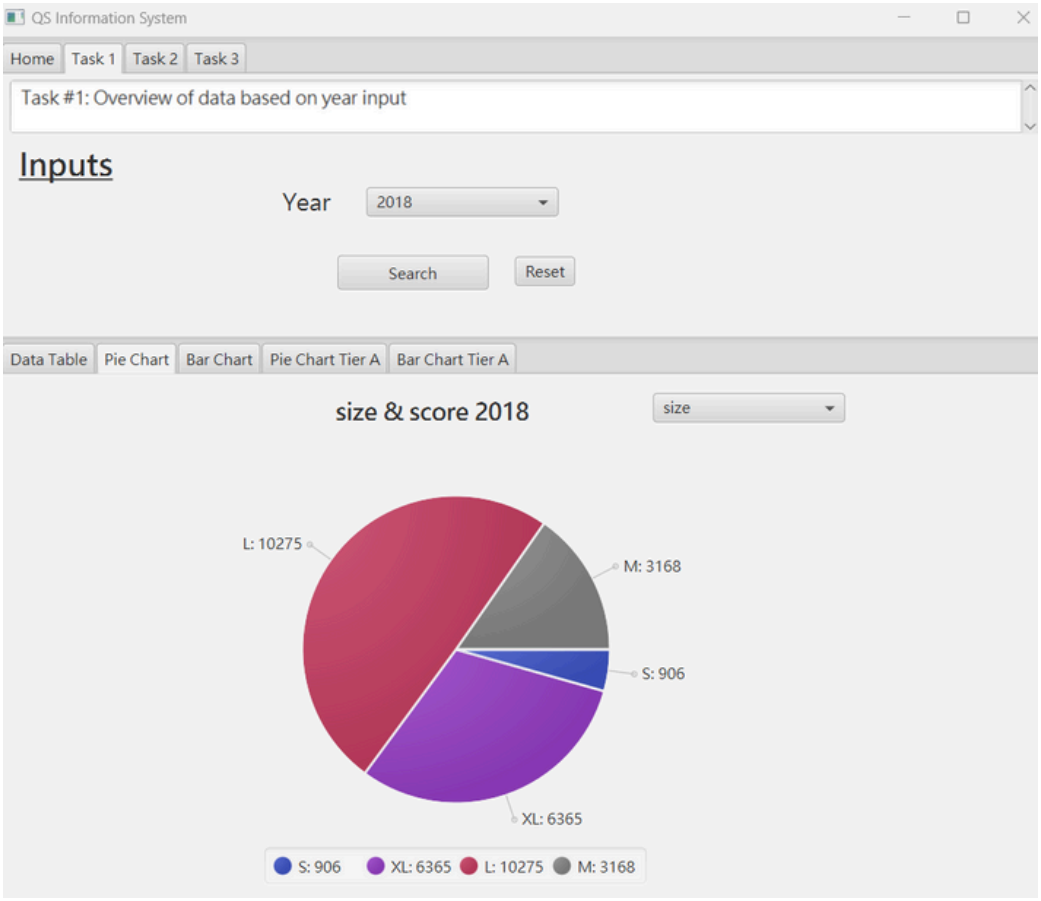
Datatable : hyperlink for universitiy logo



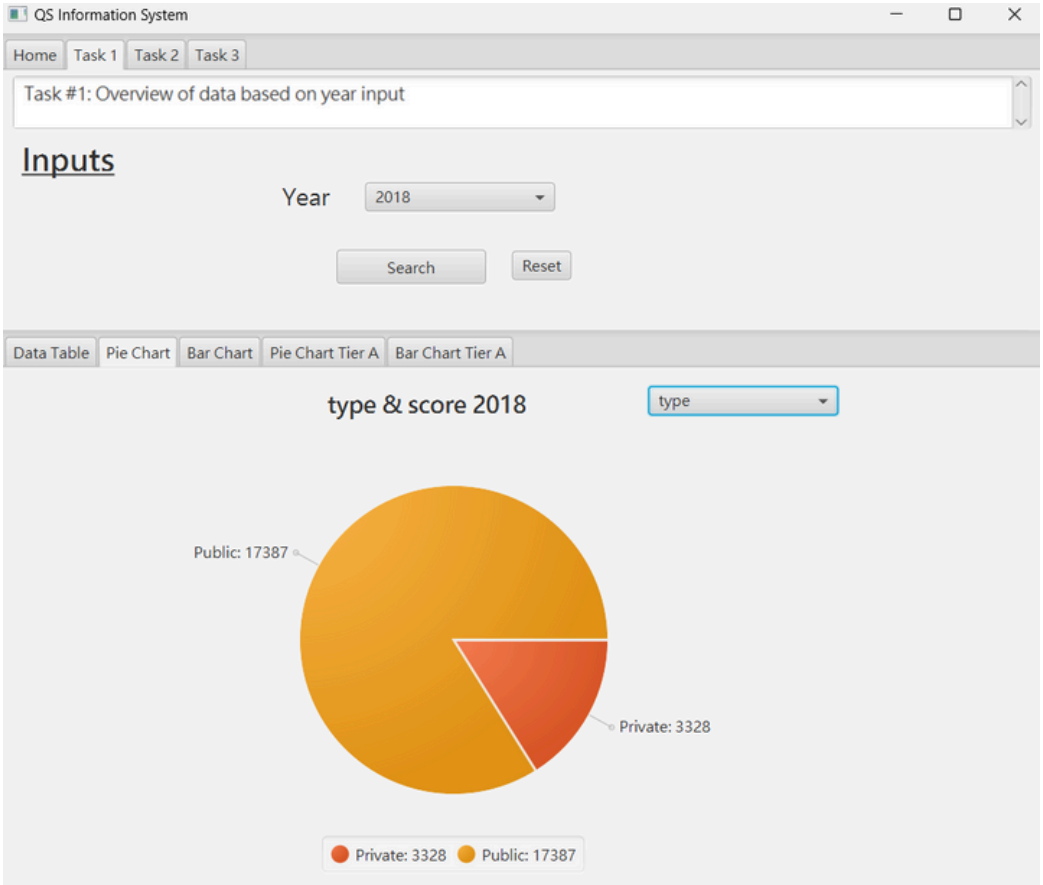
Datatable : hyperlink for university website



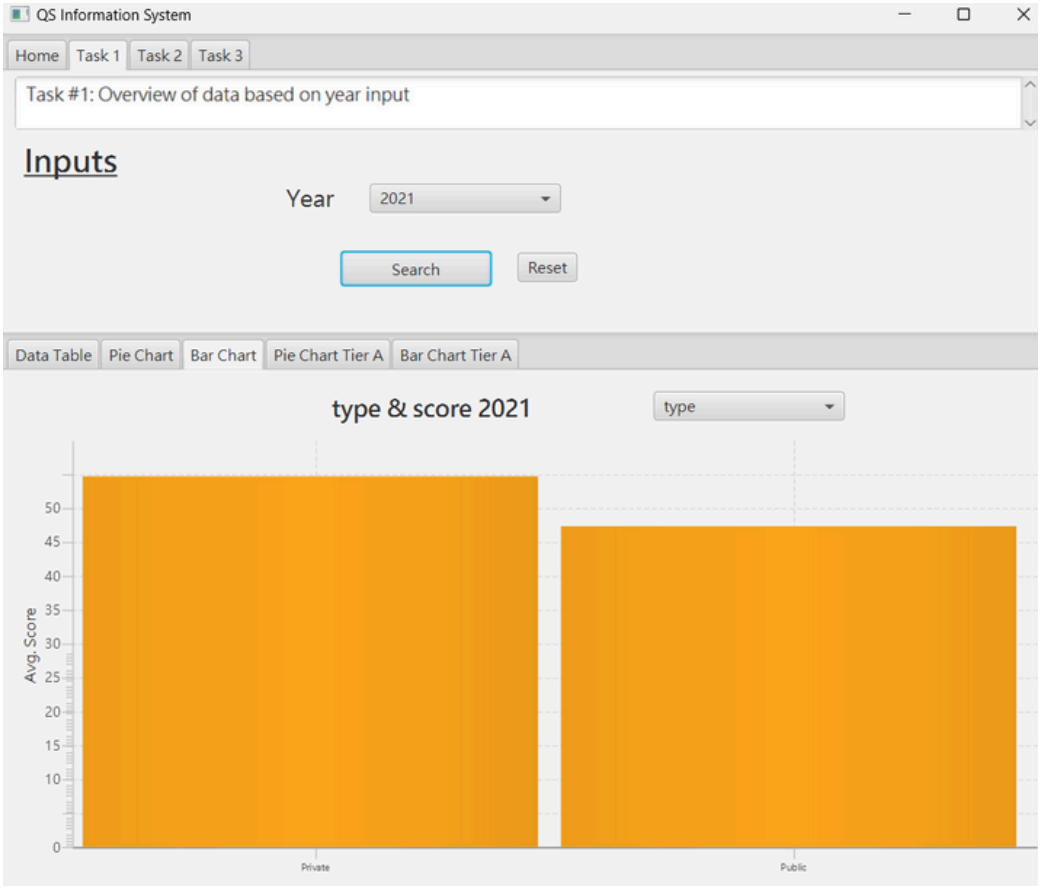
Piechart : search by size



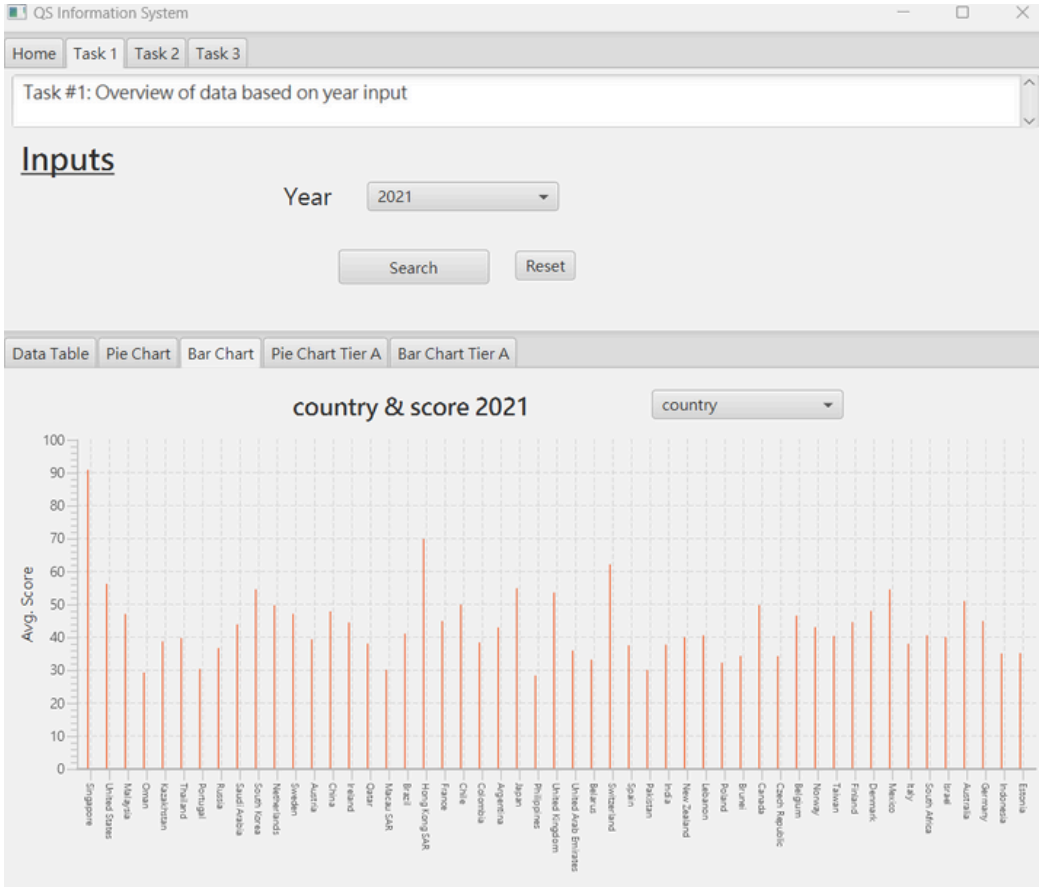
Piechart : search by type



Barchart : search by type

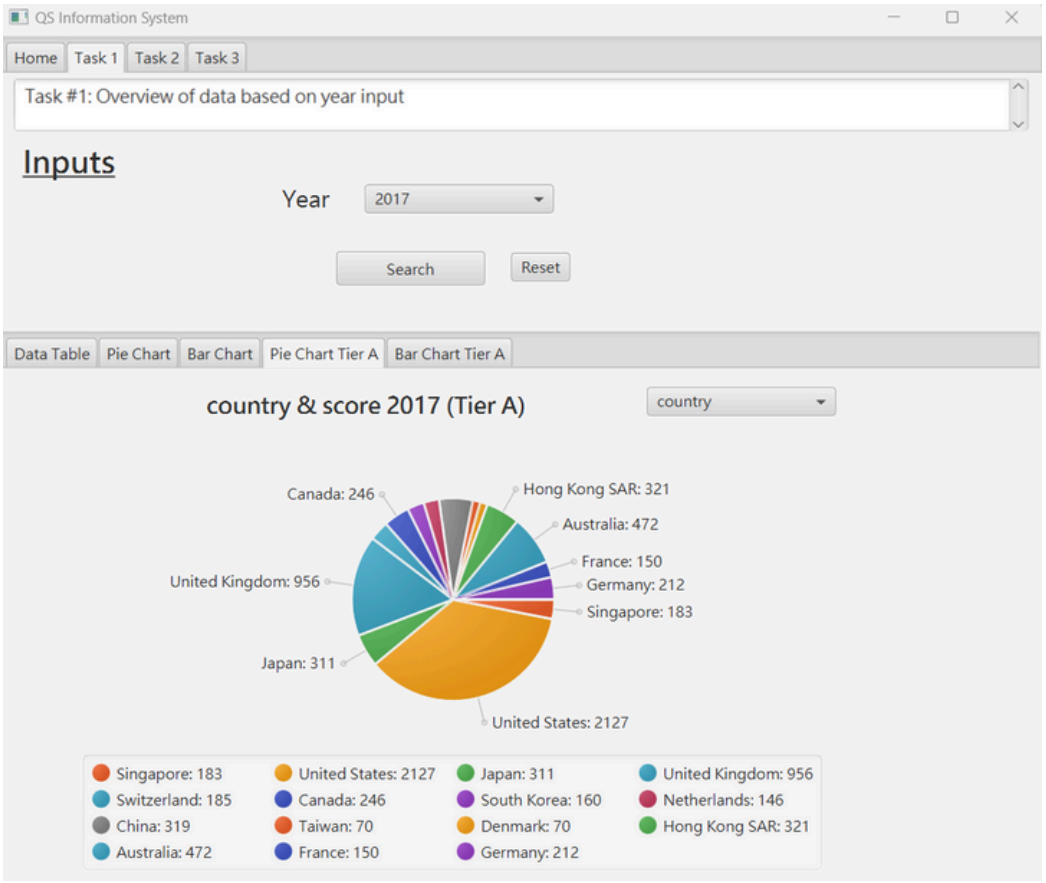


Barchart : search by country



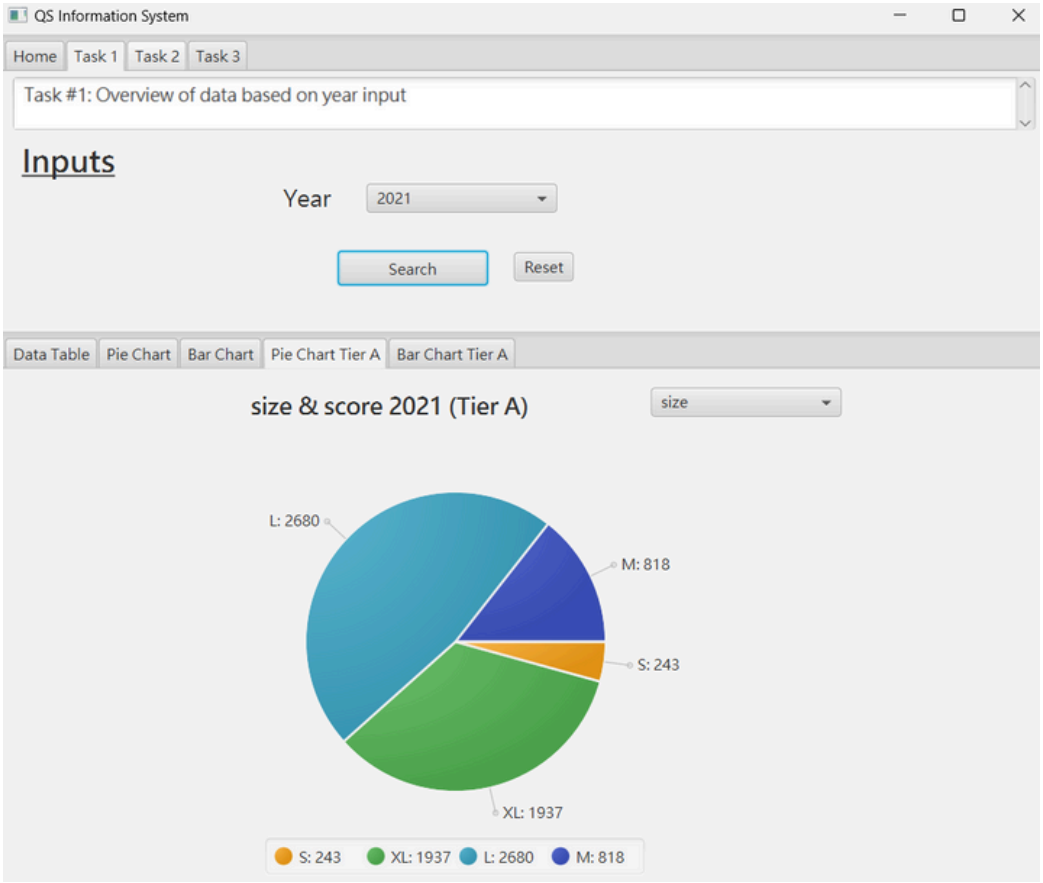
New tabs:

Piechart for tier A : search by country



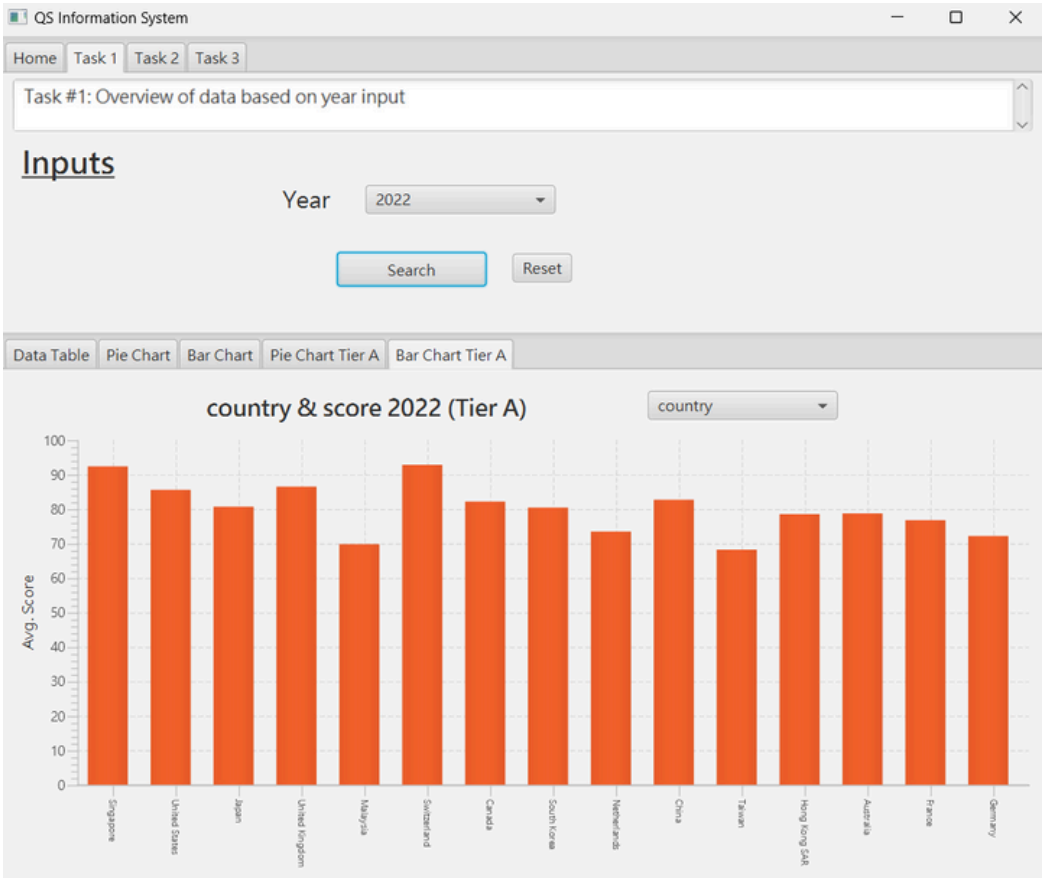
New tabs:

Piechart for tier A : search by size



New tabs:

Barchart for tier A : search by country



Documentation on the implemented tasks

Controller:

```
246     /**
247      * Clears the data and resets the page for Task 1.
248      * This method clears the data table, pie chart, and bar chart.
249      * It also resets the choice boxes and labels to their default values.
250      * Finally, it sets the current year to \"0\".
251      */
        2 usages  lung
252     @FXML
253     private void T1_onClickClear() {
254         /*
255             Your Code Here.
256             Reset the Page Task1. (including the choice box, labels and charts)
257          */
258
259         t1DataTable.getItems().clear();
260         t1PieChart.getData().clear();
```

```
285
286     /**
287      * Handles the search action for Task 1.
288      * When the search button is clicked:
289      * 1. Fetches the selected year from the choice box.
290      * 2. Clears previous data in the data table, pie chart, and bar chart.
291      * 3. Initializes an analyzer (presumably `T1Analysis`).
292      * 4. Updates the table view with information about universities.
293      * 5. Updates the pie chart to display the sum score of the selected property (based on `t1PieChartChoiceBox`).
294      * 6. Updates the bar chart to show the average score of the selected property (based on `t1BarChartChoiceBox`)
295      */
        4 usages  lung
296     @FXML
297     private void T1_onClickSearch() {
298         /*
299             Your Code Here.
300             When click search on Task1:
301                 1. Fetch the year from the choice box.
```

```
351
352     /**
353      * Custom TableCell that displays a hyperlink.
354      * When clicked, it opens the specified URL using the HostServices provided.
355      *
356      * @param <S> The type of the TableView row item.
357      * @param <T> The type of the cell value (usually String or URL).
358      */
359
        1 usage  lung
360     class HyperlinkTableCell<S, T> extends TableCell<S, T> {
            4 usages
361         private final Hyperlink hyperlink;
            1 usage
```

```
385     /**
386      * Custom TableCell that displays a hyperlink with the text "logo".
387      * When clicked, it opens the specified URL using the HostServices provided.
388      *
389      * @param <S> The type of the TableView row item.
390      * @param <T> The type of the cell value (usually String or URL).
391      */
392
        1 usage  lung
393     class LogoHyperlinkTableCell<S, T> extends TableCell<S, T> {
            4 usages
394         private final Hyperlink hyperlink;
            1 usage
```

```java
/**
 * Handles the selection event for the pie chart choice box.
 * Updates the pie chart data based on the selected choice.
 *
 * @param event The ActionEvent triggered by the choice box selection.
 */
1 usage  lung
private void handlePieChartChoiceBoxSelection(ActionEvent event) {
    String selectedChoice = t1PieChartChoiceBox.getValue();
```

```java
/**
 * Handles the selection event for the bar chart choice box.
 * Updates the bar chart data based on the selected choice.
 *
 * @param event The ActionEvent triggered by the choice box selection.
 */

1 usage  lung
private void handleBarChartChoiceBoxSelection(ActionEvent event) {
    String selectedChoice = t1BarChartChoiceBox.getValue();
    // Update data based on the selected choice (fetch from DB, modify existing data, etc.)
```

QSItem:

```java
/**
 * Sets the grade for a QSItem based on the provided mean and standard deviation.
 * The grade is determined by comparing the difference between the item's score
 * and the mean score with multiples of the standard deviation.
 *
 * @param mean The mean score for the dataset.
 * @param sd   The standard deviation of scores for the dataset.
 *
 *    grading system:
 *          >2sd -> A+
 *          >1.6sd -> A
 *          >1sd -> A-
 *          >0.6sd -> B+
 *          >0sd -> B
 *          >-0.6sd -> B-
 *          >-1sd -> C+
 *          >-1.6sd -> C
 *          >-2sd -> C-
 *          below 2sd -> D
 */
1 usage  lung
public void setGrade(double mean, double sd) {
    double diff = getDoubleScore()-mean;
    if(getDoubleScore()>0){
        if (diff>=2*sd){
```

QSList:

```java
/**
 * Initializes the data by loading information from a CSV file.
 * This method performs the following steps:
 * 1. Loads data from the specified CSV file into a list.
 * 2. Collects unique values for university, type, region, and country.
 *
 * @throws RuntimeException if any I/O or validation exception occurs during CSV processing.
 */

1 usage  lung
public static void initialize() {
    /*
        Your Code Here.
        1. Load the csv into list.
```

T1Analysis:

```java
/**
 * Represents an analysis of QSItem data for a specific year.
 * The analysis includes calculating statistics such as mean and standard deviation
 * for the scores, as well as categorizing universities based on their grades.
 *
 * @param year The year for which the analysis is performed.
 */
1 usage  lung
T1Analysis (String year) {
    /*
        Your Code Here.
        Collect the QSItem with corresponding years into tableList.
```

```java
/**
 * Retrieves PieChart data based on the specified search criteria.
 * The PieChart displays the sum of scores for universities grouped by the chosen category.
 *
 * @param searchName The category for which to retrieve data ("country", "region", or "size").
 * @return An ObservableList of PieChart.Data representing the summed scores for each category.
 */

2 usages  lung
ObservableList<PieChart.Data> getPieChartData(String searchName) {
    ObservableList<PieChart.Data> pieChartData= FXCollections.observableArrayList();
    /*
        Your Code Here.
        Return the Pie Chart Data.
        Pie Chart shows the SUM of the score.
```

```java
/**
 *
 * This function has same structure as above function.
 * But it gets data only for Tier A University.
 *
 * Retrieves PieChart data based on the specified search criteria.
 * The PieChart displays the sum of scores for universities grouped by the chosen category.
 *
 * @param searchName The category for which to retrieve data ("country", "region", or "size").
 * @return An ObservableList of PieChart.Data representing the summed scores for each category.
 */
2 usages  lung
ObservableList<PieChart.Data> getPieChartDataRankA(String searchName) {
    ObservableList<PieChart.Data> pieChartData= FXCollections.observableArrayList();
```

```java
/**
 * Retrieves bar chart data based on the specified search criteria.
 * The bar chart displays the average score for universities grouped by the chosen category.
 *
 * @param searchName The category for which to retrieve data ("country", "region", or "size").
 * @return An ObservableList of XYChart.Series representing the average scores for each category.
 */
2 usages  lung
ObservableList<XYChart.Series<String, Double>> getBarChartData(String searchName) {
    ObservableList<XYChart.Series<String, Double>> seriesList = FXCollections.observableArrayList();
```

```java
/**
 *
 * This function has same structure as above function.
 * But it gets data only for Tier A University.
 *
 * Retrieves bar chart data for a given search criteria.
 *
 * @param searchName The search criteria (e.g., "country", "region", or "size").
 * @return An ObservableList of XYChart.Series containing the average scores based on the search criteria.
 */
2 usages  lung
ObservableList<XYChart.Series<String, Double>> getBarChartDataRankA(String searchName) {
    ObservableList<XYChart.Series<String, Double>> seriesList = FXCollections.observableArrayList();
    XYChart.Series<String, Double> barData= new XYChart.Series<>();
    /*
```