# Declaration of Academic Integrity

Academic Integrity is a central tenet of Singapore Polytechnic. The polytechnic rules state that "Cheating in examinations and other assessed work is a very serious offence. This includes copying and using plagiarised material. Any student who cheats, attempts to cheat or breaches any rules for examinations and tests will face disciplinary action. The student is liable to be expelled."

Check **only one** of the two options below:

I affirm that the work I submit is my own, produced without help from any AI tool(s) and/or other source(s).

I affirm that the work I submit has been produced with the use of AI tool(s) and/or other source(s) which I have acknowledged fully in the [following section](#_Acknowledgement_of_use).

By signing this form, I declare that the above affirmation made is true, and that I have read and understood the rules stated in Students Handbook on “[Plagiarism](https://www.sp.edu.sg/sp/student-services/osc-overview/student-handbook/intellectual-property-copyright-and-plagiarism)” and “[Breach of Examination/Assessment Rules](https://www.sp.edu.sg/sp/student-services/osc-overview/student-handbook/conduct-in-examinations-breach-of-exam-rules)”.

|  |  |  |
| --- | --- | --- |
| Name:  M.Rajkaran S/O V.Muralikaran | Student ID:  2109039 | Class:  DIT/FT/2A/02 |
| Signature: | Date:  6/8/2023 | |

## Acknowledgement of use of AI tool(s) and/or other source(s) (where applicable)

### AI tool(s)

|  |  |
| --- | --- |
| **Name of AI tool** | Chatgpt |
| **Input prompt** | How do I create a new dataframe with these values |
| **Date generated** | 1.8.2023 |
| **Output generated** | smu\_filtered = graduate\_employment\_survey[graduate\_employment\_survey.university ==                                            'Singapore Management University']  smu\_filtered\_it = smu\_filtered[smu\_filtered.school.str.contains(      'Information Systems')]  # Create an empty list to store the DataFrames for each year  dfs = []  for year in smu\_filtered\_it.year.unique():      year\_rows = smu\_filtered\_it[smu\_filtered\_it['year'] == year].copy()      year\_rows['basic\_monthly\_mean'] = pd.to\_numeric(          year\_rows['basic\_monthly\_mean'])      year\_rows['employment\_rate\_overall'] = pd.to\_numeric(          year\_rows['employment\_rate\_overall'])      mean\_basic\_monthly\_mean = year\_rows['basic\_monthly\_mean'].mean()      mean\_employment\_rate = year\_rows['employment\_rate\_overall'].mean()      school = year\_rows.iloc[0]['school']      degree = year\_rows.iloc[0]['degree']      university = year\_rows.iloc[0]['university']      df\_year = pd.DataFrame({          'year': year,          'university': university,          'school': school,          'degree': degree,          'mean\_basic\_monthly\_mean': mean\_basic\_monthly\_mean,          'mean\_employment\_rate': mean\_employment\_rate      }, index=[0])      dfs.append(df\_year)  result\_df = pd.concat(dfs, ignore\_index=True) |
| **Impact on submission** | Made my graphs look better as the data was more accurate |

|  |  |
| --- | --- |
| **Name of AI tool** | ChatGPT |
| **Input prompt** | How do I label the values in bar chart |
| **Date generated** | 1.8.2023 |
| **Output generated** | for bar in bars:  width = bar.get\_width()  plt.text(width, bar.get\_y() + bar.get\_height() / 2, f'{width:.1f}', ha='left', va='center', fontsize=12) |
| **Impact on submission** | Helped me label values |