**Course Learning Outcomes:**

Upon completion of this assignment, you should be able to:

|  |  |
| --- | --- |
| CLO1 | Explain the fundamental of software development and programming concepts (C2, PLO1). |
| **CLO2** | **Construct a programmable solution using appropriate problem solving methods and programming concepts to given scenario (C3, PLO2).** |

1. **ASSIGNMENT DESCRIPTION**

**Trip Planner Application**

You are required to develop a KL Trip Planner Application to facilitate communication and transactions between travellers and service providers. Users can use the system to research, book, and manage their travel arrangements before the trip. The system should streamlines the process of planning, booking, and managing trips, catering to the diverse needs and preferences of users involved in the travel ecosystem.

The following characteristics are important to be included in your system:

1. **System administrator** manages and maintain the system. Admin features include:

* Login and Logout from the system
* User management
  + Block Traveller from using the system
  + Add , block Merchant account
* Update the current promotion and provide Trip recommendation.

1. **Service providers** are merchant or vendor offering travel-related products or services, such as hotels, tour operators, restaurants, and attractions. Merchants features include:

* Login and Logout from the system
* Manage product and services
  + Add, delete and update information such as quantity, schedule and pricing
* View, cancel and confirm Booking

1. **Guest** is a visitor or potential user exploring the trip planning system without registering for an account. Guest features include:

* Browsing and search functionalities to explore destinations, attractions, and travel services.
* View recommended itineraries and checking availability.
* Sign up for an account.

1. **Traveller** is an individual or group planning a trip, seeking information, making reservations, and managing bookings. Travellers features include:

* Login and Logout from the system
* Update own profile
* Plan a trip
  + View, search, book and cancel products and services.

1. **REQUIREMENTS**
2. This is a group assignment, which comprises of **FOUR (4)** members.
3. You are required to carry out extra research for your system and document for any logical assumptions you made after the research.
4. You are required to use **Python** programming language to implement the solution. Usage of any other language like C/C++/Java is not allowed.
5. The submitted application should run without any errors. You should;
   * + Apply **good programming practice** such as comments, variable naming conventions, indentation,and symbolic constants where it is appropriate.
     + Include **validations** to ensure the accuracy.
     + **Global variables** and **build in functions** like min, max, sort, etc… are **not allowed**.
     + Embrace modular programming technique and should be menu driven with the use of **functions**.
     + **Store all data** either in lists, dictionary, text, json or csv files. There is no limit on the number of files that can be used but they should be kept minimum.
6. You may include any extra features which you may feel relevant and that add value to the system.
7. There should be no need for graphics (user interface) in your program, as what is being assessed, is your problem solving and programming skill not the interface design.
8. **IMPORTANT INSTRUCTIONS:**

This is a group assignment. Upon submission of your assignment, you are required to present your assignment at a date and time specified by your module lecturer.

In a situation where a student / group:

* ***Failed to attempt the assignment demonstration****, a* ***maximum of 50 marks*** *awarded for the assignment.*
* *Found to be* ***involved in plagiarism****, the offence will be dealt in accordance to APU regulations on plagiarism.*
* ***Failed to submit the report and program before the given deadline****,* ***zero marks*** *will be awarded for the assignment.*
* *Submitted* ***wrong files (old / irrelevant files)****, no attempts will be given to resubmit the files. Hence, please check the files before submission.*

Each member is required to contribute towards some features in the system and documentation, present and explain his or her work accordingly. Each member should be able to answer questions during presentation with regards to the overall systems project and or specific question(s) related to the codes used in the development. Responses to these questions such as “I don’t know because I didn't work on that part of the assignment” are not acceptable and will result in a penalty for either the entire group or specific individual(s)

1. **DELIVERABLES**
2. Submit a softcopy of the report (*in Word Document format*), along with programs files (**python and data files)** via Moodle. The organization of files and folders **must adhere to the following instructions precisely**:
   * The report should be named using format
     + *“<GroupNo>\_<student ID-leader>\_<student ID-member1>\_<student ID-member2>\_<student ID-member3>\_<student ID-member3>.docx”.*
   * For example **“G1\_TP012345\_TP012344\_ TP012123.docx”**
   * All the source codes **(.py)** and **data files** should be zipped into one file and named following the above format. Make sure to **DELETE** all non-source-code files, including executables (\*.exe).
3. A documentation of the system - that incorporates basic documentation standards such as header and footer, page numbering and includes:

* Cover page
* Table of contents
* Introduction and assumptions
* Design of the program – using pseudocode **OR** flowcharts – which adheres to the requirements provided above
* Program source code and explanation
* Screenshots of sample input/output and explanation
* Conclusion
* References using APA Name Referencing
* Workload Matrix

1. **ASSESSMENT CRITERIA**
2. Design (Pseudocode or Flowchart) 30%

Detailed, logical and accurate design of programmable solution.

1. Coding / Implementation (Python code) 40%

Application of Python programming techniques (from basic to advance); good programming practices in implementing the solution as per design; and adequate validation meeting all system requirements with all possible additional features.

1. Documentation 20%

Adherence to document standard format and structure; screen captures of input/output with explanation; and inclusion of generated text files.

1. Demonstration 10%

Ability to run, trace code, explain work done and answer questions.

1. PERFORMANCE CRITERIA

Distinction (75% and above)

This grade will be assigned to work which meets all of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to advanced level. The program solution is unique with excellent coding styles and validation. The program implemented maps completely against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with hardly any errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented have clear explanation. Student must be able to provide excellent explanation of the codes and work done, show additional concepts / new ideas used in the solution, able to answer all questions posed with accurate / logical answers / explanation provided with sound arguments and clear discussion. Overall an excellent piece of work submitted.

Credit (65%-74%)

This grade will be assigned to work which is considered to be of good standard and meets most of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to at least intermediate level. The program solution is unique with good coding styles and validation. The program implemented maps well against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with minor errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented with some explanation. Student must be able to provide good explanation of the codes and work done, answer most questions posed with mostly accurate / logical answers / explanation. Overall a good assignment submitted.

Pass (50%-64%)

This grade will be assigned to work which meets at least half of the basic requirements (approximately 50%) stated in the questions. The program runs smoothly when executed. There is clear evidence and application of Python concepts at basic level. The program solution is common with basic coding styles and validation. The program implemented somewhat maps with the design (pseudocode and flowchart) as seen in the documentation. The design of the solution is average in terms of logic and style with some errors / omissions. The documentation has some missing components. Sample inputs/outputs documented but without any explanation. Student must be able to explain some codes and work done and able to answer some questions posed with some accurate / logical answers / explanation. Overall an average piece of work submitted.

Fail (Below 50%)

This grade will be assigned to work which achieved less than half of the requirements stated in the question. The program is able to compile but not able to execute or with major errors. The program solution has only basic coding styles with no validation. The program solution has little or no mapping with the design. The design of the solution has major / obvious errors / omissions. The documentation has some missing essential components. Student is barely able to explain the codes / work done and answer given on the questions posed but with mostly inaccurate / illogical answers / explanation. Overall a poor piece of work submitted.