PROJECT PLAN.

1.0 INTRODUCTION

Expert systems are computer programs designed to simulate human experts' decision-making processes. These systems are based on the knowledge and expertise of human experts in a particular domain, and they use this knowledge to solve problems in that domain.

Expert systems have been used for various tasks, including medical diagnosis, fault diagnosis, interpreting measurement data, and configuring systems. The main reasons for using expert systems are to improve human experts' productivity, apply expertise uniformly and impartially, and preserve experts' knowledge.

An expert system has three main components: a knowledge base, an inference engine, and a user interface. The knowledge base contains domain-specific knowledge that has been gathered from human experts. The inference engine is responsible for using this knowledge to solve problems. The user interface is used to communicate with the expert system and to provide input and output.

There are different approaches to building expert systems, but the most common method is the rule-based expert system methodology, known as the production system. This approach uses a set of rules to determine how the expert system should behave in different situations.

Reasoning types such as case-based and rule-based reasoning can also be used in expert systems. Case-based reasoning is a process of solving new problems by finding similar cases in the knowledge base and using the solutions of those cases to solve the latest issue. Rule-based reasoning uses a set of rules to infer new knowledge from the knowledge base.

Expert systems can improve human experts' productivity and provide uniform and impartial expertise. They can also be used to preserve experts' knowledge and to save time.

1.1 Project Scope

The project is developed using Python and its various modules. The project's main aim is to create a system that can help students improve their performance by providing them with a detailed analysis of their performance.

The project consists of three main modules – Student Information Form, Verbal, and Quantitative. In the Student Information Form, the details of the students are considered. In the Verbal and Quantitative modules, the student's performance is analyzed by providing them with questions and grades based on their answers to particular questions.

The Student Information Form module helps keep track of the student's details. It consists of various fields such as name, roll number, age, gender, etc. This module allows the storing of the students' details and provides a detailed analysis of their performance.

The Verbal module consists of a set of questions to be answered by the students. The answers provided by the students are analyzed, and grades are given based on their performance. This module helps in assessing the verbal ability of the students.

The Quantitative module also consists of a set of questions to be answered by the students. The answers provided by the students are analyzed, and grades are given based on their performance. This module helps in assessing the quantitative ability of the students.

The project is developed using the Python programming language. Python is a powerful programming language that helps create various applications. Python is easy to learn and use. It is a widely used language in the software industry.

The project is developed using the Tkinter module. Tkinter is the standard GUI library for Python. When combined with Tkinter, Python provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

1.2 Major software functions

The Expert System for Swear Analysis is a software application designed to help users analyze and understand the use of swear words in a given text. The software is developed using the Python programming language and the Tkinter GUI library. The application consists of three main modules: the Student Information Form, the Verbal module, and the Quantitative module.

The Student Information Form collects details about the students using the Expert System for Swear Analysis. This module allows the user to input the name, age, and gender of the students. The user can also specify the level of profanity the students are comfortable with.

The Verbal module analyzes the use of swear words in a given text. This module provides the user with a list of swear words and their definitions. The user can select the words they want to analyze, and the software will report on the usage of those words in the text.

The Quantitative module analyzes the use of swear words in a given text. This module provides the user with a list of swear words and their definitions. The user can select the words they want to analyze, and the software will report on the usage of those words in the text. The information will include the number of times each word was used, the percentage of the text made up of swear words, and the average number of swear words per sentence.

1.3 Performance/Behavior issues

The above project is an Expert system for swear analysis developed using Tkinter. This system analyzes students' performance in verbal and quantitative tasks. This system will provide grades to the students based on their answers to particular questions. This system is easy to use and is very efficient in analyzing students' performance.

1.4 Management and technical constraints

The Expert system development for swear analysis will be conducted in a manner that is efficient and effective given the limited resources. The approach to action will be to create a chatbot or Tkinter interface that can collect data from students and provide feedback on their performance. The chatbot or Tkinter interface will be designed to be easy to use and understand for the students. The development process will be conducted in a way that is efficient and effective given the limited resources.

2.0 Risk Management

There are a few risks associated with this project:

1. The student may not provide accurate information.

2. The student may not perform well in the verbal or quantitative sections.

3. The student may not be able to use the chatbot or Tkinter interface.

4. The fourth risk is that the student may not be able to understand the output of the chatbot or Tkinter interface.

5. The chatbot or Tkinter interface may not be able to handle the input of the student.

The approach to managing these risks is first identifying them and then creating a plan to mitigate them.

2.1 Project Risks

1. The student may not provide accurate information. This can be mitigated by having the student fill out the form in front of a teacher or administrator who can verify the information.

2. The student may not perform well in the verbal or quantitative sections. This can be mitigated by providing practice questions and grades based on their answers.

3. The student may not be able to use the chatbot or Tkinter interface. This can be mitigated by providing instructions on how to use the chatbot or Tkinter interface.

4. The fourth risk is that the student may not be able to understand the output of the chatbot or Tkinter interface. This can be mitigated by describing the production.

5. The chatbot or Tkinter interface may not be able to handle the input of the student. This can be mitigated by providing a backup plan in case the chatbot or Tkinter interface does not work.

2.2 Risk Table

Name of risk	**probability**	**impact**	**RM3 pointer**
Inaccurate information	Low	High	Management
Verbal and Quantitative sections	Low	Low	Monitoring
chatbot or Tkinter interface	High	High	Management
The output of the chatbot or Tkinter interface	Low	Low	Mitigation

2.3 Overview of Risk Mitigation, Monitoring, and Management (RM3)

The first step in risk mitigation is identifying the risks that could potentially impact the project. Once the risks have been identified, they need to be analyzed to determine the likelihood of occurrence and the potential impact on the project if they do occur. After the risks have been studied, mitigation strategies can be developed to reduce the likelihood of occurrence or reduce the project's impact if they do occur.

The second step in risk mitigation is monitoring the risks throughout the project. This involves tracking the status of the risks and monitoring for any changes that could impact the project. If a change does occur, the mitigation strategy developed for that risk will need to be implemented.

The third step in risk mitigation is the management of the risks. This involves making decisions on how to deal with the risks that have been identified. The changes identified as having the potential to impact the project must be given the most attention. Management of risks also includes developing contingency plans in case a threat does occur.

Risk mitigation is an important part of project management. By taking the time to identify, analyze, and develop mitigation strategies for risks, the project manager can increase the chances of the project being successful.

3.0 Project Schedule

3.1 Project task set

The process model chosen for this project is the waterfall model. This model is well suited for a project of this nature as it is simple and easy to understand. The main disadvantage of this model is that it does not allow for much flexibility, which can be a problem if the requirements of the project change during development.

The main activities that need to be completed to develop this system are:

1. requirements gathering

2. system design

3. implementation

4. testing

5. deployment

The task set for this project is as follows:

1. Develop a requirements specification document that outlines the functional and non-functional requirements of the system.

2. Design the system architecture and database.

3. Implement the system using a suitable programming language and database.

4. Test the system to ensure that it meets the requirements.

5. Deploy the system on a suitable web server.

3.2 Functional decomposition

The project can be broken down into four main sections:

1. Student Information Form

2. Verbal

3. Quantitative

4. Schedule

The Student Information Form section can be further broken down into:

- Entering Student Details
- Validating Student Details
- Storing Student Details

The Verbal section can be further broken down into:

- Asking Questions
- Checking Answers
- Giving Scores

The Quantitative section can be further broken down into:

- Asking Questions
- Checking Answers
- Giving Scores

The Schedule section can be further broken down into:

- Checking Student Availability
- Making Appointments
- Sending Appointment Reminders here.

3.3 Task network

1. creating a GUI application using the Tkinter module.

2. add one or more of the widgets mentioned earlier to the GUI application.

3. expose the geometry manager classes: pack, grid, and place.

4. Organize the widgets in the parent widget area.

3.4 Timeline chart

1. Week 1: Research and develop the structure of the chatbot. This will include deciding on the programming language to use, designing the conversation flow, and researching any external libraries or services that will be needed.

2. Week 2: Write the code for the chatbot. This will include setting up the basic structure of the chatbot and coding the logic for the conversation flow.

3. Week 3: Test the chatbot. This will involve trying out the chatbot with a group of people to see how it works in practice and to identify any areas that need improvement.

4. Week 4: Improve the chatbot. This will involve making changes to the chatbot based on the feedback from the testing phase.

5. Week 5: Deploy the chatbot. This will involve making the chatbot available to the public through a website or a chat platform such as Facebook Messenger.

6. Week 6: Monitor the chatbot. This will involve tracking how the chatbot is being used and making changes if necessary.

3.5 Schedule compliance

Project managers use the project schedule to monitor progress and identify potential problems or risks. They also communicate project status to project sponsors, stakeholders, and other interested parties.

Project schedules can be created using various methods, including Gantt charts, project management software, and spreadsheets.

The most important part of the project schedule is the activity list, which details all the tasks that need to be completed to finish the project. The activity list should be broken down into smaller, manageable tasks that can be assigned to individual team members.

Each task in the activity list should have a start date, end date, and assigned resources. The start and end dates track progress and identify potential delays. Trusted resources can be used to track costs and estimate the amount of time that will be required to complete the task.

Once the project schedule is complete, project managers must monitor and update it regularly. They will also need to create reports communicating the project status to stakeholders.

Project management software can automate project schedules creation, monitoring, and reporting. This software can also be used to generate Gantt charts, which are a type of graphical representation of the project schedule.

Spreadsheets can also be used to create project schedules. However, they are generally not as effective as project management software for tracking progress and communicating project status.