Develop an application that suggests best solution to a question based on past behavior & available solutions.

ABC Corp is an IT support and services company. This company provides L1 and L2 technical support across various IT functions. ABC Corp was incorporated in 2004 and currently is supporting over 50 companies for their IT services.

Problem:
Currently the company is facing a problem in handling their ticket volume due to increased user base. Resolution time got longer and not as fast as it used to be. They are losing customers.

Solution:
After doing a root cause analysis, the company found out that their L1 Technical support agents are spending too much time on a ticket which is causing a delay's in the closure of a ticket.
They found out

1. The agents are not able to find the correct resolution which is resulting in taking more time to respond. Albeit, resolution is not very accurate.

2. Current ticket resolution process is a standard one without much automation, where each agent must manually search for a ticket's resolution.

System to Build:
After some deliberations, the company unanimously decided to build a smart system to help their support agents and achieve a higher customer experience index

1. Develop a system where, when a user types in an IT problem, the system should be able to give top (suggestive) resolutions.

2. The system should learn from the old tickets and user manuals which usually the agents/users would refer to derive and suggest a solution.

3. Once the resolutions have been provided and if the user selects one from the suggestions provided, the system should be able to learn and keep that as the correct solution for similar future searches.

4. The UI could be a simple search / chat interface where an agent / user can type in the query, and the suggestions are shown (minimum UX)

Expected Outcome

- Reduce the agent's search time and close the ticket faster.

- New system to eliminate manually searching of past tickets and resolutions.

- New system to learn from past tickets and resolutions.

Example Scenario 1:

1. Technical support agent get a new ticket asking to reset a user's password.

2. Now the agent instead of manually searching through the old tickets and user manuals, will use this new search system (the system built to learn old tickets with resolutions related to the same problem).

3. The newly built system will provide suggestions on the resolution of the ticket.

Example Scenario 2:

1. A ChatBot available to end user (support seeker) who is raising a ticket. This BOT will replace L1 technical support agent (support provider).

2. Now the BOT will respond instantly with suggestive resolutions that it learnt from old tickets, resolutions, user manuals related to the same problem.

Sample Data:

1. Old IT ticket information (SampleInput.xls)

2. User manuals (SampleDocuments.zip)

3. Output Data which is expected. Here we have sample input questions which an agent would type and the sample expected correct output which the system should show are listed. These should be demo'ed when submitting a solution. (SampleOutput.xls)

Deliverables:

1. Submit the source code. A link to your code on GitHub (this repository may be public or private - if private, access must be given in the testing instructions provided). Code should be compilable and runnable. - Mandatory

2. A presentation in the attached format which include a brief explanation of what your system does and what makes it unique. A way to access your application for judging and testing (Download SolutionPresentation.pptx) - Mandatory

Judging Criteria:
This solution will be evaluated on the following parameters:

1. Accuracy of the results.

2. Number of output scenarios being matched.

3. Creative thinking and design of the system: creativity and originality of the application idea, usability, design and intuitiveness.

4. Code quality.

5. Architecture.

6. Completeness.

7. Presentation.

8. Going beyond the ask and thinking out of the box.