**MOVIE BOT**

1. **INTRODUCTION**
   1. **Overview**

In today’s world no one is having much time to do a particular work, so everyone wants everything available at their fingertips.

Here I prepared a solution to online movie ticketing system known as Movie Bot and will be able to deal with movie suggestions, offers enquiry and ticket book by a bot.

* 1. **Purpose**

The main purpose of this movie bot is to save time with improved user interactions and help users to book movie tickets with ease.

1. **LITERATURE SURVEY**
   1. **Existing Problem**

All the movie bots which are available in market helps only with either suggesting movies or giving options to watch movies with your friends. But not found any bot which helps with direct movie ticket booking. The actual problem is time saving while booking the ticket which is not there in the existing model.

* 1. **Proposed Solution**

Here I am with a solution which can help users to book a ticket by just selecting few options. A user will be able to get suggestion of movies according to genre, city and language. The bot also provides information about the offers. By just clicking on a movie a user will be able to view the price of a movie ticket and will be able to book ticket and pay using his mail id.

1. **THEORITICAL ANALYSIS**
   1. **Block Diagram**

Block diagram mainly consist of Users, Watson Assistant and the Backend system.

Backend System

Watson Assistant

Applications

* 1. **Hardware/Software Designing**

**Software Design:**

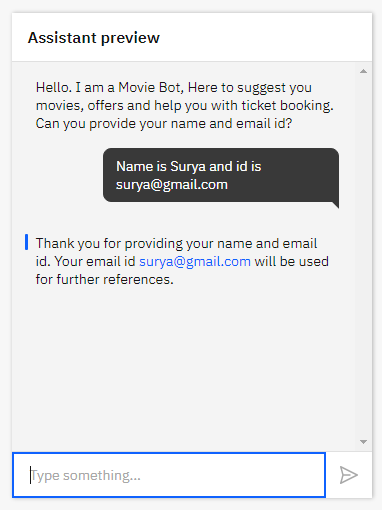
Software designing is done using Machine Learning algorithms and Watson assistant.

We need to create intents, entities and dialog flows. We need to train the model based on intents and entities. After creation of intent and entity we need to set the flow in the dialog.

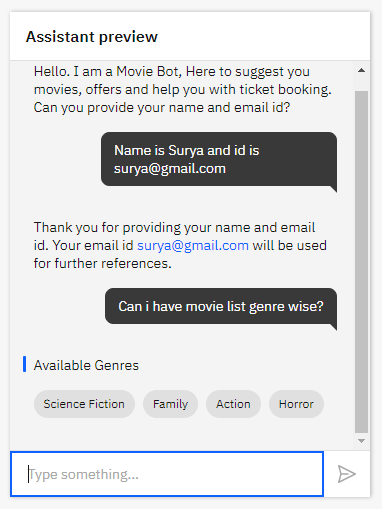
1. **EXPERIMENTAL INVESTIGATION**

Experimental investigation is done using Watson assistant.

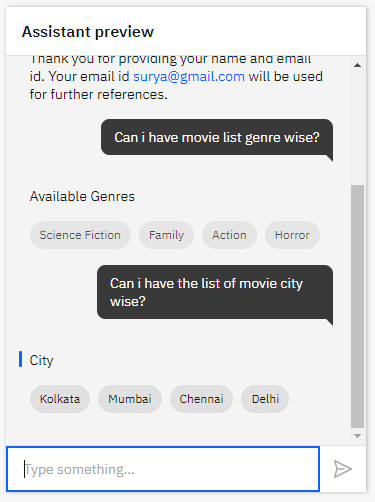
Bot Preview:



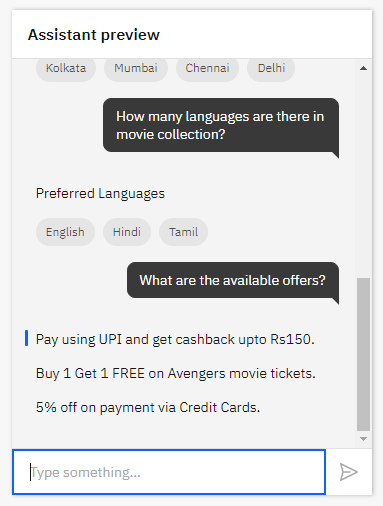
Movie Suggestion (Genre):



Movie Suggestion (City):

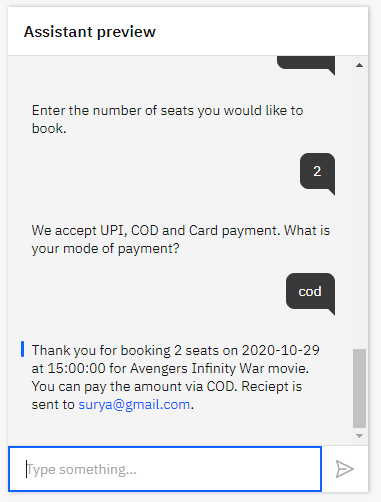


Offers:



1. **RESULT**

A working movie bot which can be able to book tickets, suggest movies, display offers and many more.



1. **ADVANTAGES AND DISADVANTAGES**
   1. **Advantages**

The main advantage of a movie bot is saving time. A user doesn’t need to go open a website and follow all the procedures to book a ticket. Now, using this bot user will be able to get any information related to movie on fingertips.

* 1. **Disadvantages**

The only disadvantage is getting dependent on technology too much.

1. **CONCLUSION**

The proposed movie bot helps users to book tickets, which can be done using simple options. It also helps user to get the list of movies according to different genres, cities and languages. Using this user friendly bot a lot of time can be saved in this busy world.

1. **BIBLIOGRAPHY**
2. <https://eu-gb.assistant.watson.cloud.ibm.com/eu-gb/crn:v1:bluemix:public:conversation:eu-gb:a~2F0b69eb8821144886953088a547d6fdd6:089072fa-ca07-4571-b2f8-75590d637dcb::/home>
3. <https://smartinternz.com/gurucool-project/88>
4. <https://www.youtube.com/watch?v=tUBJZfnxeTw>

**APPENDIX**

1. **Source Code**
2. {
3. "intents": [
4. {
5. "intent": "Book\_Tickets",
6. "examples": [
7. {
8. "text": "I would like to book a ticket"
9. }
10. ],
11. "description": ""
12. },
13. {
14. "intent": "Enquiry",
15. "examples": [
16. {
17. "text": "Any offers?"
18. },
19. {
20. "text": "Help me with the movie genres."
21. },
22. {
23. "text": "What are the available movies?"
24. },
25. {
26. "text": "What are the offers you are providing?"
27. }
28. ],
29. "description": ""
30. },
31. {
32. "intent": "Greetings",
33. "examples": [
34. {
35. "text": "Good Afternoon"
36. },
37. {
38. "text": "Good Evening"
39. },
40. {
41. "text": "Good Morning"
42. },
43. {
44. "text": "Hi"
45. }
46. ],
47. "description": ""
48. },
49. {
50. "intent": "Movie\_Availability",
51. "examples": [
52. {
53. "text": "Can I have city wise movie list?"
54. },
55. {
56. "text": "Can I have the list of movie genre wise?"
57. },
58. {
59. "text": "Can I have the list of movies according to language?"
60. }
61. ],
62. "description": ""
63. }
64. ],
65. "entities": [
66. {
67. "entity": "available",
68. "values": [
69. {
70. "type": "synonyms",
71. "value": "Aquaman",
72. "synonyms": []
73. },
74. {
75. "type": "synonyms",
76. "value": "Bala",
77. "synonyms": []
78. },
79. {
80. "type": "synonyms",
81. "value": "Bigil",
82. "synonyms": []
83. },
84. {
85. "type": "synonyms",
86. "value": "Chhichhore",
87. "synonyms": []
88. },
89. {
90. "type": "synonyms",
91. "value": "Darbar",
92. "synonyms": []
93. },
94. {
95. "type": "synonyms",
96. "value": "Joker",
97. "synonyms": []
98. }
99. ],
100. "fuzzy\_match": true
101. },
102. {
103. "entity": "book\_ticket",
104. "values": [
105. {
106. "type": "synonyms",
107. "value": "book",
108. "synonyms": [
109. "bookings",
110. "buy",
111. "purchase"
112. ]
113. }
114. ],
115. "fuzzy\_match": true
116. },
117. {
118. "entity": "city",
119. "values": [
120. {
121. "type": "synonyms",
122. "value": "Chennai",
123. "synonyms": []
124. },
125. {
126. "type": "synonyms",
127. "value": "Delhi",
128. "synonyms": []
129. },
130. {
131. "type": "synonyms",
132. "value": "Kolkata",
133. "synonyms": []
134. },
135. {
136. "type": "synonyms",
137. "value": "Mumbai",
138. "synonyms": []
139. }
140. ],
141. "fuzzy\_match": true
142. },
143. {
144. "entity": "email",
145. "values": [
146. {
147. "type": "patterns",
148. "value": "email",
149. "patterns": [
150. "[a-z0-9.\_%+-]+@[a-z0-9.-]+\\.[a-z]{2,}$"
151. ]
152. }
153. ],
154. "fuzzy\_match": true
155. },
156. {
157. "entity": "enquiry",
158. "values": [
159. {
160. "type": "synonyms",
161. "value": "available",
162. "synonyms": [
163. "streamed",