

**Group Member:**

**Bisal Shafiq**

**Ayesha Wasil**

**Ramsha Attaullah Khan**

**Sana Ahmed**

**Department: Software Engineering**

**Subject: Artificial intelligence**

**Class: BS(SE-2019)**

**Year: 4th**

**Semester: 7th**

**Artificial intelligence project: Expert Event Agent**

**Submitted to:**  **Dr. Salahuddin Sheikh**

**Contents**

[**Introduction:** 3](#_Toc106468712)

[**Source Code:** 4](#_Toc106468713)

[**Output:** 8](#_Toc106468714)

# **Introduction:**

In this project, there is an agent which is guiding you about the dress that which dresses are belongs to the event. **Artificial intelligence** dealing with software’s and intelligent agents that perceives that takes actions by perceiving in its environment. The project based on Artificial intelligence technique which is **facts and rules**. This technique is used in this project. A **fact** is a predicate expression that makes a declarative statement about the problem domain. A **rule** is a predicate expression that uses logical implication (:-) to describe a relationship among facts. A knowledge-based system also called as an expert system to solve different problems by a program set. So that we use fact and rules in **Expert Event Agent** which is the name of this project, were guide throughout the fact according to the rules. **Expert Event Agent** is one of the expert systems which is used in event dresses suggestion which dress are suitable for which event. The dress is in wardrobe and user asked the agent which dress are available in wardrobe then agent response the user and asked few questions and user provide the answer the agent give the result according to the user answered. There are five events to which each event dresses are available. If in case of unavailability of any event dress, that dresses will be casual dresses for that result given by the agent to user. Successfully implemented and results have been taken SWI-Prolog widely used and free comprehensive Prolog environment, Prolog (Programming in Logic) expert system.

# **Source Code:**

dress:- event(Make),

write('you have a collection of '),

write(Make),

nl,

undo.

/\* Hypothesis\*/

event(engagement):- engagement, !.

event(baraat):- baraat, !.

event(resaption):- resaption, !.

event(birthdayParty):- birthdayParty, !.

event(fairval):- fairval, !.

event(picnic):- picnic, !.

event(casualDresses).

/\*knowledge\*/

birthdayParty :-

agent(shirtWith\_skirt),

agent(camiDress\_With\_Jacket),

agent(floor\_Length\_Maxi),

agent(heighLong\_Gowns),

agent(jumpSuit),

agent(kurtiWith\_Jeans).

fairval :-

agent(promTulle\_Dress),

agent(blackChiffon\_LongAnarkali),

agent(lineMaxi\_Dress),

agent(fullSleeve\_TopTulleLong\_Skirt),

agent(peplumTop\_Plazo),

agent(velvetCrop\_TopNet\_Gown).

picnic :-

agent(romper),

agent(frontButton\_camiMaxi),

agent(tiered\_Dress),

agent(cami\_Jumpsuit),

agent(overSizeHoodie\_WithJeans),

agent(polkaDot\_Maxi).

engagement :-

agent(shimmerCocktail\_Gown),

agent(blackChiffon\_LongAnarkali),

agent(frontSlited\_KurtiLehenga),

agent(digitalPrinted\_LehengaCholi),

agent(pearlWork\_NetCrape\_Gown),

agent(tulle\_Gown).

baraat :-

agent(fullSleeve\_HeavyWorkAnarkali),

agent(floorLengthAnarkali\_WithBelt),

agent(gown\_Dress),

agent(embroideredLehnga\_Choli),

agent(shortKhurtiWith\_Garara),

agent(choliWith\_EmbroideredSharara).

resaption :-

agent(fullMaxi\_Gown),

agent(floorLengthAnarkali\_WithBelt),

agent(flaredSkirt\_FullsleeveCrop\_Top),

agent(tulleProm\_Dress),

agent(flared\_Lehenga),

agent(cropTop\_Lahenga).

input(Question):-

write('do you have these dresses in your wardrobe.'),

nl,

write(Question),

write('?'),

read(Response),

nl,

((Response == yes; Response == y)

->

assert(yes(Question));

assert(no(Question)), fail).

:- dynamic yes/1,no/1.

agent(S):-

(yes(S)

->

true;

(no(S)

->

fail;

input(S))).

undo :- retract(yes(\_)),fail.

undo :- retract(no(\_)),fail.

undo.

# **Output:**













