college('College of managment', ['business', 'social', 'virtual']).

college('College of science and humanities', ['translation', 'teaching', 'community']).

college('Health College', ['education', 'telehealth', 'mental']).

college('College Eningeering', ['design', 'archuval', 'computer']).

description('business', 'business research if you are interested please apply').

description('social', 'social media if you are interested please apply').

description('virtual', 'virtual teaching if you are interested please apply').

description('translation', 'if you are interested please apply').

description('teaching', 'if you are interested please apply').

description('community', 'community service if you are interested please apply').

description('education', 'health education if you are interested please apply').

description('telehealth', 'if you are interested please apply').

description('mental', 'mental health if you are interested please apply').

description('design', 'if you are interested please apply').

description('archuval', 'archuval research if you are interested please apply').

description('computer', 'computer-aided if you are interested please apply').

user(zainab, 219).

user\_opportunity(zainab, 'College of science and humanities', 'graduaction').

:- dynamic user/2.

:- dynamic user\_opportunity/3.

login :-

write('Please enter your username: '),

read(Username),

write('Please enter your password: '),

read(Password),

(user(Username, Password) ->

write('Login successful!'),

select\_college(Username);

write('Invalid username or password try again.\n'),

login).

signup :-

write('Please enter your desired username: '),

read(Username),

write('Please enter your desired password: '),

read(Password),

write('Please confirm your password: '),

read(ConfirmPassword),

(Password \= ConfirmPassword ->

write('Passwords do not match. Please try again.'),

signup;

assert(user(Username, Password)),

write('Signup successful!'),

select\_college(Username)).

select\_college(Username) :-

write('Please choose the number of the college you want:\n'),

write('1. College of managment \n'),

write('2. College of science and humanities \n'),

write('3. Health College \n'),

write('4. College Eningeering \n'),

read(X),

(X = 1 -> College = 'College of managment'; X = 2 -> College = 'College of science and humanities'; X = 3 -> College = 'Health College'; X = 4 -> College='College Eningeering'),

(college(College, Opportunities) ->

display\_opportunities(Opportunities),

select\_opportunity(Username, College, Opportunities);

write('Invalid college. Please try again.'),

select\_college(Username)).

% Define the volunteer opportunity selection predicate

select\_opportunity(Username, College, Opportunities) :-

write('Please write the name of the volunteer opportunity: '),

read(Opportunity),

(member(Opportunity, Opportunities) ->

assert(user\_opportunity(Username, College, Opportunity)),

write('Application submitted!') ;

write('Invalid name. Please try again. '),

select\_opportunity(Username, College, Opportunities)).

% Define the predicate to display the description for each opportunity

display\_opportunities([]).

display\_opportunities([Opportunity | Opportunities]) :-

write(Opportunity),

description(Opportunity, Descr),

write(': '),

write(Descr),

nl,

display\_opportunities(Opportunities).

main :-

write('Welcome to the volunteer hub!\n'),

write('Please login or signup to continue:\n'),

write('1. Login\n'),

write('2. Sign up\n'),

read(Choice),

(Choice = 1 -> login; Choice = 2 -> signup).