

```

1 // Author: John Miller
2 // Email: john.t.miller@okstate.edu
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <unistd.h>
7 #include <string.h>
8 #include <stdbool.h>
9 #include "Assistant.h"
10 #include "History.h"
11 #include "Query.h"
12 #include "Record.h"
13
14 #define MAX_HISTORY_CHARS 500
15 #define MAX_HISTORY_LINES 10
16
17 void getFromHistory(Query search, Record *result){
18     Record found;
19
20     FILE *historyFile;
21     historyFile = fopen("history.txt", "r");
22     char line[MAX_HISTORY_CHARS];
23
24     if(historyFile){
25         while(fgets(line, sizeof(line), historyFile)){
26
27             //split line into array of strings on commas
28             int s=0;
29             char *nextTokenPnt = strtok(line, "\t");
30             char *substrings[14];
31             while(nextTokenPnt != NULL){
32                 substrings[s] = nextTokenPnt;
33                 s++;
34                 nextTokenPnt = strtok(NULL, "\t");
35             }
36
37             //printf("%s, %s, %s, %s\n",substrings[0], substrings[1], substrings[2], substrings[6]);
38             if(strcmp(substrings[1], search.employee_name)==0 && strcmp(substrings[2], search.job_title)==0 &&
strcmp(substrings[6], search.status)==0){ //employee is in history
39                 found.id = atoi(substrings[0]);
40                 strcpy(found.employee_name,substrings[1]);
41                 strcpy(found.job_title,substrings[2]);
42                 found.base_pay = strtod(substrings[3],NULL);
43                 found.overtime_pay = strtod(substrings[4],NULL);
44                 found.benefit = strtod(substrings[5],NULL);
45                 strcpy(found.status, substrings[6]);
46                 found.satisfaction_level = strtod(substrings[7],NULL);
47                 found.number_project = atoi(substrings[8]);
48                 found.average_monthly_hours = atoi(substrings[9]);
49                 found.time_spend_company_in_yrs = atoi(substrings[10]);
50                 found.work_accident = atoi(substrings[11]);
51                 found.promotion_last_5years = atoi(substrings[12]);
52
53                 break;
54             }
55         }
56     }
57     else
58     {
59         printf("Error Reading File: history.txt\n");
60     }
61
62     fclose(historyFile);
63     *result = found;
64 }
65

```

```

66 bool isInHistory(Query search){
67     FILE *historyFile;
68     historyFile = fopen("history.txt", "r");
69     char line[MAX_HISTORY_CHARS];
70     bool found = false;
71     char *rest = line;
72
73     if(historyFile){
74         while(fgets(line, sizeof(line), historyFile)){
75
76
77             //split line into array of strings on commas
78             int s=0;
79             char *nextTokenPnt = strtok_r(line, "\t", &rest);
80             char *substrings[13];
81
82             while(nextTokenPnt != NULL){
83                 substrings[s] = nextTokenPnt;
84                 s++;
85                 nextTokenPnt = strtok_r(NULL, "\t", &rest);
86             }
87
88             // printf("%s,%s,%s,%s\n",substrings[0], substrings[1], substrings[2], substrings[6]);
89             // printf("search: employee name: %s jobtitle %s status %s", search.employee_name, search.job_title,
search.status);
90             if(strcmp(substrings[1], search.employee_name)==0 && strcmp(substrings[2], search.job_title)==0 &&
strcmp(substrings[6], search.status)==0){ //employee is in history
91                 found = true;
92                 break;
93             }
94
95         }
96     }
97     else{
98         printf("Error Reading File: history.txt\n");
99     }
100
101     fclose(historyFile);
102
103     return found;
104 }
105
106 void writeToHistory(Record *record){
107     char history[MAX_HISTORY_LINES+1][MAX_HISTORY_CHARS];
108
109     char writeLine[MAX_HISTORY_CHARS];
110
111     sprintf(writeLine,"%d\t%s\t%s\t%f\t%f\t%f\t%f\t%d\t%d\t%d\t%d\t%d\n", record->id,record->
employee_name,record->job_title,record->base_pay,record->overtime_pay,record->benefit,record->status,record->
satisfaction_level,record->number_project,record->average_monthly_hours,record->time_spend_company_in_yrs,record
->work_accident,record->promotion_last_5years);
112     //strcpy(writeLine, "TESTING\n");
113     // printf("Writing to history:\n%s\n", writeLine);
114
115     FILE *historyFile;
116     historyFile = fopen("history.txt", "r");
117     char line[MAX_HISTORY_CHARS];
118
119     int lineNumber = 0;
120
121     if(historyFile){
122         while(fgets(line,sizeof(line), historyFile)){
123             strcpy(history[lineNumber],line);
124             lineNumber++;
125         }
126

```

```

127     fclose(historyFile);
128
129     if(lineNumber<MAX_HISTORY_LINES){//just append
130         strcpy(history[lineNumber],writeLine);
131         lineNumber++;
132         //write to file 0-9
133         FILE *writeHistoryFile;
134         writeHistoryFile = fopen("history.txt", "w+");
135
136         if(writeHistoryFile){
137             for(int i=0; i<lineNumber; i++){
138                 fprintf(writeHistoryFile,"%s",history[i]);
139             }
140         }
141         else{
142             printf("Error Writing File: history.txt\n");
143         }
144         fclose(writeHistoryFile);
145     }
146     else{
147         strcpy(history[MAX_HISTORY_LINES],writeLine);
148         //write to file 1-10
149         FILE *writeHistoryFile;
150         writeHistoryFile = fopen("history.txt", "w+");
151
152         if(writeHistoryFile){
153             for(int i=1; i<=MAX_HISTORY_LINES; i++){
154                 fprintf(writeHistoryFile,"%s",history[i]);
155             }
156         }
157         else{
158             printf("Error Writing File: history.txt\n");
159         }
160         fclose(writeHistoryFile);
161     }
162 }
163 else
164 {
165     printf("Error Reading File: history.txt\n");
166     fclose(historyFile);
167 }
168

```