#include"Menus.c"

struct node

int NodeHeight

char Username[30]

char Password[30]

struct node \*left

struct node \*right

//Nambah didalam ini diganti ama nama structnya

struct treeNode \*root[10]

char SecurityAnswer[30]

char SecurityQuestion[300]

int key

char UsernameCurrent[30];

int HashKey;

int Height(struct node \*NodeInput):

if NodeInput is NULL

return 0

else

return NodeInput->NodeHeight

END if

END Function

int Max(int num1,int num2):

if num1>num2

return num1

else

return num2

END if

END Function

int CheckBalance(struct node \*NodeInput):

if NodeInput is NULL

return 0

else

return Height(NodeInput->left)-Height(NodeInput->right)

END if

END Function

struct node \*RightRotate(struct node \*OldCurrent)

struct node \*NewCurrent = OldCurrent->left

struct node \*RightNewCurrent=NewCurrent->right

NewCurrent->right=OldCurrent

OldCurrent->left=RightNewCurrent

OldCurrent->NodeHeight=1 + Max(Height(OldCurrent->left),Height(OldCurrent->right))

NewCurrent->NodeHeight=1 + Max(Height(NewCurrent->left),Height(NewCurrent->right))

return NewCurrent

END Function

struct node \*LeftRotate(struct node \*OldCurrent):

struct node \*NewCurrent=OldCurrent->right

struct node \*LeftNewCurrent = NewCurrent->left

NewCurrent->left=OldCurrent

OldCurrent->right=LeftNewCurrent

OldCurrent->NodeHeight= 1 + Max(Height(OldCurrent->left),Height(OldCurrent->right))

NewCurrent->NodeHeight= 1 + Max(Height(NewCurrent->left),Height(NewCurrent->right))

return NewCurrent

END Function

struct node \*MinVal (struct node \*root):

struct node \*Current = root

while Current->left is not NULL

Current=Current->left

END while

return Current

END Function

struct node \*insert(struct node \*root,char \*UsernameInput, char \*PasswordInput,int keyInput, char \*SecurityQuestionIn, char \*AnswerIn):

if root is NULL

struct node \*Newnode = (struct node\*)malloc(sizeof(struct node))

int i

for(i=0;i<10;i++) Newnode->root[i] = NULL

strcpy(Newnode->Username,UsernameInput)

strcpy(Newnode->Password,PasswordInput)

strcpy(Newnode->SecurityQuestion,SecurityQuestionIn)

strcpy(Newnode->SecurityAnswer,AnswerIn)

Newnode->key=keyInput

Newnode->NodeHeight=1

Newnode->right=NULL

Newnode->left=NULL

return Newnode

END if

If keyInput<root->key

root->left=insert(root->left,UsernameInput,PasswordInput,keyInput,SecurityQuestionIn,AnswerIn)

else if keyInput>root->key

root->right=insert(root->left,UsernameInput,PasswordInput,keyInput,SecurityQuestionIn,AnswerIn);

END if

root->NodeHeight= 1 + Max(Height(root->left),Height(root->right))

int i

for(i=0;i<10;i++) root->root[i] = NULL

int BalanceFactor = CheckBalance(root)

if BalanceFactor>1 && keyInput<root->left->key

return RightRotate(root)

END if

If BalanceFactor>1 && keyInput>root->left->key

root->left=LeftRotate(root->left)

return RightRotate(root)

END if

If BalanceFactor<-1 && keyInput>root->right->key

return LeftRotate(root)

END if

If BalanceFactor<-1 && keyInput<root->right->key

root->right=RightRotate(root->right)

return LeftRotate(root)

END if

return root

END Function

struct node \*DeleteNode(struct node \*root, int key):

if root is NULL

return root

END if

If key<root->key

root->left=DeleteNode(root->left,key)

else if key>root->key

root->right=DeleteNode(root->right,key)

else

if root->left is NULL

struct node \*temp = root->right

free(root)

return temp

else if root->right is NULL

struct node \*temp = root->left

free(root)

return temp

else

struct node \*temp = MinVal(root->right)

root->key=temp->key

root->right=DeleteNode(root->right,temp->key)

END if

END if

root->NodeHeight= 1 + Max(Height(root->left),Height(root->right));

int BalanceFactor = CheckBalance(root);

if BalanceFactor>1 && CheckBalance(root->left)>=0

return RightRotate(root)

END if

If BalanceFactor>1 && CheckBalance(root->left)<0

root->left=LeftRotate(root->left)

return RightRotate(root)

END if

If BalanceFactor<-1 && CheckBalance(root->right)<=0

return LeftRotate(root)

END if

If BalanceFactor<-1 && CheckBalance(root->right)>0

root->right=RightRotate(root->right)

return LeftRotate(root)

END if

return root

END Function

struct node \*Search(struct node \*root,int KeyString):

if root is NULL

return root

END if

If KeyString is equal to root->key

return root

else if KeyString<root->key

return Search(root->left,KeyString)

else if(KeyString>root->key

return Search(root->right,KeyString)

else return NULL

END if

END Function

void CreateAccount(struct node \*\*root):

char Usernametest[30],Passwordtest[30]

int length,UsernameSum

struct node \*Data

Display"<<Create Account>>\n\n"

do

Display"Username[5-30 chars]: "

Input Usernametest

getchar()

length = strlen(Usernametest)

UsernameSum=0

For int i=0;i<length;i++

If i<2

UsernameSum+=Usernametest[i]%10

else

UsernameSum+=Usernametest[i]

END if

END for

Data = Search((\*root),UsernameSum)

If Data is not NULL

Display"Username is taken\n"

END if

While Data is not NULL || length < 5 || length >30

END DO While

do

Display"Password[5-30 chars]: "

Input Passwordtest

getchar()

length = strlen(Passwordtest)

while length<5 || length>30

END Do while

char SecurityQuestion[300],Answer[30]

int menu

do

Display"\n<<Choose one of security question below>>\n"

Display"1. What is the middle name of your youngest child?\n"

Display"2. What was the name of your first stuffed animal?\n"

Display"3. In what city or town did your mother and father meet?\n"

Display"4. What was the first exam you failed?\n"

Display">> "

Input menu

switch(menu)

case 1:

system("cls")

Display"1. What is the middle name of your youngest child?\n"

strcpy(SecurityQuestion,"What is the middle name of your youngest child?")

break

case 2:

system("cls")

Display"2. What was the name of your first stuffed animal?\n"

strcpy(SecurityQuestion,"What was the name of your first stuffed animal?")

break

case 3:

system("cls")

Display"3. In what city or town did your mother and father meet?\n"

strcpy(SecurityQuestion,"In what city or town did your mother and father meet?")

break

case 4:

system("cls")

Display"4. What was the first exam you failed?\n"

strcpy(SecurityQuestion,"What was the first exam you failed?")

break

END Switch

While menu<1 || menu > 4

END While

Display "Answer: "

fflush(stdin)

Input Answer

getchar()

length = strlen(Answer)

for int i=0;i<length;i++

if Answer[i]<'a'

Answer[i]= Answer[i]+32

END if

END for

\*root = insert((\*root),Usernametest,Passwordtest,UsernameSum,SecurityQuestion,Answer)

Display"\n<<Account Created>>\n"

END Function

void ChangePass(struct node \*root):

int length,Sum=0

Display"<<Change Password>>\n\n"

length = strlen(UsernameCurrent)

for int i=0;i<length;i++

if i<2

Sum+=UsernameCurrent[i]%10

else

Sum+=UsernameCurrent[i]

END if

END for

struct node \*Data = Search(root,Sum)

char Passwordtest[30]

do

Display"New Password[5-30 chars]: "

Input Passwordtest

getchar()

length = strlen(Passwordtest)

while length<5 || length>30

END Do while

strcpy(Data->Password,Passwordtest)

Display"<<Password Changed!!>>\n\n"

END Function

void DeleteAccount(struct node \*\*root):

int length

char UsernameCheck[30]

Display"<<Delete Account>>\n\n"

Display"Username: "

Input UsernameCheck

getchar()

length=strlen(UsernameCheck)

int UsernameSum=0

for int i=0;i<length;i++

if i<2

UsernameSum+=UsernameCheck[i]%10

else

UsernameSum+=UsernameCheck[i]

END if

END for

struct node \*Data = Search(\*root,UsernameSum)

if Data is NULL

Display"Account Doesnot Exist\n"

return

else

char PasswordCheck[30]

Display"Password: "

Input PasswordCheck

If strcmp(PasswordCheck,Data->Password)is not 0

Display"Invalid Password\n"

return

else

int menu

do

Display"Are you sure you want to delete your current account?\n"

Display"1. Yes\n"

Display"2. No\n"

Display">> "

Input menu

switch(menu)

case 1:

(\*root) = DeleteNode((\*root),UsernameSum)

Display"Delete Success\n"

break

case 2:

Display"Going back to menu\n"

break

END switch

While menu<1 || menu>2

End Do while

END if

END if

END Function

void mainMenu(struct node \*node):

char temp = 'a'

int i

strcpy(session.name, node->Username)

for i = 0; i < 10; i++

session.hashRoot[i] = node->root[i]

END for

While temp is not '0'

Display"----------------------------\n Hello, %s\n----------------------------\n", session.name

Display"0. Log Out\n1. Change my Password Manager password\n2. Add New\n3. View\n4. Delete\n"

temp = getch()

switch(temp)

case '0':

system("cls")

for(i=0;i<10;i++) node->root[i] = session.hashRoot[i]

session\_destroy()

Display"\nLogged Out, “

return

case '1':

system("cls")

ChangePass(node)

break

case '2':

system("cls")

add()

break

case '3'

system("cls")

view()

break

case '4':

system("cls")

delete()

break

END Switch

system("cls")

END while

END Function

void LoginAccount(struct node \*root):

if root is NULL

Display"No account registered\n"

return

END if

char UsernameCheck[30],PasswordCheck[30]

Display"<<Login>>\n\n"

Display"Username: "

Input UsernameCheck

getchar()

int length=strlen(UsernameCheck)

int UsernameSum=0

for int i=0;i<length;i++)\

if i<2

UsernameSum+=UsernameCheck[i]%10

else

UsernameSum+=UsernameCheck[i]

END if

END for

struct node \*Data = Search(root,UsernameSum)

if Data is NULL

Display "Account Doesnot Exist\n"

return

else

strcpy(UsernameCurrent,UsernameCheck)

int limit=3,limit2=3

do

Display"Password: "

Input PasswordCheck

getchar()

if strcmp(PasswordCheck,Data->Password)is not 0

limit--

Display"Invalid Password\n"

else

break

END if

While limit>0

END Do While

If limit is 0

int menu,length2

do

system("cls")

char AnswerCheck[30],NewPassword[30]

Display"Forgot Password?\n"

Display"1. Yes\n"

Display"2. No\n"

Display">> "

Input menu

switch(menu){

case 1:

do

system("cls")

Display"<<Answer this question>>[Remaining Attempt: %d]\n",limit2

Display"%s\n",Data->SecurityQuestion

Display">> "

Input AnswerCheck

length2 = strlen(AnswerCheck)

for int i=0;i<length2;i++

if AnswerCheck[i]<'a'

AnswerCheck[i]= AnswerCheck[i]+32

END if

END for

If strcmp(Data- >SecurityAnswer,AnswerCheck)is not 0

limit2--

continue

else

Display"Enter a new Password\n"

Input NewPassword

strcpy(Data->Password,NewPassword)

getchar()

return

END if

while(limit2>0)

END do while

system("cls")

Display"<<Attempt limit is reached>>\n"

Display"<<Your account is now blocked>>\n"

Display"<<Please contact our customer service>>\n"

Display"CS (Davin): 082233445566\n"

return

case 2:

return

END Switch

While menu<1 || menu >2

END do While

End if

END if

Display"\n\n<<Login Success>>\n\n"

system("pause")

system("cls")

mainMenu(Data)

END Function

int main():

struct node \*root=NULL

char menu

session\_destroy()

do

Display"<<Password Manager>>\n\n"

Display"<<Menu>>\n"

Display"1. Create Account\n"

Display"2. Login\n"

Display"3. Delete Account\n"

Display"4. Exit\n"

menu = getch()

switch(menu)

case '1':

system("cls")

CreateAccount(&root)

system("pause")

break

case '2':

system("cls")

LoginAccount(root)

system("pause")

break

case '3':

system("cls")

DeleteAccount(&root)

system("pause")

break

case '4':

return 0

END Switch

system("cls"

while(menu!='4')

End Do while

END Function