Development of a feature-rich, Shopping Management System

**A Project Report for Internship Programme**

###### ***Submitted by***

##### Arijit Chakraborty

##### Subhradip Kundu

##### Kajal Roy

##### Prabir Pal

***in partial fulfillment for the award of the degree of***

##### **B.TECH**

in

**ELECTRONIS AND COMMUNICATION ENGINEERING**

Narula Institute of Technology

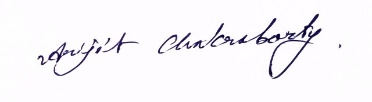


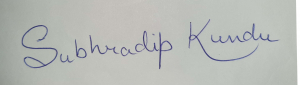
At

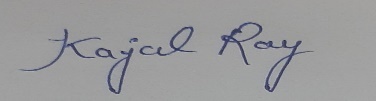
**BONAFIDE CERTIFICATE**

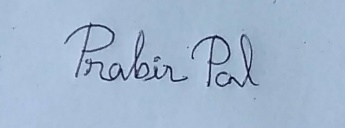
Certified that this project work was carried out under my supervision

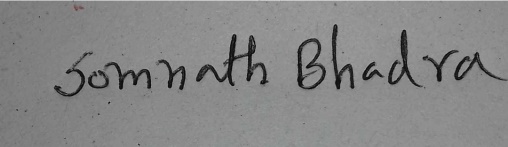
***“Development* of a feature-rich, practical Shopping Management System** “is the bonafide work of

***Name of the student: Arijit Chakraborty Signature:*** ******

***Name of the student: Subhradip Kundu Signature:*** ******

***Name of the student: Kajal Roy Signature:*** ******

***Name of the student: Prabir Pal Signature: ***

****

**SIGNATURE :**

Name : **SOMNATH BHADRA**

**PROJECT MENTOR**

**Acknowledgement**

I take this opportunity to express my deep gratitude and sincerest thank to my project mentor, Mr.**Somnath Bhadra** for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

I will like to give a special mention to my colleagues. Last but not the least I am grateful to all the faculty members of Euphoria GenX or their support.

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **Description** | **Page No.** |
| 1 | Title of the project | 5 |
| 2 | Introduction and Objectives of the Project | 5 |
| 3 | Project Category (RDBMS/OOPS/Networking/Multimedia/Artificial Intelligence/Expert Systems etc.) | 5 |
| 4 | Tools/Platform, Hardware and Software Requirement specifications | 5-6 |
| 5 | Goals of Implementation | 6 |
| 6 | Non-functional Requirements | 6-8 |
| 7 | Feasibility Study | 9-11 |
| 8 | Project Planning | 11 |
| 9 | User Interface Design | 12-13 |
| 10 | Coding | 14-73 |
| 11 | Testing | 73-77 |
| 12 | Cost Estimation of the Project along with Cost Estimation Model | 77-78 |
| 13 | Future scope and further enhancement of the Project | 78 |
| 14 | Bibliography | 78 |

1. Title of the project :-

Development of a feature-rich, Shopping Management System.

1. Introduction and Objectives of the Project :-

The main objective of the project is to help to the ongoing users help to attain an easy way to negative the customer’s problems. Maintaining the value of materials. It is basically a very instant processing system by which customer can get their product with a proper packaging and invoice bill. It’s basically build in the C++ platform.

**PROJECT CATEGORY:-**

Desktop Application

1. Tools/Platform, Hardware & Software Requirement specifications :-

* **Tools :-**

Block code

Ms-Office

* **Platform :-**

Microsoft Windows 10

**Hardware Requirement Specification**

|  |  |
| --- | --- |
| **Client Machine** | |
| **HDD** | 200 MB |
| **Processor** | Pentium 4 or newer processor that supports SSE2 |
| **Memory** |  |

1. Goals of Implementation :-

The implementation aims to simplify shopping for a customer.

1. Non Functional Requirements :-

In addition to the obvious features and functions that you will provide in your system, there are other requirements that don't actually DO anything, but are important characteristics nevertheless. These are called "non-functional requirements" or sometimes "Quality Attributes." For example, attributes such as performance, security, usability, compatibility. Aren’t a "feature" of the system, but are a required characteristic. You can't write a specific line of code to implement them; rather they are "emergent" properties that arise from the entire solution. The specification needs to describe any such attributes the customer requires. You must decide the kind of requirements that apply to your project and include those that are appropriate.

Each requirement is simply stated in English. Each requirement must be objective and quantifiable; there must be some measurable way to assess whether the requirement has been met.

Often deciding on quality attributes requires making tradeoffs, e.g., between performance and maintainability. In the APPENDIX you must include an engineering analysis of any significant decisions regarding tradeoffs between competing attributes.

Here are some examples of non-functional requirements:

* **Performance requirements :-**

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.

* **Operating constraints :-**

List any run-time constraints. This could include system resources, people, and needed software.

The application must run without any manual intervention.

* **Platform constraints :-**

Discuss the target platform. Be as specific or general as the user requires. If the user doesn't care, there are still platform constraints.

Since the application will be developed in PHP it is platform independent.

* **Accuracy and Precision :-**

Requirements about the accuracy and precision of the data. (Do you know the difference?) Beware of 100% requirements; they often cost too much.

* **Modifiability :-**

Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).

Minimal

* **Portability :-**

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

Minimal

* **Reliability :-**

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

* **Security :-**

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time ...) to break into the system. Do not discuss solutions (e.g. passwords) in a requirements document.

Only secured users can access the application.

No one can go to any independent page without logging in.

* **Usability :-**

Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

* **Legal :-**

There may be legal issues involving privacy of information, intellectual property rights, export of restricted technologies, etc.

1. Feasibility Study :-

You should provide a feasibility report in the following format:

* **Product:**A general statement of the product; give a brief description of what the proposed system will do, highlighting where the proposed system meets the specified business requirements of the organization.
* **Technical Feasibility:**Will the proposed system perform to the required specification? Outline technical systems options you propose to use, which will give a technical solution satisfying the requirements and constraints of the system, as outlined in the terms of reference.
* **Social Feasibility:**Consideration of whether the proposed system would prove acceptable to the people who would be affected by its introduction. Describe the effect on users from the introduction of the new system; consider whether there will be a need for retraining the workforce. Will there be a need for relocation of some of the workforce? Will some jobs become deskilled? Will the current workforce be able to perform effectively any new tasks introduced by the proposed system? Describe how you propose to ensure user co-operation before changes are introduced.
* **Economic Feasibility:**Consider the cost/benefits of the proposed system. Detail the costs that will be incurred by the organization adopting the new system; consider development costs and running costs. Detail benefits that the new system will bring, direct economic benefits such as reduced costs, and indirect benefits, such as improved management information and better customer service. Illustrate the cost/benefit of the new system by applying a suitable cost/benefit analysis method such as the payback method.
* **Market Research:**A comprehensive market research identifying a need for the product. Detail all market research you carried out, listing sources of information. Justify any conclusions you have drawn from your research. Identify the potential customer base for your product, together with evidence of customer need for the product. Describe how you propose to compete with similar products on the market.
* **Alternative Solution:**Consideration of alternative solutions should be documented. At least two alternative business or technical systems options should be considered. Detail the differences between these options and the proposed system. Justify your choice of the proposed system and the reasons for rejecting the alternative options.

At this point, all of the planning for the project has been done and if the feasibility study has shown that the project is likely to succeed within its constraints, then it only remains for us to start the requirements analysis and thus proceed with the project.

* **Feasibility study :-**

You should provide a feasibility report in the following format:

* Product: A general statement of the product; give a brief description of what the proposed system will do, highlighting where the proposed system meets the specified business requirements of the organization.
* Technical Feasibility: Will the proposed system perform to the required specification? Outline technical systems options you propose to use, which will give a technical solution satisfying the requirements and constraints of the system, as outlined in the terms of reference.
* Social Feasibility: Consideration of whether the proposed system would prove acceptable to the people who would be affected by its introduction. Describe the effect on users from the introduction of the new system; consider whether there will be a need for retraining the workforce. Will there be a need for relocation of some of the workforce? Will some jobs become deskilled? Will the current workforce be able to perform effectively any new tasks introduced by the proposed system? Describe how you propose to ensure user co-operation before changes are introduced.
* Economic Feasibility: Consider the cost/benefits of the proposed system. Detail the costs that will be incurred by the organization adopting the new system; consider development costs and running costs. Detail benefits that the new system will bring, direct economic benefits such as reduced costs, and indirect benefits, such as improved management information and better customer service. Illustrate the cost/benefit of the new system by applying a suitable cost/benefit analysis method such as the payback method.
* Market Research: A comprehensive market research identifying a need for the product. Detail all market research you carried out, listing sources of information. Justify any conclusions you have drawn from your research. Identify the potential customer base for your product, together with evidence of customer need for the product. Describe how you propose to compete with similar products on the market.

Alternative Solution: Consideration of alternative solutions should be documented. At least two alternative business or technical systems options should be considered. Detail the differences between these options and the proposed system. Justify your choice of the proposed system and the reasons for rejecting the alternative options. At this point, all of the planning for the project has been done and if the feasibility study has shown that the project is likely to succeed within its constraints, then it only remains for us to start the requirements analysis and thus proceed with the project.

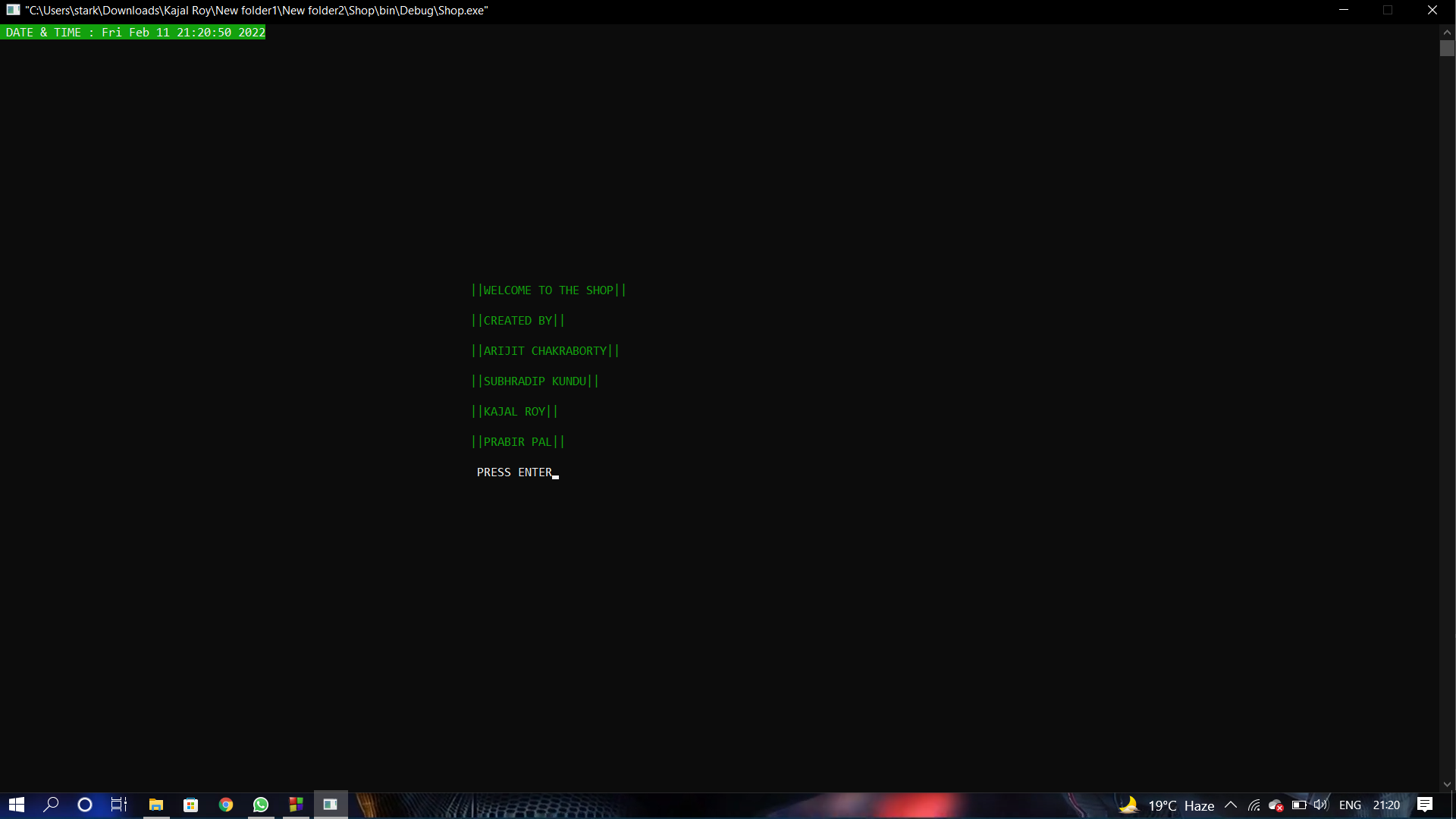
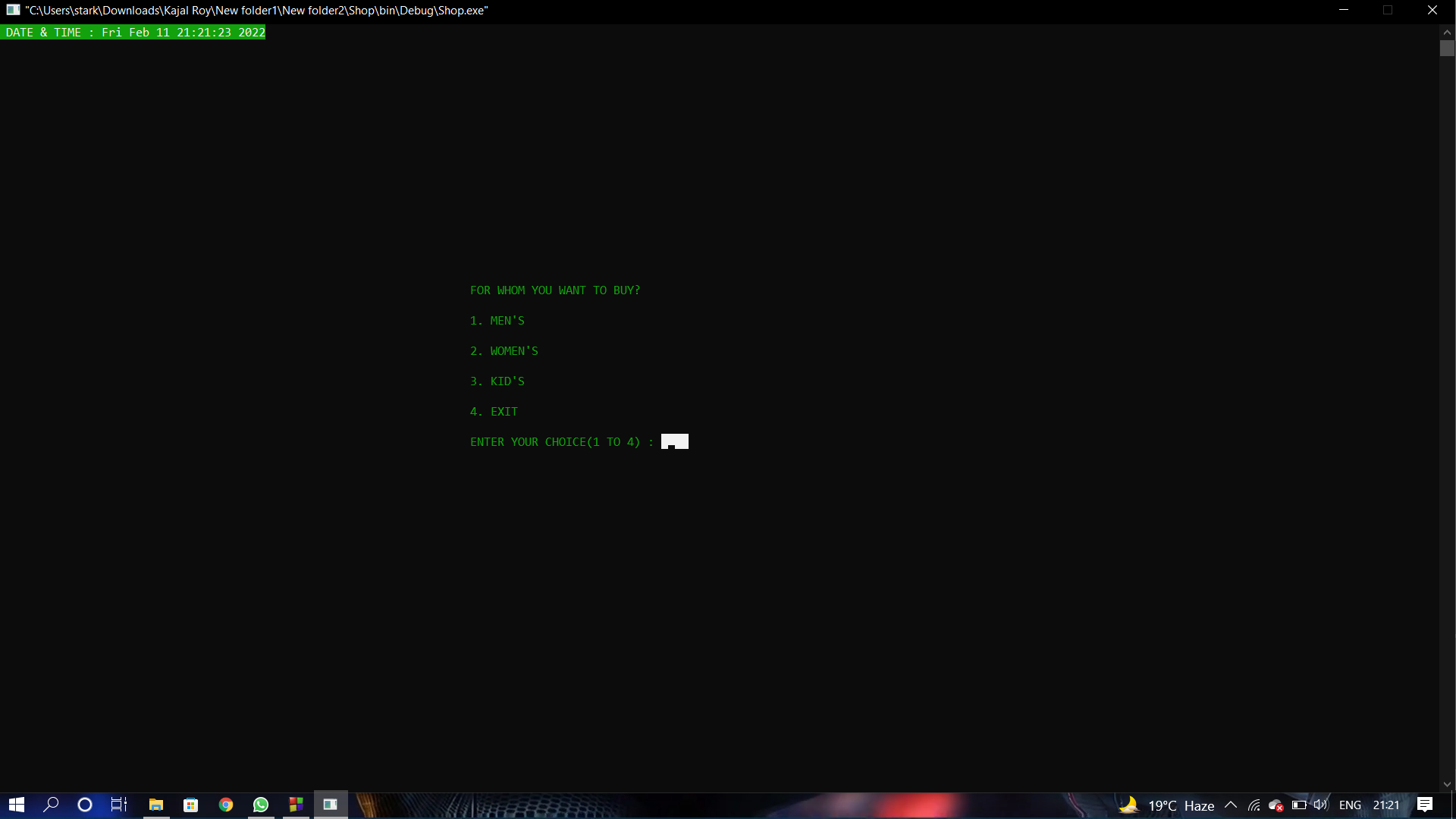
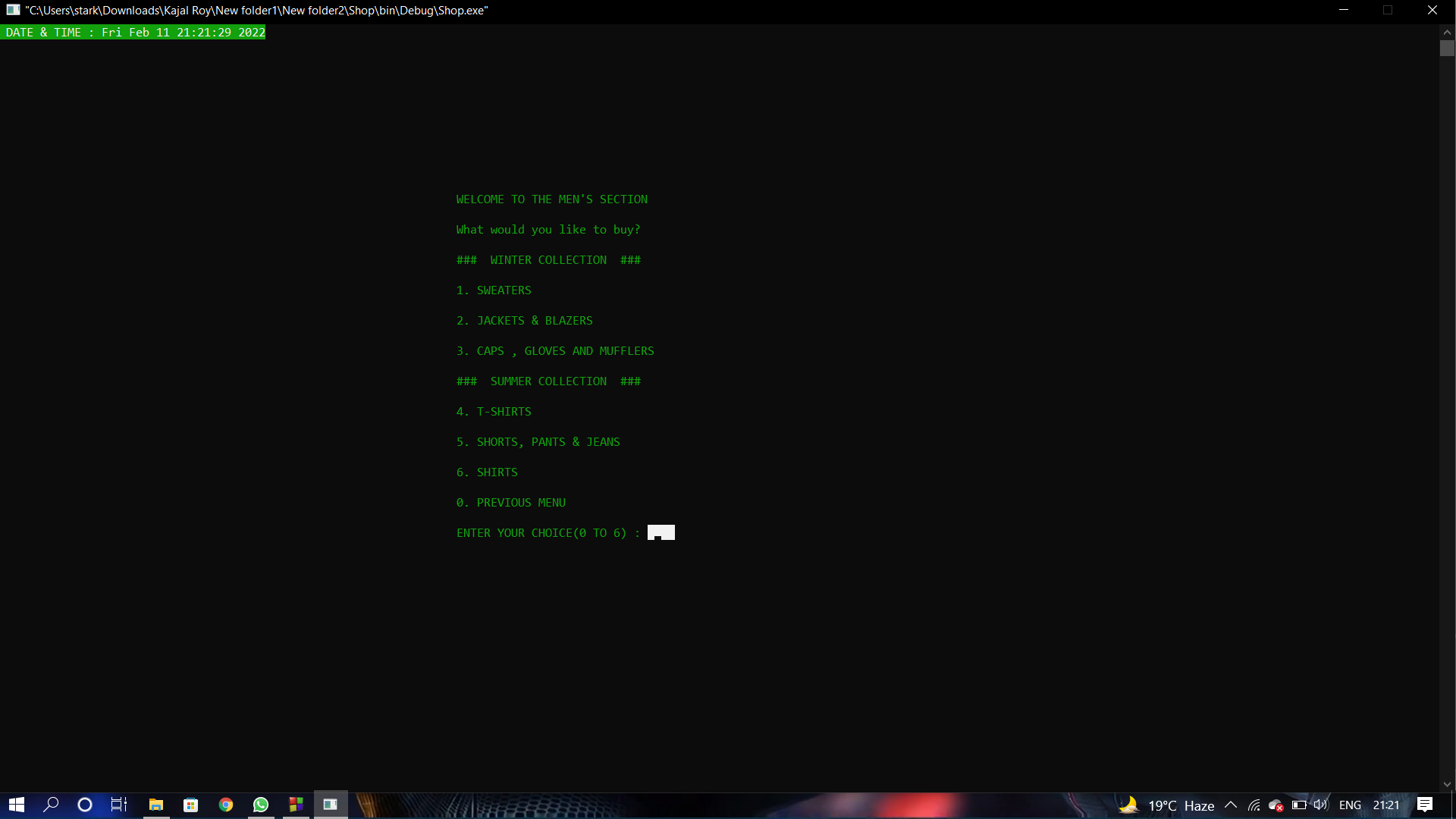
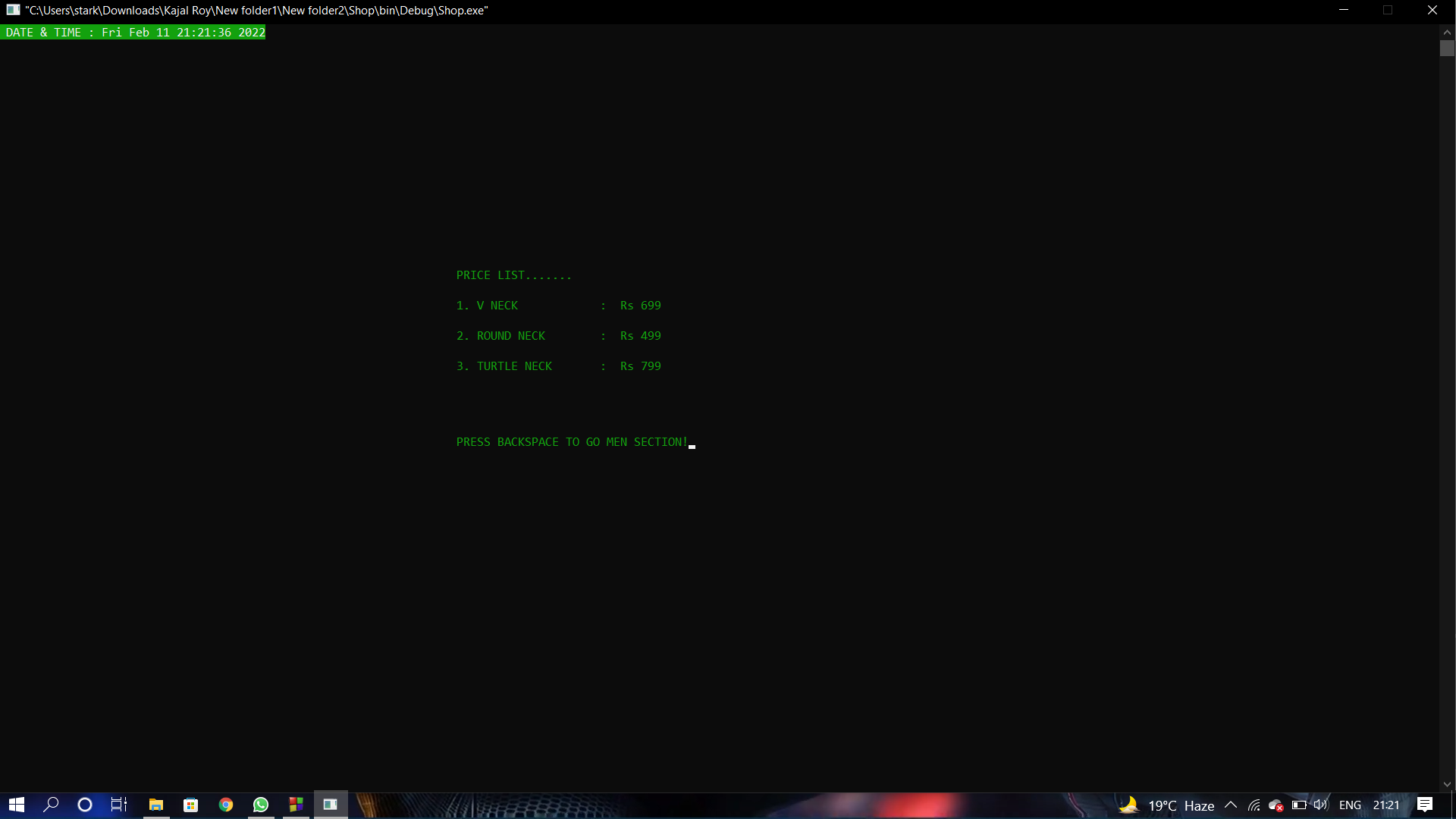
1. Project Planning :-

Project planning is concerned with identifying the following for every project:

* Activities
* Milestones
* Deliverable.

A plan must be drawn up to guide the development towards the project goal. A plan is drawn up at the start of a project. This plan should be used as the driver for the project. The initial plan is not static, and must be modified as the project progresses.

Planning is required for development activities from specification through to delivery of the system.

1. User Interface Design :-
2. Coding:-

**Main.cpp**

#include<bits/stdc++.h>

#include<fstream>

#include<conio.h>

#include"../../../conio2.h"

#include"../../../men\_section.h"

#include"../../../women\_section.h"

#include"../../../windowms.h"

#include"../../../locTime.h"

#include"../../../kids\_section.h"

#define el cout<<endl;

using namespace std;

void window1();

void welcome()

{

time();

int a = 70;

int b = 18;

gotoxy(a,b);

string element[]={"||WELCOME TO THE SHOP||", "||CREATED BY||", "||ARIJIT CHAKRABORTY||", "||SUBHRADIP KUNDU||", "||KAJAL ROY||", "||PRABIR PAL||"};

for(int i = 0;i<6;i++)

{

textcolor(GREEN);

gotoxy(a+3,b+2\*i);

cout<<element[i];

}

{

textcolor(WHITE);

gotoxy(a+3,b+2\*6);

cout<<" PRESS ENTER";

}

int choice = getch();

switch(choice)

{

case 13:

window1();

break;

}

}

void window1()

{

int a = 70;

int b = 18;

system("cls");

time();

gotoxy(a,b);

textcolor(GREEN);

string element[]={"FOR WHOM YOU WANT TO BUY?", "1. MEN'S", "2. WOMEN'S", "3. KID'S", "4. EXIT"};

for(int i = 0;i<5;i++)

{

textcolor(GREEN);

gotoxy(a+1,b+2\*i);

cout<<element[i];

}

gotoxy(a+1,b+2\*5);

cout<<"ENTER YOUR CHOICE(1 TO 4) : ";

textbackground(WHITE);

cout<<" ";

textbackground(BLACK);

int choice;

gotoxy(a+29,b+10);

cin>>choice;

textbackground(BLACK);

switch(choice)

{

case 1:

menmain();

break;

case 2:

womenmain();

break;

case 3:

kidsmain();

break;

case 4:

exit(1);

break;

default:

gotoxy(a,b+12);

cout<<"WRONG INPUT!";

}

}

int main()

{

MoveWindowSize(-8,-5,2000,2000);

welcome();

}

**kids.cpp**

#include<bits/stdc++.h>

#include<fstream>

#include<conio.h>

#include"../conio2.h"

#include"../locTime.h"

#define el cout<<endl;

using namespace std;

int kids(int a, int b,string mn[]);

void kid\_sweaters\_jackets(int a, int b);

void kid\_caps\_hats(int a,int b);

void kid\_BODYSUITS(int a,int b);

void kid\_ROMPERS\_SLEEPSUITS(int a,int b);

void kid\_CLOTHINGSET(int a,int b);

void window1();

void kidsmain()

{

string mn[8]=

{

"WELCOME TO THE KIDS'S SECTION",

"What would you like to buy? ",

"1. SWEATERS AND JACKETS",

"2. CAPS AND HAT",

"3. BODYSUITS",

"4. ROMPERS&SLEEPSUITS",

"5. CLOTHINGSET",

"0. PREVIOUS MENU",

};

int a=68, b=12;

kids(a,b,mn);

}

int kids(int a, int b, string mn[])

{

system("cls");

time();

int i=0;

while(i<8)

{

gotoxy(a,b+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+22);

cout<<"ENTER YOUR CHOICE(0 TO 5) : ";

textbackground(WHITE);

cout<<" ";

textbackground(BLACK);

int choice;

gotoxy(a+29,b+22);

cin>>choice;

textbackground(BLACK);

switch(choice)

{

case 1:

kid\_sweaters\_jackets(a,b);

break;

case 2:

kid\_caps\_hats(a,b);

break;

case 3:

kid\_BODYSUITS(a,b);

break;

case 4:

kid\_ROMPERS\_SLEEPSUITS(a,b);

break;

case 5:

kid\_CLOTHINGSET(a,b);

break;

case 0:

window1();

break;

default:

gotoxy(a,b+25);

cout<<"WRONG INPUT!";

}

}

void kid\_sweaters\_jackets(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. V NECK\t\t: Rs 699 ",

"2. ROUND NECK\t: Rs 499",

"3. TURTLE NECK\t: Rs 799",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

kidsmain();

break;

case 27:

exit(1);

break;

}

}

void kid\_caps\_hats(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. JEANS JACKET\t: Rs 899",

"2. LEATHER JACKET\t: Rs 1499",

"3. HOODIE\t\t: Rs 699",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

kidsmain();

break;

case 27:

exit(1);

break;

}

}

void kid\_BODYSUITS(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. CAP\t: Rs 299",

"2. GLOVES\t: Rs 399",

"3. MUFFLERS\t: Rs 349",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

kidsmain();

case 27:

exit(1);

break;

}

}

void kid\_ROMPERS\_SLEEPSUITS(int a,int b)

{

system("cls");

time();

string mn[5]=

{

"PRICE LIST.......",

"1. COTTON HALF T-SHIRT\t: Rs 349",

"2. NYLON HALF T-SHIRT\t: Rs 249",

"3. COTTON FULL T-SHIRT\t: Rs 399",

"4. NYLON HALF T-SHIRT\t: Rs 299"

};

int i=0;

while(i<5)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-2+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

kidsmain();

case 27:

exit(1);

break;

}

}

void kid\_CLOTHINGSET(int a,int b)

{

system("cls");

time();

string mn[5]=

{

"PRICE LIST.......",

"1. COTTON SHORTS\t: Rs 499",

"2. COTTON PANTS\t: Rs 999",

"3. CARGO JEANS\t: Rs 1299",

"4. DENIM JEANS\t: Rs 1499"

};

int i=0;

while(i<5)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-2+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

kidsmain();

case 27:

exit(1);

break;

}

}

**Men.cpp**

#include<bits/stdc++.h>

#include<fstream>

#include<conio.h>

#include"../conio2.h"

#include"../locTime.h"

#define el cout<<endl;

using namespace std;

int men(int a, int b,string mn[]);

void men\_sweater(int a, int b);

void men\_jackets\_blazers(int a,int b);

void men\_CAPS\_GLOVES\_MUFFLERS(int a,int b);

void men\_T\_SHIRTS(int a,int b);

void men\_SHORTS\_PANTS\_JEANS(int a,int b);

void men\_SHIRTS(int a,int b);

void window1();

void menmain()

{

string mn[11]=

{

"WELCOME TO THE MEN'S SECTION",

"What would you like to buy? ",

"### WINTER COLLECTION ###",

"1. SWEATERS",

"2. JACKETS & BLAZERS",

"3. CAPS , GLOVES AND MUFFLERS",

"### SUMMER COLLECTION ###",

"4. T-SHIRTS",

"5. SHORTS, PANTS & JEANS",

"6. SHIRTS",

"0. PREVIOUS MENU",

};

int a=68, b=12;

men(a,b,mn);

}

int men(int a, int b, string mn[])

{

system("cls");

time();

int i=0;

while(i<11)

{

gotoxy(a,b+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+22);

cout<<"ENTER YOUR CHOICE(0 TO 6) : ";

textbackground(WHITE);

cout<<" ";

textbackground(BLACK);

int choice;

gotoxy(a+29,b+22);

cin>>choice;

textbackground(BLACK);

switch(choice)

{

case 1:

men\_sweater(a,b);

break;

case 2:

men\_jackets\_blazers(a,b);

break;

case 3:

men\_CAPS\_GLOVES\_MUFFLERS(a,b);

break;

case 4:

men\_T\_SHIRTS(a,b);

break;

case 5:

men\_SHORTS\_PANTS\_JEANS(a,b);

break;

case 6:

men\_SHIRTS(a,b);

break;

case 0:

window1();

break;

default:

gotoxy(a,b+25);

cout<<"WRONG INPUT!";

}

}

void men\_sweater(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. V NECK\t\t: Rs 699 ",

"2. ROUND NECK\t: Rs 499",

"3. TURTLE NECK\t: Rs 799",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

menmain();

break;

case 27:

exit(1);

break;

}

}

void men\_jackets\_blazers(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. JEANS JACKET\t: Rs 899",

"2. LEATHER JACKET\t: Rs 1499",

"3. HOODIE\t\t: Rs 699",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

menmain();

break;

case 27:

exit(1);

break;

}

}

void men\_CAPS\_GLOVES\_MUFFLERS(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. CAP\t: Rs 299",

"2. GLOVES\t: Rs 399",

"3. MUFFLERS\t: Rs 349",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

menmain();

case 27:

exit(1);

break;

}

}

void men\_T\_SHIRTS(int a,int b)

{

system("cls");

time();

string mn[5]=

{

"PRICE LIST.......",

"1. COTTON HALF T-SHIRT\t: Rs 349",

"2. NYLON HALF T-SHIRT\t: Rs 249",

"3. COTTON FULL T-SHIRT\t: Rs 399",

"4. NYLON HALF T-SHIRT\t: Rs 299"

};

int i=0;

while(i<5)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-2+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

menmain();

case 27:

exit(1);

break;

}

}

void men\_SHORTS\_PANTS\_JEANS(int a,int b)

{

system("cls");

time();

string mn[5]=

{

"PRICE LIST.......",

"1. COTTON SHORTS\t: Rs 499",

"2. COTTON PANTS\t: Rs 999",

"3. CARGO JEANS\t: Rs 1299",

"4. DENIM JEANS\t: Rs 1499"

};

int i=0;

while(i<5)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-2+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

menmain();

case 27:

exit(1);

break;

}

}

void men\_SHIRTS(int a,int b)

{

system("cls");

time();

string mn[5]=

{

"PRICE LIST.......",

"1. COTTON HALF SHIRT\t\t: Rs 549",

"2. COTTON FULL SHIRT\t\t: Rs 699",

"3. NYLON FULL T-SHIRT\t: Rs 449",

"4. NYLON HALF T-SHIRT\t: Rs 599"

};

int i=0;

while(i<5)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-2+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

menmain();

case 27:

exit(1);

break;

}

}

**Women.cpp**

#include<bits/stdc++.h>

#include<fstream>

#include<conio.h>

#include"../conio2.h"

#include"../locTime.h"

#define el cout<<endl;

using namespace std;

int women(int a, int b, string mn[]);

void women\_sweater(int a,int b);

void women\_jackets(int a,int b);

void women\_CAPS\_GLOVES\_MUFFLERS(int a,int b);

void women\_TOPS\_JEANS(int a,int b);

void women\_SAREES(int a,int b);

void women\_SALWARSUITS\_KURTIS(int a,int b);

void women\_WESTERNS(int a,int b);

void window1();

void womenmain()

{

string mn[12]=

{

"WELCOME TO THE WOMEN'S SECTION",

"What would you like to buy? ",

"### WINTER COLLECTION ###",

"1. SWEATERS",

"2. JACKETS",

"3. CAPS , GLOVES AND SHAWL",

"### SUMMER COLLECTION ###",

"4. TOPS & JEANS",

"5. SAREES",

"6. SALWAR SUITS & KURTIS",

"7. WESTERNS",

"0. PREVIOUS MENU",

};

int a=68, b=12;

women(a,b,mn);

}

int women(int a, int b, string mn[])

{

system("cls");

time();

int i=0;

while(i<12)

{

gotoxy(a,b+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+24);

cout<<"ENTER YOUR CHOICE(0 TO 7) : ";

textbackground(WHITE);

cout<<" ";

textbackground(BLACK);

int choice;

gotoxy(a+29,b+24);

cin>>choice;

textbackground(BLACK);

switch(choice)

{

case 1:

women\_sweater(a,b);

break;

case 2:

women\_jackets(a,b);

break;

case 3:

women\_CAPS\_GLOVES\_MUFFLERS(a,b);

break;

case 4:

women\_TOPS\_JEANS(a,b);

break;

case 5:

women\_SAREES(a,b);

break;

case 6:

women\_SALWARSUITS\_KURTIS(a,b);

break;

case 7:

women\_WESTERNS(a,b);

break;

case 0:

window1();

break;

default:

gotoxy(a,b+27);

cout<<"WRONG INPUT!";

}

}

void women\_sweater(int a,int b)

{

system("cls");

time();

string mn[5]=

{

"PRICE LIST.......",

"1. POLO NECK\t: Rs 1099",

"2. CARDIGAN\t: Rs 1499",

"3. WOOLEN PULL-OVER SWEATER\t: Rs 999",

"4. THERMAL\t: Rs 799",

};

int i=0;

while(i<5)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+3+3\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

womenmain();

break;

case 27:

exit(1);

break;

}

}

void women\_jackets(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. JEANS JACKET\t: Rs 999 ",

"2. LEATHER JACKET\t: Rs 1599",

"3. HOODIE\t: Rs 799",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

womenmain();

break;

case 27:

exit(1);

break;

}

}

void women\_CAPS\_GLOVES\_MUFFLERS(int a,int b)

{

system("cls");

time();

string mn[4]=

{

"PRICE LIST.......",

"1. CAP\t: Rs 299",

"2. GLOVES\t: Rs 399",

"3. SHAWLS\t: Rs 1399",

};

int i=0;

while(i<4)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

womenmain();

case 27:

exit(1);

break;

}

}

void women\_TOPS\_JEANS(int a,int b)

{

system("cls");

time();

string mn[5]=

{

"PRICE LIST.......",

"1. COTTON TOP\t: Rs 399",

"2. CROP TOP\t: Rs 299",

"3. FULL SLEEVE TOP\t: Rs 349",

"4. DENIM JEANS\t: Rs 999"

};

int i=0;

while(i<5)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-2+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

womenmain();

case 27:

exit(1);

break;

}

}

void women\_SAREES(int a,int b)

{

system("cls");

time();

string mn[6]=

{

"PRICE LIST.......",

"1. COTTON SAREES\t: Rs 1499",

"2. SYNTHETIC SAREES\t: Rs 999",

"3. SILK SAREES\t: Rs 2999",

"4. DENIM JEANS\t: Rs 1499",

"5. KANJIVARAM SAREES\t: Rs 3999"

};

int i=0;

while(i<6)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-4+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

womenmain();

case 27:

exit(1);

break;

}

}

void women\_SALWARSUITS\_KURTIS(int a,int b)

{

system("cls");

time();

string mn[6]=

{

"PRICE LIST.......",

"1. PATIALA SALWAR SUITS\t: Rs 799",

"2. ANARKALI FROCK\t: Rs 1499",

"3. PALAZZO SALWAR SUITS\t: Rs 949",

"4. LONG STRAIGHT KURTIS\t: Rs 599",

"5. ANARKALI KURTIS\t: Rs 699"

};

int i=0;

while(i<6)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-4+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

womenmain();

case 27:

exit(1);

break;

}

}

void women\_WESTERNS(int a,int b)

{

system("cls");

time();

string mn[6]=

{

"PRICE LIST.......",

"1. ONE PIECE\t: Rs 1299",

"2. DANGRI\t: Rs 1199",

"3. COCKTAIL DRESSES\t: Rs 1499",

"4. WRAP DRESSES\t: Rs 1699",

"5. SLIP DRESSES\t: Rs 2099"

};

int i=0;

while(i<6)

{

gotoxy(a,b+5+2\*i);

textcolor(GREEN);

cout<<mn[i]<<endl;

i++;

}

gotoxy(a,b-4+4\*i);

cout<<"PRESS BACKSPACE TO GO MEN SECTION!";

int choice = getch();

switch(choice)

{

case 8:

womenmain();

case 27:

exit(1);

break;

}

}

**time.cpp**

#include<bits/stdc++.h>

#include<ctime>

#include<conio.h>

#include"../conio2.h"

using namespace std;

void time()

{

time\_t tt;

struct tm \*ti;

time(&tt);

ti = localtime(&tt);

textbackground(GREEN);

textcolor(WHITE);

cout<<" DATE & TIME : "<<asctime(ti);

textbackground(BLACK);

}

**window size and position.cpp**

#include <iostream>

#include <fstream>

#include <windows.h>

#include <cwchar>

#include <ctime>

#include <conio.h>

#include"../conio2.h"

#define \_WIN32\_WINNT 0x0500

void MoveWindowSize(int posx, int posy, int x, int y)

{

RECT rectClient, rectWindow, ConsoleRect;

HWND hWnd = GetConsoleWindow();

GetClientRect(hWnd, &rectClient);

GetWindowRect(hWnd, &rectWindow);

MoveWindow(hWnd, posx, posy, rectClient.right - rectClient.left, rectClient.bottom - rectClient.top, TRUE);

HWND console = GetConsoleWindow();

GetWindowRect(console, &ConsoleRect);

MoveWindow(console, ConsoleRect.left, ConsoleRect.top, x, y, TRUE);

HWND consoleWindow = GetConsoleWindow();

SetWindowLong(consoleWindow, GWL\_STYLE, GetWindowLong(consoleWindow, GWL\_STYLE) & ~WS\_MAXIMIZEBOX & ~WS\_SIZEBOX);

}

**Windowsize.cpp**

#include<bits/stdc++.h>

#include<windows.h>

#define \_WIN32\_WINNT 0x0500

using namespace std;

void window\_size()

{

HWND console = GetConsoleWindow();

RECT ConsoleRect;

GetWindowRect(console, &ConsoleRect);

MoveWindow(console, ConsoleRect.left, ConsoleRect.top, 1000, 1000, TRUE);

HWND consoleWindow = GetConsoleWindow();

SetWindowLong(consoleWindow, GWL\_STYLE, GetWindowLong(consoleWindow, GWL\_STYLE) & ~WS\_MAXIMIZEBOX & ~WS\_SIZEBOX);

}

**conio.c**

/\* A conio implementation for Mingw/Dev-C++.

\*

\* Written by:

\* Hongli Lai <hongli@telekabel.nl>

\* tkorrovi <tkorrovi@altavista.net> on 2002/02/26.

\* Andrew Westcott <ajwestco@users.sourceforge.net>

\* Michal Molhanec <michal@molhanec.net>

\*

\* Offered for use in the public domain without any warranty.

\*

\*/

#include <stdio.h>

#include <stdlib.h>

//#include <unistd.h>

#include <windows.h>

#include <string.h>

#include "../conio2.h"

#ifdef \_\_cplusplus

extern "C" {

#endif

static int \_\_BACKGROUND = BLACK;

static int \_\_FOREGROUND = LIGHTGRAY;

static struct text\_info \_\_text\_info = {

1, 1,

LIGHTGRAY + (BLACK << 4),

LIGHTGRAY + (BLACK << 4),

80, 25

};

static int \_\_CONIO\_TOP = 0;

static int \_\_CONIO\_LEFT = 0;

static void

\_\_fill\_text\_info (void)

{

CONSOLE\_SCREEN\_BUFFER\_INFO info;

GetConsoleScreenBufferInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &info);

\_\_CONIO\_LEFT = info.srWindow.Left;

\_\_CONIO\_TOP = info.srWindow.Top;

\_\_text\_info.curx = info.dwCursorPosition.X - \_\_CONIO\_LEFT + 1;

\_\_text\_info.cury = info.dwCursorPosition.Y - \_\_CONIO\_TOP + 1;

\_\_text\_info.attribute = info.wAttributes;

\_\_text\_info.screenwidth = info.srWindow.Right - info.srWindow.Left + 1;

\_\_text\_info.screenheight = info.srWindow.Bottom - info.srWindow.Top + 1;

}

void

gettextinfo (struct text\_info \* info)

{

\_\_fill\_text\_info();

\*info = \_\_text\_info;

}

void

inittextinfo (void)

{

CONSOLE\_SCREEN\_BUFFER\_INFO info;

GetConsoleScreenBufferInfo (GetStdHandle(STD\_OUTPUT\_HANDLE), &info);

\_\_text\_info.normattr = info.wAttributes;

}

void

clrscr (void)

{

DWORD written;

int i;

\_\_fill\_text\_info();

for (i = \_\_CONIO\_TOP; i < \_\_CONIO\_TOP + \_\_text\_info.screenheight; i++) {

FillConsoleOutputAttribute (GetStdHandle (STD\_OUTPUT\_HANDLE),

\_\_FOREGROUND + (\_\_BACKGROUND << 4), \_\_text\_info.screenwidth,

(COORD) {\_\_CONIO\_LEFT, i},

&written);

FillConsoleOutputCharacter (GetStdHandle(STD\_OUTPUT\_HANDLE), ' ',

\_\_text\_info.screenwidth,

(COORD) {\_\_CONIO\_LEFT, i},

&written);

}

gotoxy (1, 1);

}

void

clreol (void)

{

COORD coord;

DWORD written;

\_\_fill\_text\_info();

coord.X = \_\_CONIO\_LEFT + \_\_text\_info.curx - 1;

coord.Y = \_\_CONIO\_TOP + \_\_text\_info.cury - 1;

FillConsoleOutputAttribute (GetStdHandle (STD\_OUTPUT\_HANDLE),

\_\_FOREGROUND + (\_\_BACKGROUND << 4),

\_\_text\_info.screenwidth - \_\_text\_info.curx + 1, coord, &written);

FillConsoleOutputCharacter (GetStdHandle (STD\_OUTPUT\_HANDLE),

' ', \_\_text\_info.screenwidth - \_\_text\_info.curx + 1, coord, &written);

gotoxy (\_\_text\_info.curx, \_\_text\_info.cury);

}

void

delline (void)

{

COORD coord;

SMALL\_RECT rect;

CHAR\_INFO fillchar;

\_\_fill\_text\_info();

coord.X = \_\_CONIO\_LEFT;

coord.Y = \_\_CONIO\_TOP + \_\_text\_info.cury - 1;

rect.Left = \_\_CONIO\_LEFT;

rect.Top = \_\_CONIO\_TOP + \_\_text\_info.cury;

rect.Right = \_\_CONIO\_LEFT + \_\_text\_info.screenwidth - 1;

rect.Bottom = \_\_CONIO\_TOP + \_\_text\_info.screenheight - 1;

fillchar.Attributes = \_\_FOREGROUND + (\_\_BACKGROUND << 4);

#ifdef UNICODE

fillchar.Char.UnicodeChar = L' ';

ScrollConsoleScreenBufferW (GetStdHandle (STD\_OUTPUT\_HANDLE),

&rect, NULL, coord, &fillchar);

#else

fillchar.Char.AsciiChar = ' ';

ScrollConsoleScreenBufferA (GetStdHandle (STD\_OUTPUT\_HANDLE),

&rect, NULL, coord, &fillchar);

#endif

gotoxy (\_\_text\_info.curx, \_\_text\_info.cury);

}

void

insline (void)

{

COORD coord;

SMALL\_RECT rect;

CHAR\_INFO fillchar;

\_\_fill\_text\_info();

coord.X = \_\_CONIO\_LEFT;

coord.Y = \_\_CONIO\_TOP + \_\_text\_info.cury;

rect.Left = \_\_CONIO\_LEFT;

rect.Top = \_\_CONIO\_TOP + \_\_text\_info.cury - 1;

rect.Right = \_\_CONIO\_LEFT + \_\_text\_info.screenwidth - 1;

rect.Bottom = \_\_CONIO\_TOP + \_\_text\_info.screenheight - 2;

fillchar.Attributes = \_\_FOREGROUND + (\_\_BACKGROUND << 4);

#ifdef UNICODE

fillchar.Char.UnicodeChar = L' ';

ScrollConsoleScreenBufferW (GetStdHandle (STD\_OUTPUT\_HANDLE),

&rect, NULL, coord, &fillchar);

#else

fillchar.Char.AsciiChar = ' ';

ScrollConsoleScreenBufferA (GetStdHandle (STD\_OUTPUT\_HANDLE),

&rect, NULL, coord, &fillchar);

#endif

gotoxy (\_\_text\_info.curx, \_\_text\_info.cury);

}

void

movetext (int left, int top, int right, int bottom, int destleft, int desttop)

{

struct char\_info \* buffer;

buffer = malloc ((right - left + 1) \* (bottom - top + 1) \* sizeof(struct char\_info));

gettext (left, top, right, bottom, buffer);

puttext (destleft, desttop, destleft + right - left, desttop + bottom - top, buffer);

free(buffer);

}

void

\_conio\_gettext (int left, int top, int right, int bottom,

struct char\_info \* buf)

{

int i;

SMALL\_RECT r;

CHAR\_INFO\* buffer;

COORD size;

\_\_fill\_text\_info();

r = (SMALL\_RECT) {\_\_CONIO\_LEFT + left - 1, \_\_CONIO\_TOP + top - 1,

\_\_CONIO\_LEFT + right - 1, \_\_CONIO\_TOP + bottom - 1};

size.X = right - left + 1;

size.Y = bottom - top + 1;

buffer = malloc (size.X \* size.Y \* sizeof(CHAR\_INFO));

ReadConsoleOutput (GetStdHandle (STD\_OUTPUT\_HANDLE),

(PCHAR\_INFO) buffer, size, (COORD) {0, 0}, &r);

for (i = 0; i < size.X \* size.Y; i++)

{

#ifdef UNICODE

buf[i].letter = buffer[i].Char.UnicodeChar;

#else

buf[i].letter = buffer[i].Char.AsciiChar;

#endif

buf[i].attr = buffer[i].Attributes;

}

free (buffer);

}

void

puttext (int left, int top, int right, int bottom, struct char\_info \* buf)

{

int i;

SMALL\_RECT r;

CHAR\_INFO\* buffer;

COORD size;

\_\_fill\_text\_info();

r = (SMALL\_RECT) {\_\_CONIO\_LEFT + left - 1, \_\_CONIO\_TOP + top - 1,

\_\_CONIO\_LEFT + right - 1, \_\_CONIO\_TOP + bottom - 1};

size.X = right - left + 1;

size.Y = bottom - top + 1;

buffer = malloc (size.X \* size.Y \* sizeof(CHAR\_INFO));

for (i = 0; i < size.X \* size.Y; i++)

{

#ifdef UNICODE

buffer[i].Char.UnicodeChar = buf[i].letter;

#else

buffer[i].Char.AsciiChar = buf[i].letter;

#endif

buffer[i].Attributes = buf[i].attr;

}

WriteConsoleOutput (GetStdHandle (STD\_OUTPUT\_HANDLE),

buffer, size, (COORD) {0, 0}, &r);

free (buffer);

}

void

gotoxy(int x, int y)

{

COORD c;

c.X = \_\_CONIO\_LEFT + x - 1;

c.Y = \_\_CONIO\_TOP + y - 1;

SetConsoleCursorPosition (GetStdHandle(STD\_OUTPUT\_HANDLE), c);

}

void

cputsxy (int x, int y, char \* str)

{

gotoxy (x, y);

cputs (str);

}

void

putchxy (int x, int y, char ch)

{

gotoxy (x, y);

putch (ch);

}

void

\_setcursortype (int type)

{

CONSOLE\_CURSOR\_INFO Info;

if (type == 0) {

Info.bVisible = FALSE;

} else {

Info.dwSize = type;

Info.bVisible = TRUE;

}

SetConsoleCursorInfo (GetStdHandle (STD\_OUTPUT\_HANDLE),

&Info);

}

void

textattr (int \_attr)

{

\_\_FOREGROUND = \_attr & 0xF;

\_\_BACKGROUND = \_attr >> 4;

SetConsoleTextAttribute (GetStdHandle(STD\_OUTPUT\_HANDLE), \_attr);

}

void

normvideo (void)

{

textattr (\_\_text\_info.normattr);

}

void

textbackground (int color)

{

\_\_BACKGROUND = color;

SetConsoleTextAttribute (GetStdHandle (STD\_OUTPUT\_HANDLE),

\_\_FOREGROUND + (color << 4));

}

void

textcolor (int color)

{

\_\_FOREGROUND = color;

SetConsoleTextAttribute (GetStdHandle (STD\_OUTPUT\_HANDLE),

color + (\_\_BACKGROUND << 4));

}

int

wherex (void)

{

\_\_fill\_text\_info();

return \_\_text\_info.curx;

}

int

wherey (void)

{

\_\_fill\_text\_info();

return \_\_text\_info.cury;

}

char \*

getpass (const char \* prompt, char \* str)

{

int maxlength = str[0];

int length = 0;

int ch = 0;

int x, y;

cputs(prompt);

\_\_fill\_text\_info();

x = \_\_text\_info.curx;

y = \_\_text\_info.cury;

while (ch != '\r') {

ch = getch();

switch (ch) {

case '\r' : /\* enter \*/

break;

case '\b' : /\* backspace \*/

if (length > 0) putchxy (x + --length, y, ' ');

gotoxy (x + length, y);

break;

default:

if (length < maxlength) {

putchxy (x + length, y, '\*');

str[2 + length++] = ch;

}

}

}

str[1] = length;

str[2 + length] = '\0';

return &str[2];

}

void

highvideo (void)

{

if (\_\_FOREGROUND < DARKGRAY) textcolor(\_\_FOREGROUND + 8);

}

void

lowvideo (void)

{

if (\_\_FOREGROUND > LIGHTGRAY) textcolor(\_\_FOREGROUND - 8);

}

void

delay (int ms)

{

Sleep(ms);

}

void

switchbackground (int color)

{

struct char\_info\* buffer;

int i;

buffer = malloc(\_\_text\_info.screenwidth \* \_\_text\_info.screenheight \*

sizeof(struct char\_info));

\_conio\_gettext(1, 1, \_\_text\_info.screenwidth, \_\_text\_info.screenheight,

buffer);

for (i = 0; i < \_\_text\_info.screenwidth \* \_\_text\_info.screenheight; i++) {

unsigned short attr = buffer[i].attr & 0xF;

buffer[i].attr = (color << 4) | attr;

}

puttext(1, 1, \_\_text\_info.screenwidth, \_\_text\_info.screenheight, buffer);

free(buffer);

}

void

flashbackground (int color, int ms)

{

struct char\_info\* buffer;

buffer = malloc(\_\_text\_info.screenwidth \* \_\_text\_info.screenheight \*

sizeof(struct char\_info));

\_conio\_gettext(1, 1, \_\_text\_info.screenwidth, \_\_text\_info.screenheight,

buffer);

switchbackground(color);

delay(ms);

puttext(1, 1, \_\_text\_info.screenwidth, \_\_text\_info.screenheight, buffer);

free(buffer);

}

void

clearkeybuf (void)

{

while (kbhit()) {

getch();

}

}

#ifdef \_\_cplusplus

}

#endif

**conio2.h**

/\*\* @file conio2.h

\* A conio implementation for Mingw/Dev-C++.

\*

\* Written by:

\* Hongli Lai <hongli@telekabel.nl>

\* tkorrovi <tkorrovi@altavista.net> on 2002/02/26.

\* Andrew Westcott <ajwestco@users.sourceforge.net>

\* Michal Molhanec <michal@molhanec.net>

\*

\* Offered for use in the public domain without any warranty.

\*/

#ifndef \_CONIO2\_H\_

#define \_CONIO2\_H\_

#include <conio.h>

#ifdef UNICODE

#include <windows.h> // we need wchar\_t

#endif

#ifdef \_\_cplusplus

extern "C" {

#endif

/\*\*

\* Colors which you can use in your application.

\*/

typedef enum

{

BLACK, /\*\*< black color \*/

BLUE, /\*\*< blue color \*/

GREEN, /\*\*< green color \*/

CYAN, /\*\*< cyan color \*/

RED, /\*\*< red color \*/

MAGENTA, /\*\*< magenta color \*/

BROWN, /\*\*< brown color \*/

LIGHTGRAY, /\*\*< light gray color \*/

DARKGRAY, /\*\*< dark gray color \*/

LIGHTBLUE, /\*\*< light blue color \*/

LIGHTGREEN, /\*\*< light green color \*/

LIGHTCYAN, /\*\*< light cyan color \*/

LIGHTRED, /\*\*< light red color \*/

LIGHTMAGENTA, /\*\*< light magenta color \*/

YELLOW, /\*\*< yellow color \*/

WHITE /\*\*< white color \*/

} COLORS;

/\*@{\*/

/\*\*

\* This defines enables you to use all MinGW conio.h functions without

\* underscore.

\*/

#define cgets \_cgets

#define cprintf \_cprintf

#define cputs \_cputs

#define cscanf \_cscanf

#ifdef UNICODE

#define cgetws \_cgetws

#define getwch \_getwch

#define getwche \_getwche

#define putwch \_putwch

#define ungetwch \_ungetwch

#define cputws \_cputws

#define cwprintf \_cwprintf

#define cwscanf \_cwscanf

#endif

/\*@}\*/

/\*\*

\* Define alias for \_conio\_gettext.

\* If you want to use gettext function from some other library

\* (e.g. GNU gettext) you have to define \_CONIO\_NO\_GETTEXT\_ so you won't get

\* name conflict.

\*/

#ifndef \_CONIO\_NO\_GETTEXT\_

#define gettext \_conio\_gettext

#endif

#define ScreenClear clrscr

/\*\*

\* @anchor cursortypes

\* @name Cursor types

\* Predefined cursor types. \*/

/\*@{\*/

#define \_NOCURSOR 0 /\*\*< no cursor \*/

#define \_SOLIDCURSOR 100 /\*\*< cursor filling whole cell \*/

#define \_NORMALCURSOR 20 /\*\*< cursor filling 20 percent of cell height \*/

/\*@}\*/

/\*\*

\* Structure holding information about screen.

\* @see gettextinfo

\* @see inittextinfo

\*/

struct text\_info {

unsigned char curx; /\*\*< cursor coordinate x \*/

unsigned char cury; /\*\*< cursor coordinate y \*/

unsigned short attribute; /\*\*< current text attribute \*/

unsigned short normattr; /\*\*< original value of text attribute after

start of the application. If you don't

called the <TT>inittextinfo</TT> on the

beginning of the application, this always

will be black background and light gray

foreground \*/

unsigned char screenwidth; /\*\*< screen width \*/

unsigned char screenheight; /\*\*< screen height \*/

};

/\*\*

\* Structure used by gettext/puttext.

\* @see \_conio\_gettext

\* @see puttext

\*/

struct char\_info {

#ifdef UNICODE

wchar\_t letter; /\*\*< character value \*/

#else

char letter; /\*\*< character value \*/

#endif

unsigned short attr; /\*\*< attribute value \*/

};

/\*\*

\* Returns information of the screen.

\* @see text\_info

\*/

void gettextinfo (struct text\_info \* info);

/\*\*

\* Call this if you need real value of normattr attribute in the text\_info

\* structure.

\* @see text\_info

\*/

void inittextinfo (void);

/\*\*

\* Clears rest of the line from cursor position to the end of line without

\* moving the cursor.

\*/

void clreol (void);

/\*\*

\* Clears whole screen.

\*/

void clrscr (void);

/\*\*

\* Delete the current line (line on which is cursor) and then moves all lines

\* below one line up. Lines below the line are moved one line up.

\*/

void delline (void);

/\*\*

\* Insert blank line at the cursor position.

\* Original content of the line and content of lines below moves one line down.

\* The last line is deleted.

\*/

void insline (void);

/\*\*

\* Gets text from the screen. If you haven't defined <TT>\_CONIO\_NO\_GETTEXT\_</TT>

\* prior to including <TT>conio2.h</TT> you can use this function also under

\* the <TT>gettext</TT> name.

\* @see char\_info

\* @see puttext

\* @param left Left coordinate of the rectangle, inclusive, starting from 1.

\* @param top Top coordinate of the rectangle, inclusive, starting from 1.

\* @param right Right coordinate of the rectangle, inclusive, starting from 1.

\* @param bottom Bottom coordinate of the rectangle, inclusive, starting from 1.

\* @param buf You have to pass buffer of size

\* <TT>(right - left + 1) \* (bottom - top + 1) \* sizeof(char\_info)</TT>.

\*/

void \_conio\_gettext (int left, int top, int right, int bottom,

struct char\_info \* buf);

/\*\*

\* Puts text back to the screen.

\* @see char\_info

\* @see \_conio\_gettext

\* @param left Left coordinate of the rectangle, inclusive, starting from 1.

\* @param top Top coordinate of the rectangle, inclusive, starting from 1.

\* @param right Right coordinate of the rectangle, inclusive, starting from 1.

\* @param bottom Bottom coordinate of the rectangle, inclusive, starting from 1.

\* @param buf You have to pass buffer of size

\* <TT>(right - left + 1) \* (bottom - top + 1) \* sizeof(char\_info)</TT>.

\*/

void puttext (int left, int top, int right, int bottom, struct char\_info \* buf);

/\*\*

\* Copies text.

\* @param left Left coordinate of the rectangle, inclusive, starting from 1.

\* @param top Top coordinate of the rectangle, inclusive, starting from 1.

\* @param right Right coordinate of the rectangle, inclusive, starting from 1.

\* @param bottom Bottom coordinate of the rectangle, inclusive, starting from 1.

\* @param destleft Left coordinate of the destination rectangle.

\* @param desttop Top coordinate of the destination rectangle.

\*/

void movetext (int left, int top, int right, int bottom, int destleft,

int desttop);

/\*\*

\* Moves cursor to the specified position.

\* @param x horizontal position

\* @param y vertical position

\*/

void gotoxy(int x, int y);

/\*\*

\* Puts string at the specified position.

\* @param x horizontal position

\* @param y vertical position

\* @param str string

\*/

void cputsxy (int x, int y, char \* str);

/\*\*

\* Puts char at the specified position.

\* @param x horizontal position

\* @param y vertical position

\* @param ch char

\*/

void putchxy (int x, int y, char ch);

/\*\*

\* Sets the cursor type.

\* @see @ref cursortypes

\* @param type cursor type, under Win32 it is height of the cursor in percents

\*/

void \_setcursortype (int type);

/\*\*

\* Sets attribute of text.

\* @param \_attr new text attribute

\*/

void textattr (int \_attr);

/\*\*

\* Sets text attribute back to value it had after program start.

\* It uses text\_info's normattr value.

\* @see text\_info

\*/

void normvideo (void);

/\*\*

\* Sets text background color.

\* @see COLORS

\* @param color new background color

\*/

void textbackground (int color);

/\*\*

\* Sets text foreground color.

\* @see COLORS

\* @param color new foreground color

\*/

void textcolor (int color);

/\*\*

\* Reads the cursor X position.

\* @returns cursor X position

\*/

int wherex (void);

/\*\*

\* Reads the cursor Y position.

\* @returns cursor Y position

\*/

int wherey (void);

/\*\*

\* Reads password. This function behaves like cgets.

\*

\* @see cgets

\* @param prompt prompt which will be displayed to user

\* @param str string for the password. <TT>str[0]</TT> have to contain

\* length of the <TT>str</TT> - 3

\* @returns <TT>&str[2]</TT>, the password will be stored in <TT>str</TT>

\* beginning at <TT>str[2]</TT>, in <TT>str[1]</TT> will be length of the

\* string without <TT>\\0</TT>, at <TT>str[2 + str[1]]</TT> will be \\0.

\*/

char \* getpass (const char \* prompt, char \* str);

/\*\*

\* Makes foreground colors light.

\* If the current foreground color is less than <TT>DARKGRAY</TT> adds

\* 8 to the its value making dark colors light.

\* @see COLORS

\* @see lowvideo

\*/

void highvideo (void);

/\*\*

\* Makes foreground colors dark.

\* If the current foreground color is higher than <TT>LIGHTGRAY</TT> substracts

\* 8 from its value making light colors dark.

\* @see COLORS

\* @see highvideo

\*/

void lowvideo (void);

/\*@{\*/

/\*

\* You may need to link with libmsvcr70.a or libmsvcr70d.a or libmsvcr71.a

\* or libmsvcr71d.a if you want any of these functions.

\*/

#ifdef UNICODE

\_CRTIMP wchar\_t \* \_\_cdecl \_cgetws(wchar\_t \*);

\_CRTIMP unsigned short \_\_cdecl \_getwch(void);

\_CRTIMP unsigned short \_\_cdecl \_getwche(void);

\_CRTIMP unsigned short \_\_cdecl \_putwch(wchar\_t);

\_CRTIMP unsigned short \_\_cdecl \_ungetwch(unsigned short);

\_CRTIMP int \_\_cdecl \_cputws(const wchar\_t \*);

\_CRTIMP int \_\_cdecl \_cwprintf(const wchar\_t \*, ...);

\_CRTIMP int \_\_cdecl \_cwscanf(const wchar\_t \*, ...);

#endif

/\*@}\*/

/\*\*

\* Pauses program execution for a given time.

\* @see switchbackground

\* @param ms miliseconds

\*/

void delay (int ms);

/\*\*

\* Replaces background color in the whole window. The text however

\* is left intact. Does not modify textbackground().

\* @see flashbackground

\* @param color background color

\*/

void switchbackground (int color);

/\*\*

\* Changes background color for a given time and then it restores it back.

\* You can use it for visual bell. Does not modify textbackground().

\* @see switchbackground

\* @see delay

\* @param color background color

\* @param ms miliseconds

\*/

void flashbackground (int color, int ms);

/\*\*

\* Clears the keyboard buffer.

\* To see it in effect run <TT>conio\_test</TT> and try to press a key during

\* the 'Flashing...' phase.

\*/

void clearkeybuf (void);

#ifdef \_\_cplusplus

}

#endif

#endif /\* \_CONIO2\_H\_ \*/

**kids\_section.h**

#ifndef KIDS\_SECTION\_H\_INCLUDED

#define KIDS\_SECTION\_H\_INCLUDED

void kidsmain();

#endif // KIDS\_SECTION\_H\_INCLUDED

**men\_section.h**

#ifndef MEN\_SECTION\_H\_INCLUDED

#define MEN\_SECTION\_H\_INCLUDED

void menmain();

#endif // MEN\_SECTION\_H\_INCLUDED

**women\_section.h**

#ifndef WOMEN\_SECTION\_H\_INCLUDED

#define WOMEN\_SECTION\_H\_INCLUDED

void womenmain();

#endif // WOMEN\_SECTION\_H\_INCLUDED

**locTime.h**

#ifndef LOCTIME\_H\_INCLUDED

#define LOCTIME\_H\_INCLUDED

void time();

#endif // LOCTIME\_H\_INCLUDED

**window\_size.h**

#ifndef WINDOW\_SIZE\_H\_INCLUDED

#define WINDOW\_SIZE\_H\_INCLUDED

void window\_size();

#endif // WINDOW\_SIZE\_H\_INCLUDED

**windowms.h**

#ifndef WINDOWMS\_H\_INCLUDED

#define WINDOWMS\_H\_INCLUDED

void MoveWindowSize(int posx, int posy, int x, int y);

#endif // WINDOWMS\_H\_INCLUDED

**11.Testing :-**

## Team Interaction

The following describes the level of team interaction necessary to have a successful product.

* The Test Team will work closely with the Development Team to achieve a high quality design and user interface specifications based on customer requirements. The Test Team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.
* The Test Team will work closely with Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.
* Since the application interacts with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

# Test Objective

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. We will be testing a Binary Search Tree Application utilizing a pre-order traversal format. There will be eight key functions used to manage our application: load, store, clear, search, insert, delete, list in ascending order, and list in descending order. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

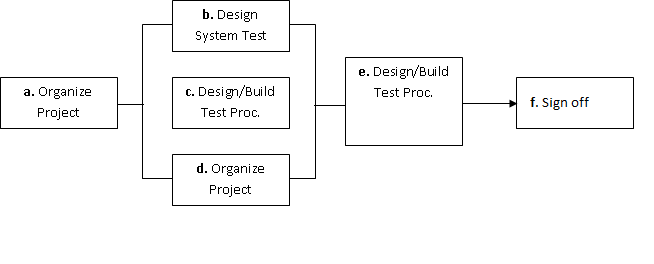
* **Process Overview**

The following represents the overall flow of the testing process:

Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.

1. Identify which particular test(s) will be used to test each module.
2. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
3. Identify the expected results for each test.
4. Document the test case configuration, test data, and expected results.
5. Perform the test(s).
6. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report .
7. Successful unit testing is required before the unit is eligible for component integration/system testing.
8. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
9. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

# Testing Process



The diagram above outlines the Test Process approach that will be followed.

**organize Project** involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.

**a.Design/Build System Test** involves identifying Test Cycles, Test Cases, Entrance & Exit Criteria, Expected Results, etc. In general, test conditions/expected results will be identified by the Test Team in conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are derived from the Program Specifications Document.

**b. Design/Build Test Procedures** includes setting up procedures such as Error Management systems and Status reporting.

**c. Build Test Environment** includes requesting/building hardware, software and data set-ups.

**e. Execute System Tests –** The tests identified in the Design/Build Test Procedures will be executed. All results will be documented and Bug Report Forms filled out and given to the Development Team as necessary.

**f. Signoff** - Signoff happens when all pre-defined exit criteria have been achieved.

# Testing Strategy

The following outlines the types of testing that will be done for unit, integration, and system testing. While it includes what will be tested, the specific use cases that determine how the testing is done will be detailed in the Test Design Document. The test cases that will be used for designing use cases is shown in Figure 2.1 and onwards.

* **Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Somnath Bhadra | |
| **Test Type** | | Unit Testing | |
| **Test Case Number** | | 1 | |
| **Test Case Name** | | User Identification | |
| **Test Case Description** | | The user should enter his/ her accurate user id and password so that he/she can able to go for the further options. The test case will check the application for the same since a user can only login with the correct user id , password. | |
| **Item(s) to be tested** | | | |
| 1 | Verification of the user id and password with the record in the database. | | |
| **Specifications** | | | |
| **Input** | | | **Expected**  **Output/Result** |
| 1. Correct User id and password 2. Incorrect Id or Password | | | 1. Successful login 2. Failure Message |

## Unit Testing :-

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

### White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

* **Black Box Testing**

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

## System Testing :-

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases we will use the black-box method of testing.

12.Cost Estimation of the Project along with Cost Estimation Model :-

Analogous estimate of effort or cost

Used for Early Estimate or Individual Activity Estimate

Sample example shown below is for two major deliverables of a software project. You use a previous project as a benchmark for analogous estimation. Using your experience you will estimate a multiplier.

* **Multipliers:**

1. Prototyping: 0.75.
2. Testing: 0.5
3. Deployment: 0.5

Finally, if you want to convert to cost, you would use current rates for the resource.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **WBS ID** | **Previous**  **Similar Project**  **Activity** | **Previous**  **Effort** | **Current**  **Project**  **Estimate** | **Multiplier** | **Effort**  **(Previous Effort \* 0.75)** | **Cost**  **(Rs. 500/hr.)** |
| 1 | Prototyping | 40 Work-Hours | Prototyping | 0.75 | 30 Work-hours | Rs. 15000/- |
| 2 | Testing | 20 Work-Hours | Testing | 0.50 | 10 Work-Hours | Rs. 5000/- |
| **Total** |  |  |  |  | **40 Work- Hours** | **Rs. 20000/-** |

Note: Effort is also called Size and unit of estimation is called either Work-Hour, person-hours.

13. Future scope and further enhancement of the Project :-

In the future, shoppers will likely increasingly rely on **a mix of brick-and-mortar and digital channels for discovery, researching and purchasing**. Whether in-store or online, brands can meet shoppers' needs by using data and machine learning to understand behaviors and deliver personalized, seamless experiences.

14. Bibliography :-

* Object Oriented Programming With C++, E Balagurusamy, TMH
* Object Oriented Programming in Turbo C++, Robert Lafore, Galgotia
* The Compete Reference C++, Herbert Schlitz, TMH
* C++ programming in easy steps