## MODERN PERIODIC TABLE

A Dissertation submitted in partial fulfillment of the requirements for the award of degree of

# MASTER OF COMPUTER APPLICATIONS

By

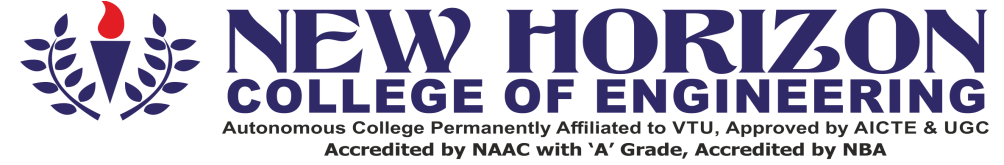
**AKASH PRADHAN**

**1NH18MCA03**

Under the Guidance of

**Dr. V. ASHA**

**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

****

Ring Road, Near Marathalli,

Bangalore – 560 103

**2018-2019**

## MODERN PERIODIC TABLE

A Dissertation submitted in partial fulfillment of the requirements for the award of degree of

# MASTER OF COMPUTER APPLICATIONS

By

**AKASH PRADHAN**

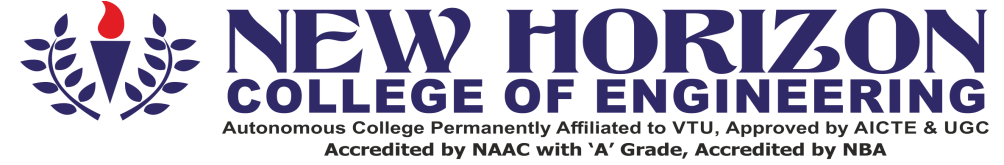
**1NH18MCA03**

Under the Guidance of

**Dr. V. ASHA**

Professor

**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

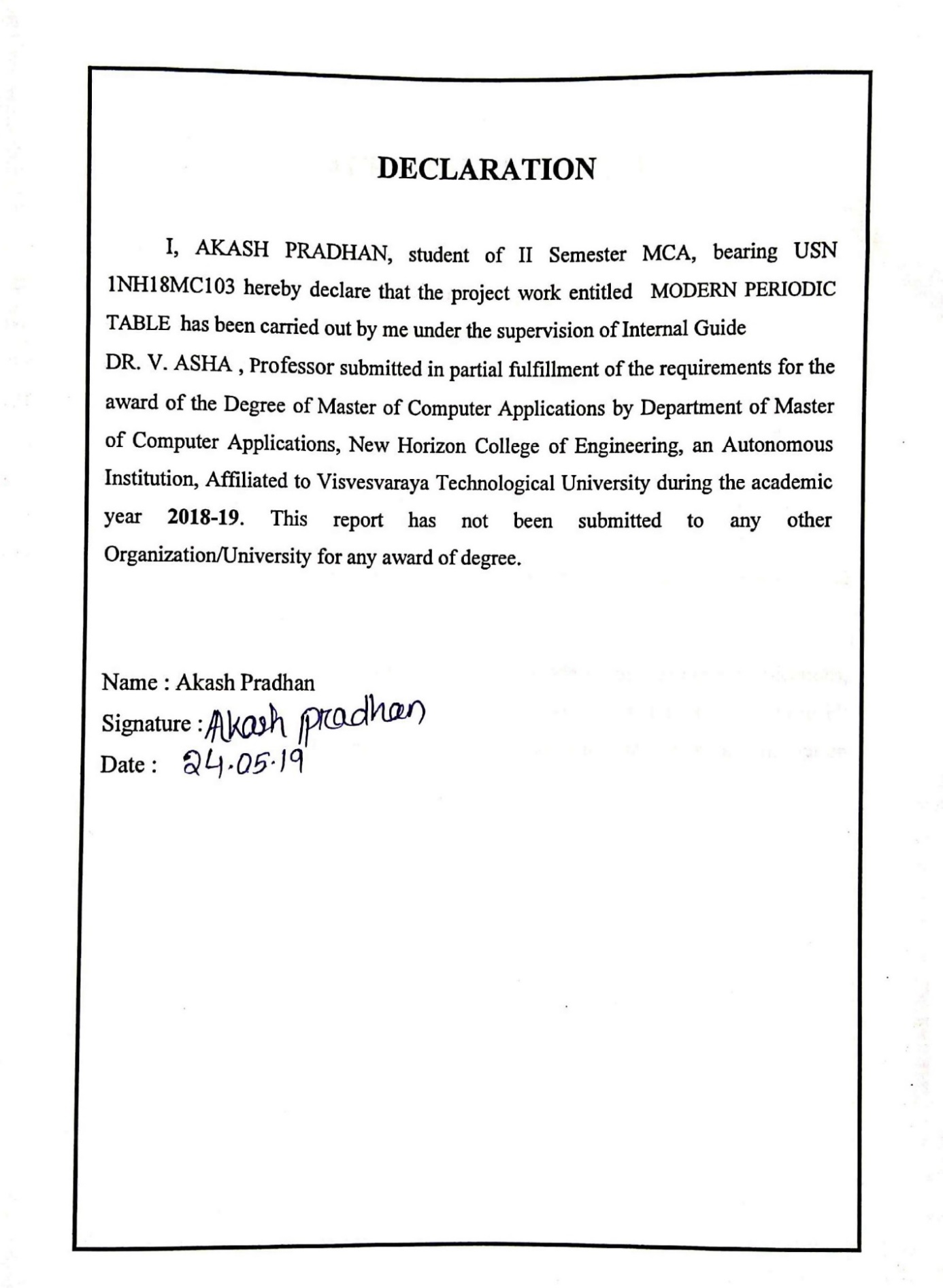
****

Ring Road, Near Marathalli,

Bangalore – 560 103

**2018-2019**





**ACKNOWLEDGEMENT**

I would like to thank **Dr. Mohan Manghnani ,** Chairman of New Horizon College of Engineering for providing good infrastructure and Hi-Tech lab facilities to develop and improve student’s skills.

I sincerely express my gratitude to the college Principal **Dr. Manjunatha** for supporting the students in all their technical activities and giving guidance to them. I would like to thank **Dr. V. Asha**, HoD, Department of MCA, New Horizon College of Engineering for granting permission to undertake this project. I would like to express my gratitude to the project guide Prof. **Dr. V. Asha** for giving all the instructions and guidelines at every stage of the Project work.

I thank all the staff members of the Department of Master of Computer Applications, for extending their constant support to complete the project. I express my heartfelt thanks to my parents and friends who were a constant source of support and inspiration throughout the project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter**  **No.** | **Title** | **Page No** |
|  | **ABSTRACT**  **LIST OF TABLES**  **LIST OF FIGURES** | (i)  (ii)  (iii) |
| **1**  1.1  1.2  1.3  1.4  1.5  1.6 | **INTRODUCTION**  General Introduction  Problem Statement  Existing System  Objective of the Work  Proposed System with Methodology  Feasibility Study | 1-4 |
| **2**  2.1 | **REVIEW OF LITERATURE**  Review Summary | 5-7 |
| **3**  3.1  3.2 | **SYSTEM CONFIGURATION**  Hardware requirements  Software requirements | 8 |
| **4**  4.1  4.2 | **MODULE DESCRIPTION**  Module 1  Module 2 | 9-10 |
| **5**  5.1  5.2 | **SYSTEM DESIGN**  DFD / UML Diagrams  Data Base Design | 11 |
| **6**  6.1  6.1.1  6.1.2  6.2 | **SYSTEM IMPLEMENTATION**  Implementation  Source Code  Screen Shots | 12-29 |
| **7**  7.1 | **SYSTEM TESTING**  Test Cases | 28 |
| **8**  8.1  8.2  8.3 | **RESULTS AND DISCUSSIONS**  Conclusion  Limitations  Future Enhancements | 29 |
| **9**  9.1  9.2 | **REFERENCES**  Text Reference  Web Reference | 30 |

**ABSTRACT**

The Modern Periodic Table project is a simple console application built in without the use of graphics. It is developed using the C programming language for the purpose of storing name, symbol, atomic number, atomic weight, and some important properties ,as well as to display them as per requirement of the user.

This project will help you to understand file handling in C i.e. creating a file and accessing the stored data in the file, modifying and removing the stored data. It will also help you to understand the use of functions as well as different parameters of C programming language.

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Table No.** | **Title** | **Page No.** |
| 1 | 1.1 | Element list | 5 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Figure No.** | **Title** | **Page No.** |
| 1 | 1.1 | Atomic Structure | 2 |
| 2 | 2.1 | Simple Periodic Table | 6 |
| 3 | 5.1 | DFD Main Function | 12 |
| 4 | 5.2 | DFD Explore Function | 13 |

**CHAPTER 1**

**INTRODUCTION**

**1.1** **General Introduction**

We know about the periodic table we studied in chemistry. The periodic table not easy to remember all symbol with atomic number. So i made a project about modern periodic table which is may helpful for student.

Student can easily find the name, symbol, atomic number, atomic weight, and some important properties in one click.

 It is developed using the C programming language for the purpose of storing name, symbol, atomic number, atomic weight, and some important properties as well as to display them as per requirement of the user.

The source code for this project is complete and totally error free. The C source code is simple and understandable, and consists of around 550 lines of code.

In this project we can create a file using file pointer ,we can not only accessing the stored data in file but also modifying and remove the stored data from the file.

**1.2 Problem Statement**

In the periodic table Each element name is represented by an element symbol, it could be first letter of the name of the element in English, Latin, or German. For example

* C for carbon
* K for *ka lium* (potassium in Latin)

The capitalized first letter followed by another lowercase letter from the name of the element in English or Latin. For example

* Si for silicon

a systematic element name and symbol — essentially a placeholder.

In the table there is two number are usually added to each cell.

6

**C**

12.011

**12**

12.011

**atomic number**

**atomic mass**

Fig 1.1 Atomic Structure

This process the symbol and atomic mass are difficult to find and remember also.

* 1. **Existing System**

Modern periodic table is a innovative thought to simplify the how to search a certain data using name, symbol, atomic number etc.

In this present system the user first insert some data and after that that will be store in file, after that these data are fetch from file give output result in your screen.

Following details are maintain by the user

* Insert data with name, atomic number, atomic mass, symbol
* Explore data that means it will fetch from file
  1. **Objective of the Work**

This project is to create to search the element periodic table easily. To search for the element by periodic name you have type **Element Name Atomic Number**. Also, as mentioned you can search the elements by their atomic number. After you search the element it will show you the results by displaying their details.

You can search an element by using any of the following method:

1. By name of element
2. By symbol of element
3. By atomic number of element
4. By atomic weight of element
   1. **Proposed System with Methodology**

In the previous system, details are store Manually. To share the details between the file and explore function.

Functional and advantage

* Data is Centralized which has overcome the shearing problem
* As data is maintained electronically its’s easy for update
* Maintenance is easy and performance is good
* Mainly the system has automated

All details are given by the user which is need name, symbol, atomic number and atomic mass. All the details are store in database.

Whenever the user needs to search the data just type its name, symbol, number etc.

* 1. **Feasibility Report**

Preliminary investigation examine project feasibility, the likelihood the system useful for the student. The main objective of the feasibility study is to test the technical operation and adding the new data module and debugging old paper periodic table.

In the basic design of this project structured array is used, the cost required to make the project is negligible. No major training and new skills are required to use this project. The project is time saving and processes fast and gives results as per user’s data. Improvised performance compared to manual systems.

**CHAPTER 2**

**REVIEW OF LITERATURE**

In 1864 a German chemist Julius Lothar Meyer organized the element by atomic mass and grouped them according to their chemical properties.

Later that decade a Russian chemist Dimitri Mendeleev organized all the known elements according to similar properties. He left gaps in his table for what he thought were undiscovered elements, and he made some bold predictions regarding the properties of those undiscovered elements. When elements were later discovered whose properties closely matched Mendeleev’s predictions .

Because certain properties of the elements repeat on a regular basis throughout the table (that is, they are periodic), it became known as the periodic table.

Mendeleev, who first published his periodic table in 1869, is usually credited with the origin of the modern periodic table.

### Features of the Periodic Table

Elements that have similar chemical properties are grouped in columns called groups.As well as being numbered, some of these groups have names for example, alkali metals(the first column of elements), alkaline earth metals (the second column of elements).Each row of elements on the periodic table is called a **period**. Periods have different lengths, the first period has only 2 elements (hydrogen and helium), while the second and third periods have 8 elements each. The fourth and fifth periods have 18 elements each, and later periods are so long that a segment from each is removed and placed beneath the main body of the table.

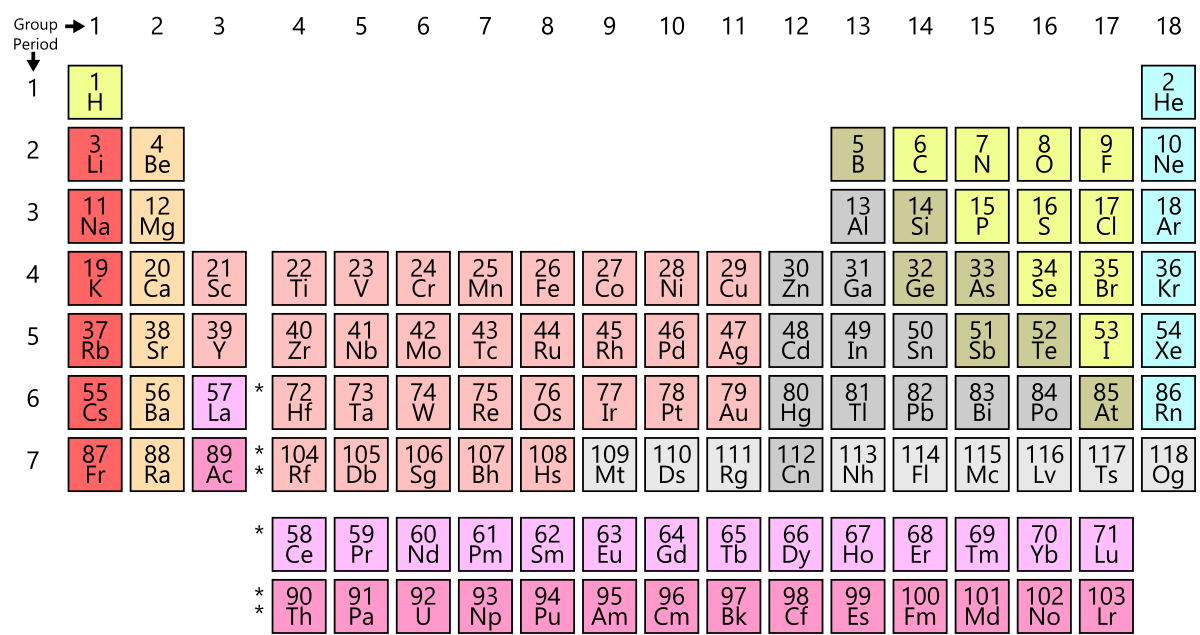


Fig 2.1 Simple Periodic Table

* The periodic table is an arrangement of the elements in order of increasing atomic number.
* Elements that exhibit similar chemistry appear in vertical columns called groups (numbered 1–18 from left to right); the seven horizontal rows are called periods.
* Some of the groups have widely-used common names, including the alkali metals (Group 1) and the alkaline earth metals (Group 2) on the far left, and the halogens (Group 17) and the noble gases (Group 18) on the far right.
* The elements can be broadly divided into metals, nonmetals, and metalloids.
* Semi metals exhibit properties intermediate between those of metals and nonmetals.
* Metals are located on the left of the periodic table, and nonmetals are located on the upper right. They are separated by a diagonal band of metalloids.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Symbol** | **Atomic number** | **Atomic mass** | **Block** | **properties** |
| [Hydrogen](https://byjus.com/chemistry/hydrogen/) | H | 1 | 1.0079 | 1 | gas |
| [Helium](https://byjus.com/chemistry/helium/) | He | 2 | 4.0026 | 18 | gas |
| [Lithium](https://byjus.com/chemistry/lithium/) | Li | 3 | 6.9411 | 1 | solid |
| [Beryllium](https://byjus.com/chemistry/beryllium/) | Be | 4 | 9.0122 | 2 | solid |
| [Boron](https://byjus.com/chemistry/boron/) | B | 5 | 10.811 | 13 | solid |
| [Carbon](https://byjus.com/chemistry/carbon1/) | C | 6 | 12.0107 | 14 | Solid |
| [Nitrogen](https://byjus.com/chemistry/nitrogen/) | N | 7 | 14.0067 | 15 | gas |

Table 1.1 Element list Table

**CHAPTER 3**

**SYSTEM CONFIGURATION**

**3.1 Hardware Requirements**

For this project it needs minimum system hardware requirement for run this program properly.

* RAM: 256 MB or more
* HARD DISK 40 GB or more
* Processor P3 or High
* Compiler Standard C Compiler

**3.2 Software Requirements**

* Operating System : Microsoft Windows 7 Ultimate
* Language Used : C programming
* Front End : C++ Software
* Back End : None

Developer++ or TurboC++ compileris a [free](https://en.wikipedia.org/wiki/Free_software) full-featured [Integrated Development Environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) distributed under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) for programming in [c](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). It is written in [Delphi](https://en.wikipedia.org/wiki/Delphi_(programming_language)). It is bundled with, and uses, the [MinGW](https://en.wikipedia.org/wiki/MinGW" \o "MinGW) or [TDM-GCC](https://en.wikipedia.org/wiki/TDM-GCC) 64bit port of the [GCC](https://en.wikipedia.org/wiki/GNU_Compiler_Collection) as its compiler. Dev-C++ can also be used in combination with [Cygwin](https://en.wikipedia.org/wiki/Cygwin) or any other GCC-based compiler. Dev-C++ is generally considered a [Windows-only](https://en.wikipedia.org/wiki/Microsoft_Windows) program, but there are attempts to create a Linux version: header files and path delimiters are switchable between platforms.

**CHAPTER 4**

**MODULE DESCRIPTION**

**4.1 Storage of Element Information**

In the project, you can add any new element with its name, symbol, atomic number, atomic weight and its some important properties. When new element information is to be added to this Modern Periodic Table, you have to enter 1 in the main menu and input information in given format. This information is stored in file created on the hard disk of computer by program itself.

After press 1 you see this type of user input screen where you have to input this information like Name, Symbol, Atomic No, Atomic Weight, etc.

After successfully entered all information it will store in a file for further process.

In this below function user can add the data

* **void add()** 
  + This function is used to input or add the information of new element to the program.
  + By this function user can add the name, symbol, atomic number etc.

**4.2 Exploration of Element Information**

Another main function of this project is to explore or to display the stored information. You can search an element by using any of the following method .

* By name of element
* By symbol of element
* By atomic number of elements
* By atomic weight of element

I am using below main function to designing my project

* **void explore()**
  + This function is used to explore the stored information in the file created.
  + In this function first a prompt will show that is user can search the an item by their name, symbol, properties, atomic number etc.
  + That is the stored information or data are display with there all properties.
* **void mainscreen()**
* It is included in source code of project file in order to print the text style and to control its color.
* It implement for design the project and for attractive.
* **Void print()**
  + This function is used to print the main screen or menu of the project.
* **Int main()**
* In main function used for called the user define function and print error message if user select wrong choice.

**PERIODIC TABLE**

**Enter corresponding number**

**1.Search by Name**

**2.Search by Symbol**

**3.Search by Atomic Number**

**4.Search by Atomic weight**

**5.Element from different block**

**6.Return to main menu**

Fig 4.1 main menu

In this function a user can search the data by using the Name, Symbol, Atomic Number, Atomic Weight, and by block.

The 6th option is for return to the main menu which means if user don’t want to search then it will go back to back.

**CHAPTER 5**

**SYSTEM DESIGN**

**LEVEL 1**

ADD Details

EXPLORE Details

Exit

Fig 5.1 DFD Main Function

**LEVEL 1.1**

EXPLORE

Fig 5.2 explore function

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 Implementation**

The project was implemented by c programming language using data structure. In this project my aim is to store data in a file and fetch these data from file when needed. Before implement the project we have to download Code::Block software to develop the program.

**6.2 Source Code**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#include <windows.h>

#include<iostream>

using namespace std;

struct element{

char name[20];

char sb[5];

int atm;

float atms;

char block;

char atc[20];

char prop[250];

}p,q;

int rw,cl;

FILE\*fp;

void add();

void explor();

void print();

void mainscreen();

void gotoxy(int,int);

void textcolor(int);

int main()

{

int a,i,n,y;

char c,d;

char date1[15],date2[15],string1[20];

unsigned int tsz;

mainscreen();

label1:

gotoxy(25,9);

textcolor(14);

cprintf("ENTER THE CORRESSPONDIND NO ->");

gotoxy(25,9);

textcolor(11);

cprintf("\n------------------------------\n");

gotoxy(22,11);textcolor(10);

cprintf("1.Add new Element Information");

gotoxy(22,13);

cprintf("2.Explore");

gotoxy(22,15);

cprintf("3.Quit");

gotoxy(22,17);

fflush(stdin); /\*Used to clear the buffer and accept next string \*/

d=getch();

switch(d)

{

case '1': /\*add item \*/

{

add();

break;

}

case '2': /\* Explor \*/

{

explor();

break;

}

case '3': /\* Quit \*/

{

//clrscr()();

mainscreen();

textcolor(14);

gotoxy(30,20);

cprintf("THANK U.....");

gotoxy(30,22);

cprintf("BYE...........");

getch();

exit(1);

break;

}

default:

{

//clrscr()();

mainscreen();

textcolor(12+128);

gotoxy(29,12);

cprintf("Wrong choice");

gotoxy(29,13);textcolor(11);

cprintf("Retype choice");

goto label1;

}

}

mainscreen();

goto label1;

return 0;

}

void add()

{

char ch;

label1:

mainscreen();

gotoxy(15,8);textcolor(11);

cprintf("Enter the Information of Elements:"); gotoxy(14,9);

cprintf("==================================");

getch();

gotoxy(15,10);textcolor(10);

cprintf("Name:");

gotoxy(15,12);

cprintf("Symbol:");

gotoxy(15,14);

cprintf("Atomic No: ");

gotoxy(15,16);

cprintf("Atomic Wt: ");

gotoxy(15,18);

cprintf("Atomic Config:");

gotoxy(15,20);

cprintf("Block:");

gotoxy(15,22);

cprintf("Properties:");

fflush(stdin);

gotoxy(23,12);

scanf(" %c",&p.sb);

fflush(stdin);

gotoxy(25,14);

scanf(" %d",&p.atm);

fflush(stdin);

gotoxy(21,20);

scanf(" %c",&p.block);

if(p.block!='S'&&p.block!='P'&&p.block!='D'&&p.block!='F')

p.block=' ';

fflush(stdin);

gotoxy(26,22);

scanf(" %[^\n]",p.prop);

if((fp=fopen("data","ab+"))==NULL)

{

printf("Cannot open the file");

getch();

exit(1);

}

fwrite(&p,sizeof(p),1,fp);

fclose(fp);

printf("\n\n\n\t\tEnter 'y' for next record(y/n):");

ch=getch();

if(ch=='y')

{

goto label1;

}

}

void explor()

{

char d,c;

FILE \*f;

int given\_atmic\_no,a,i,tsz,n;

float given\_atmic\_mass;

int flag;

char string[20];

startofexplore:

switch(d)

{

case '1':

{

//clrscr()();

mainscreen();

gotoxy(15,20);

textcolor(12);

cprintf("Enter the Name of Element:");

textcolor(3);

fflush(stdin);

scanf("%[^\n]",string);

printf("%s",string);

fclose(fp);

getch();

break;

}

case '6':

{

return;

}

case '3':

{

mainscreen();

gotoxy(15,20);

textcolor(12);

cprintf("Enter the Atomic No. Element:");

textcolor(3);

fflush(stdin);

scanf("%d",&given\_atmic\_no);

if((fp=fopen("data","rb+"))==NULL)

{

printf("\n cannot open the record file 1");

getch();

exit(1);

}

flag=1;

while(fread(&p,sizeof(p),1,fp))

{

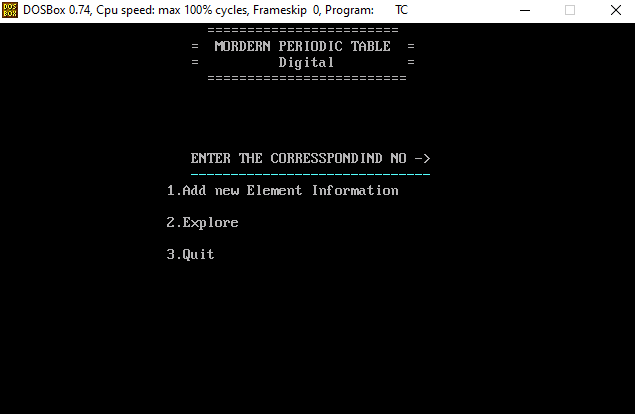
if(p.atm==given\_atmic\_no)

print();

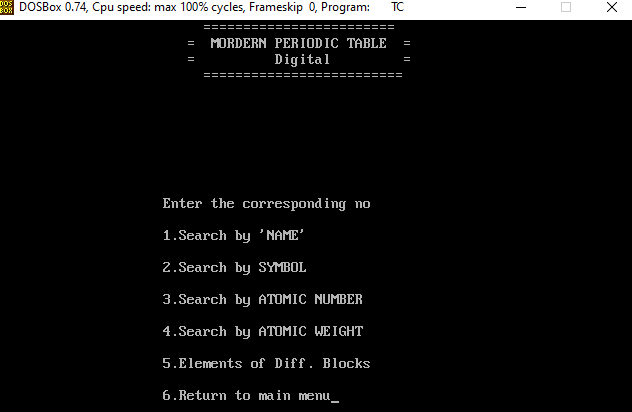
flag=0;

}

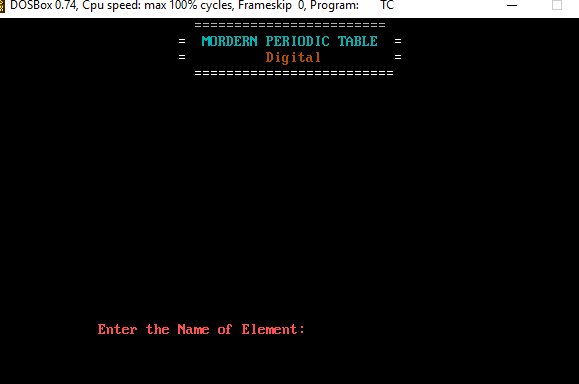
**6.3 Screenshot**



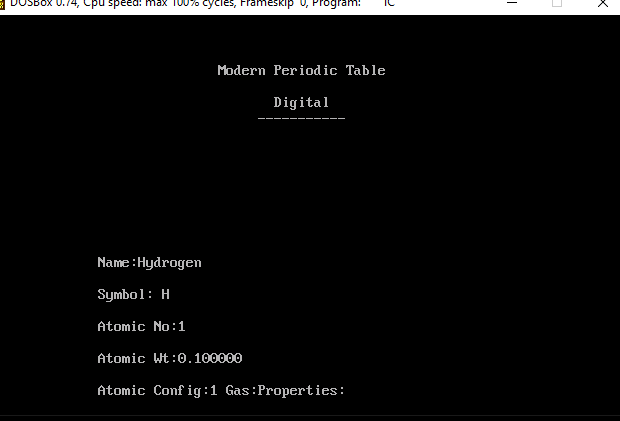
In the above screen shot it shows the main screen when you run the program. It will ask some condition statement that are for add the data into the file and second one is the explore the data that is describe below in another screen shot . And we have last option that is quite which will break this program and return to the program.



The above screenshot is about explore function .The explore function ask some option to input data from the user and search by what by you want to search like if you want to search by name then write the symbol name after that you get output as per the that symbol.



In above diagram shown that to Enter the name of Element Which is entered by the user. Whenever user Want to search an element by the name this function will run background and it will show as a output.



**CHAPTER 7**

**SYSTEM TESTING**

**7.1 Test Cases**

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

Types of Testing

**1.Unit Testing**

It focuses on smallest unit of software design. In this we test an individual unit or group of interrelated units. It is often done by programmer by using sample input and observing its corresponding outputs.

#### **2. Integration Testing**

#### The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components are combined to produce output.

#### Integration testing are of two types

(i)Top down

(ii)Bottom up

**(a) Black Box Testing:** It is used for validation. In this we ignore internal working mechanism and focus on what is the output?

**(b) White Box Testing:** It is used for verification. In this we focus om internal mechanism i.e. how the output is achieved?

#### **3.** **Regression Testing:** Every time new module is added leads to changes in program. This type of testing makes sure that whole component works properly even after adding components to the complete program.

#### **4. Alpha Testing:** This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.

#### **5. Beta Testing:** The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment

#### **6. System Testing:** In this software is tested such that it works fine for different operating system. It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working. In this we have security testing, recovery testing, stress testing and performance testing.

#### **7. Stress Testing:** In this we give unfavorable conditions to the system and check how they perform in those condition.

#### **8. Performance Testing:** It is designed to test the run-time performance of software within the context of an integrated system. It is used to test speed and effectiveness of program.

**7.2 Maintenance**

IEEE Standard 1219 defines this as

“The modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment”

Maintenance has been an important concern for software since several years and is a problem for the future.

Different types of software maintenance

**a) Corrective Maintenance**

Removing defects in programs

**b) Adaptive Maintenance**

Modifying or changing the application for a new set of features (hardware, OS, programming language etc…)

**c) Perfective Maintenance**

Improving the performance by paying stress on quality parameters

**d) Preventive Maintenance**

Detecting faults much ahead of time

**CHAPTER 8**

**RESULTS AND DISCUSSIONS**

**8.1 Conclusion**

This C mini project on Modern Periodic Table for college mini project for use of functions and file handling in C. The project formulates a clear concept of application of C programming language.

This project will help to understand file handling in C i.e. creating a file and accessing the stored data in the file, modifying and removing the stored data. It will also help you to understand the use of functions as well as different parameters of C programming language.

**8.2 Limitations**

Though this project gives everything whatever is required, then also it has its limitations:

1. The project does not contain any database, hence cannot save the data permanently.

2. Each time the code gets executed, it starts from initial stage, that means all the entries filled in the previous execution are erased automatically

**CHAPTER 9**

**REFERENCES**

**9.1 TEXT REFERENCES**

1. Programming in ANSI C: Balaguruswamy, 6th Edition.
2. Roger S. Pressman : Software Engineering – A Practitioners approach,7th Edition, Mc-Graw Hill Education, 2012.
3. Petrucci, Ralph H., William S. Harwood, F. G. Herring, and Jeffrey D. Madura. General Chemistry: Principles and Modern Applications. 9th ed.Upper Saddle River: Pearson Education, Inc., 2007.
4. Sisler, Harry H. Electronic structure, properties, and the periodic law. New york; Reinhold publishing corporation, 1963.

**8.2 WEB REFERENCE**

1. <https://www.sigmaaldrich.com/technical-documents/articles/biology/periodic-table-of-elements-names.html>
2. <https://en.wikipedia.org/wiki/List_of_chemical_elements>
3. <https://www.tutorialspoint.com/software_engineering/software_testing_overview.htm>
4. <https://study.com/academy/lesson/periodic-table-projects-for-middle-school.html>

