**DECISION MAKER BASH SCRIPT**

##### A PROJECT REPORT

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

##### ANJALI SINGH

##### AMINA SHAHEEN

***for the award of the degree***

***of***

##### *BACHELOR OF TECHNOLOGY*

**IN**

COMPUTER SCIENCE AND ENGINEERING



**INTEGRAL UNIVERSITY LUCKNOW**

JULY 2020

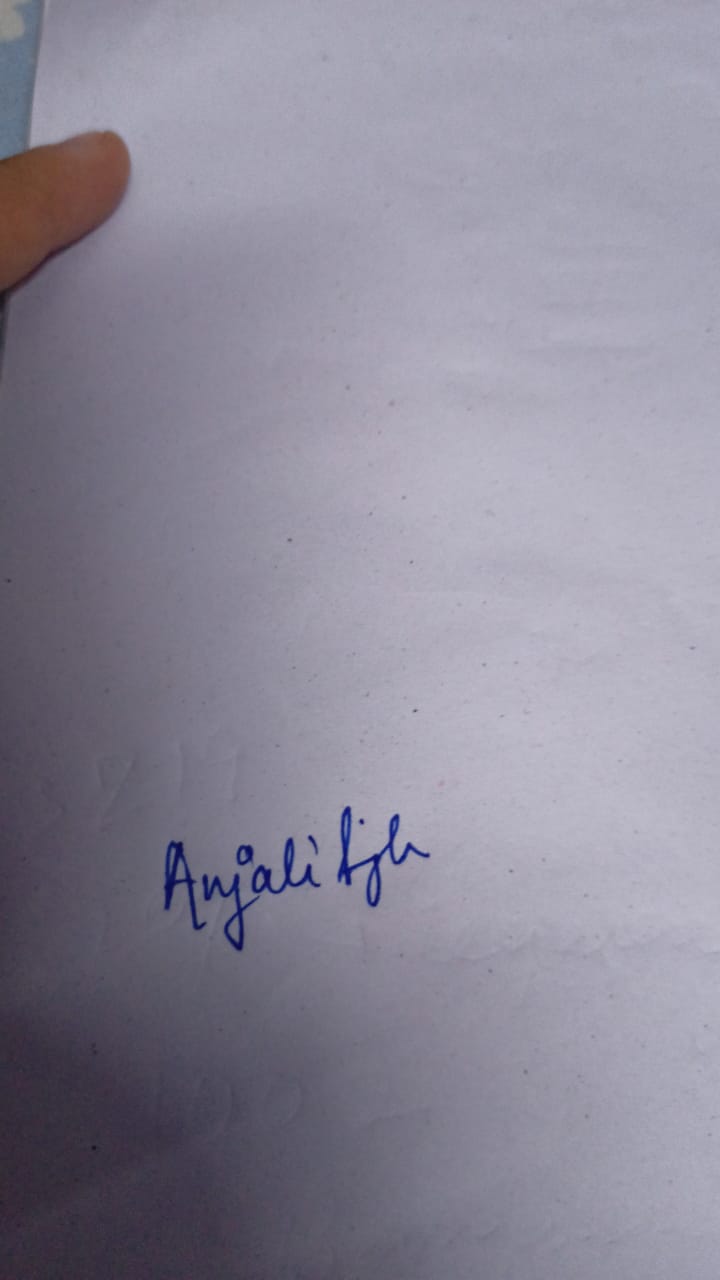
**INTEGRAL UNIVERSITY: LUCKNOW**

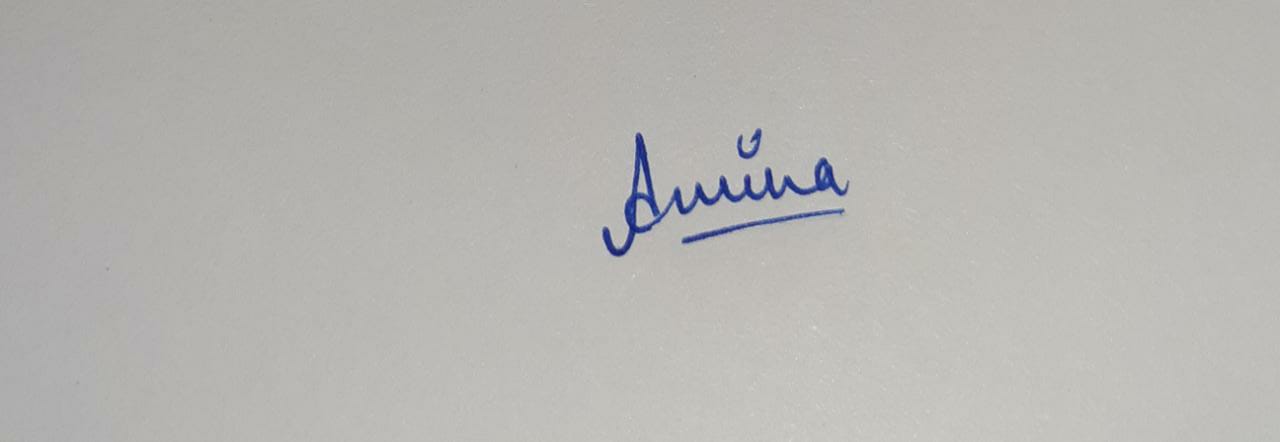
**UNDERTAKING**

We, Anjali Singh and Amina Shaheen, students of B.Tech. (CSE/IT), hereby declare that the work detailed in this Project entitled **“DECISION MAKER BASH SCRIPT”** submitted to the Department of Computer Science and Engineering, Integral University, Lucknow for the award of the Bachelor of Technology degree is our original work. We have neither plagiarized nor submitted this work for the award of any other degree. In case, this undertaking is found incorrect, We accept that my degree may be unconditionally withdrawn.

Date:

Venue: Lucknow

ANJALI SINGH 

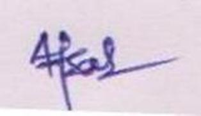
AMINA SHAHEEN 

**INTEGRAL UNIVERSITY: LUCKNOW**

**CERTIFICATE**

This is to certify that the work contained in this project report entitled **“ DECISION MAKING BASH SCRIPT”** by

**Anjali Singh and Amina Shaheen** is a faithful record of work that has been carried out by the students, under my supervision and the level of work is good for submission. To the best of our knowledge, this work has not been submitted for award of any degree or diploma to this University or elsewhere.



Mr. Afsaruddin,

Assistant Professor

Dept. of CSE

Integral University, Lucknow

**INTEGRAL UNIVERSITY: LUCKNOW**

**RECOMMENDATION**

On the basis of the declaration submitted by **ANJALI SINGH** and **AMINA SHAHEEN,** internal assessment carried out by department on date 28/07/2020 and the certificate issued by the Guide **Mr. AFSARUDDIN,** the work entitled **“ DECISION MAKER BASH SCRIPT”** submitted to the department of CSE, is recommended for final examination.

Signature B.Tech Project Coordinator

Name & Designation Signature of HOD

Name & Designation

Date:\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

With a deep sense of gratitude, we wish to express our sincere thanks to our guide, MR**. AFSARUDDIN**, Computer Science and Engineering Department for giving us the opportunity to work under him on the project.

We truly appreciate and value his esteemed guidance and encouragement from the beginning to end of this project. We are extremely grateful to him.

We want to thank all the teachers for providing a solid background for our studies and research thereafter. They have been a great source of inspiration to us and we thank them from the bottom of our heart.

We also want to thank our parents, who taught us the value of hard work by their own example. They rendered us enormous support during the whole tenure of our stay in Integral University, Lucknow. Finally, we would like to thank our department for giving us the opportunity and platform to make our effort a successful one.

**ANJALI SINGH (1601012028)**

**AMINA SHAHEEN (1601012022)**

**ABSTRACT**

DECISION MAKER BASH SCRIPT is a route finding bash script to find the shortest possible route between two points and calculate the fare according to the route chosen for a journey. The data is fed related to route information and using ETL,DBMS and C++ based script and the best route hence relevant cost is found.

A Bash script is a plain text file which contains a series of commands. A bash is a type of interpreter that processes shell commands. A shell interpreter takes commands in plain text format and calls Operating system services to do something.

In this Decision maker bash script, first we have raw data. The raw data is extracted using ETL(Extract,Transform, Load). Data extraction involves extracting data from homogeneous or heterogeneous sources;and transformation processes data by data cleansing and transforming them 9into a proper storage format/structure for the purpose of querying and analysis;finally, data loading describes the insertion of data into the final target database such as an operational data store, a data mart, data lake or a data warehouse.

The extracted data is now converted into a C++ based script. This script is then loaded into the database and the required information is extracted.

This data is then converted into a data structure and finally this is processed to get the final result.

**TABLE OF CONTENTS**

**CHAPTER TITLE PAGE NO.**

1. Cover Page & Title Page 1

2. Declaration by the group members 2

3. Certificate 3

4. Recommendation 4

5. Acknowledgement 5

6. Abstract 6

7. Table of content 7-8

8. List of Figure 9

9. List of Table 10

10. List of Symbols, Abbreviations and Nomenclature 11

. 11. Chapters

11.1 Introduction 12

11.2 Problem Identification & Feasibility Study 13-17

11.3 Requirement Analysis 18-20

11.4 Review of Previous work 21

11.5 Proposed Work (Framework or Algorithm) 22-33

11.6 Hardware & Software Specification 34-39

11.7 Design 40

Context Diagram 40

Data Flow Diagrams 41-42

Entity Relationship Diagrams 43

Flowchart 43

Snapshot of the Project 44-50

11.8. Result & Snapshot 51-52

11.9 Conclusion 53

12. References 54-58

13. Bio data of each group member 59-63

**LIST OF FIGURES**

**FIG NO. TITLE PAGE NO.**

1. ETL 13

2. Dynamic Programming of Graphs 26

3. Agile Model in SDLC 37

4. Context Diagram for Decision Maker Script 40

5. Level-1 DFD 41

6. Level-2 DFD 42

7. Entity Relationship Diagram for the Database 43

8. Database Connectivity 44-45

9. Database Structure 46-50

10. Result & Snapshot 51-52

**LIST OF TABLES**

**DATA TABLES:**

CITY\_MASTER

|  |  |  |
| --- | --- | --- |
| Name Column | Data Type | Constraints |
| city\_id | int(20) | Primary key |
| city\_name | Varchar(40) | NULL |

FARE\_LIST

|  |  |  |
| --- | --- | --- |
| Name Column | Data Type | Constraints |
| city\_id | int(20) | Foreign Key |
| connected\_city\_id | int(20) | Primary Key |
| fare | int (20) | NULL |

**LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE**

* ETL - Extract Transform And Load.
* JSON - JavaScript Object Notation.

**INTRODUCTION**

A Bash Script is a plain text file which contains a series of commands. These commands are a mixture of commands we would normally type ourselves on the command line(such as **ls** or **cp** for example).

Bash is a [command processor](https://en.wikipedia.org/wiki/Command-line_interpreter) that typically runs in a [text window](https://en.wikipedia.org/wiki/Terminal_emulator) where the user types commands that cause actions. Bash can also read and execute commands from a file, called a [shell script](https://en.wikipedia.org/wiki/Shell_script). It is a type of interpreter that processes shell commands. A shell interpreter takes commands. A shell scripting is writing a program for a shell to execute and a shell script is a file or program that shell will execute. We use these commands in a terminal which is the interface to the shell interpreter.

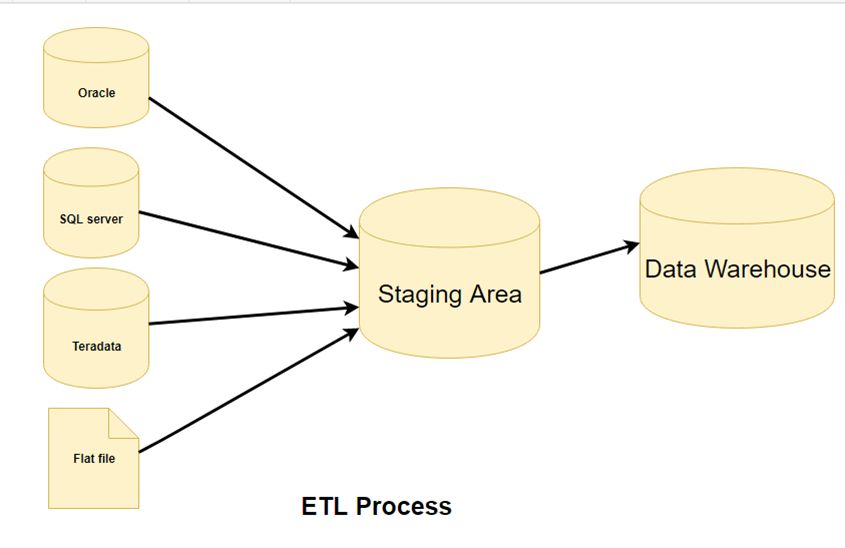
A shell is a fully-fledged programming language in itself. It can define variables, functions and we can do conditional execution of shell commands as well. The typical operations performed by shell scripts include file manipulation, program execution and printing text. A Shell provides you with an interface to the Unix system. It gathers input from you and executes programs based on that input. When a program finishes executing, it displays that program's output.

The Decision maker bash script is a C++ based bash script which uses shell scripting to execute the problem of finding the best path with minimum cost between two points. We have our information stored in JSON text format. JSON is a text format that is completely language independent. Information in JSON files is stored in the form of key-pair. With file-handling we extract the data from a JSON file using jsoncpp library and then convert it into a script. Now with mysql connectivity it is loaded into the database and the required information is extracted. The extracted data is converted into a graph and the required result is generated.

**PROBLEM IDENTIFICATION AND FEASIBILITY STUDY**

**PROBLEM DESCRIPTION**

In the project titled **“DECISION MAKER BASH SCRIPT”** we are trying to extract data from a JSON file and then make a C++ based Bash Script which we transform into a data structure such as graph, through which we calculate minimum cost between two vertices in the graph. This can be done using the Shortest Path Algorithm.



**FEASIBILITY STUDY**

Once a preliminary area of script has been identified, it is then subjected to a more rigorous examination in a feasibility study. This type of study determines, if a project can and should be taken. The contents and recommendations of such a study will be used to select the best system that meets performance requirements and objectives.

In order to test the feasibility of the proposed system **“DECISION MAKER BASH SCRIPT”** the following activities were studied in detail to select the best system that meets the user’s requirement.

1. Characteristics of the system.
2. Objectives & requirements of the system.
3. Feasibility study for user and system management

The important tests of feasibility are studied and described below:-

**1.** Technical feasibility

**2.** Economical feasibility

**3.** Operational feasibility

Feasibility Study is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

1. What are the user’s demonstrable needs and how does a candidate system meet them.

2. What resources are available for a given candidate system?

3. What are the likely impacts of the candidate system on the organization?

4. Is it worth it to solve the problem?

During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this.

**Technical feasibility:**

This evaluation determines whether the technology needed for the proposed system is available or not.

* Can the work for the project be done with current equipment existing software technology & available personnel?
* Can the system be upgraded if developed?
* If new technology is needed then what can be developed?

This is concerned with specifying equipment and software that will successfully satisfy the user requirement.

An important issue for the development of a project is the selection of suitable front-end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

**The aspects of our study included the following factors:**

**Front-end selection:**

1. It must have a graphical user interface that assists employees that are not from IT background.

2. Scalability and extensibility.

3. Flexibility.

4. Robustness.

5. According to the user requirement.

6. Platform independent.

7. Easy to debug and maintain.

8. Event driven programming facility.

**Back-end Selection:**

1. Easy to install.

2. Popularity.

3. Operating System compatible.

4. Various drivers must be available.

5. Easy to implant with the Front-end.

According to above stated features we selected MySQL as the backend.

The technical feasibility is frequently the most difficult area encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed system.

It is more efficient, easy to understand and has a user-friendly GUI (Graphical User Interface) which can be understood by everyone. Hence this project has good technical feasibility.

**Economic feasibility:**

Economic justification is generally the “Bottom Line” consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weigh the cost and the benefits associated with the candidate system and if it is cost-efficient then the project is undertaken. The financial and the economic questions during the preliminary investigation are verified to estimate the following:

1. The cost to conduct a full system investigation.
2. The cost of hardware and software for the class of application being considered.
3. The benefits in the form of reduced cost.

The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.

1. This feasibility checks whether the system can be developed with the available funds.
2. The system does not require an enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible. The cost of the project depends upon the number of man-hours required.

Economic feasibility is mainly concerned with the cost incurred in the implementation of the software. Since this project is developed using **C++** which is more commonly available and the cost of installation is not very high. The proposed system is efficient, reliable and quickly responding.

**Operational Feasibility:**

It is mainly related to human organizations and political aspects. The points to be considered are:

What changes will be brought with the system?

What new skills will be required? Do the existing members have these skills? If not, can they be trained in due course of time?

The system is operationally feasible as it is very easy for the end users to operate it. It only needs basic information about Linux platforms.

**Schedule feasibility:**

Time evaluation is the most important consideration in the development of a project. The time schedule required for the development of this project is very important since more development time affects machine time, cost and cause delay in the development of other systems.

**REQUIREMENT ANALYSIS**

This section contains all the software requirements at a level of detail, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements

These include two types of Requirements i.e. Functional and Non Function requirements:

1. **Functional Requirements**

Functional requirements explain what has to be done by identifying the necessary task, action or activity that must be accomplished. Functional requirements analysis will be used as the top-level functions for functional analysis. The project is developed using C++, Data Structure, Bash Scripting and Mysql 5.5.

1. **Non-Functional Requirements**

In system engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements defining specific behavior or function. Non-functional requirements define the need in terms of performance, design constraints, and other requirements like security, maintainability, portability. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture.

**Performance Requirements**

Performance requirements define acceptable response times for system functionality. The load time for user interface screens will depend on the speed of the internet used by the user who is accessing the script. The performance will also depend upon the Operating System used(here Linux) and its versions.

**Other Requirements**

Other requirements in the system include requirements such as security, maintainability, portability.

**Security**

Linux based systems are highly secured by default. Encrypted volumes are created for sensitive files. A host integrity testing facility is provided to verify that the running system has not been compromised or tampered. You may also easily restore program files for all of the software that is included with your distribution with the software management tools. No one except the person who has been granted permission can access the data.

**Maintainability**

Online Survey web App is developed on C++, Bash Scripting and Mysql 5.5 which will be easy to extend in future.

# Portability

The project will run properly on any Linux based system.

# Accessibility

The accessibility of the script will be operating system dependent. Every file and directory on a LINUX-style system is marked with three sets of file permissions that determine how it may be accessed, and by whom:

* The permissions for the *owner*, the specific account that is responsible for the file.
* The permissions for the *group* that may use the file.
* The permissions that apply to all *other* accounts.

Each set may have none or more of the following permissions on the item:

* *read*
* *write*
* *execute*

# Effectiveness

The Script will be effective enough as it will give very good performance in relation to the effort.

**REVIEW OF PREVIOUS WORK**

The Decision maker bash script consists of following modules:

**JSON PARSER**

It takes raw data in form of a JSON file as an input and then populates the database.

**GRAPH CREATOR**

It creates a graph according to the tables in the database. The output from the first module JSON PARSER is taken as an input after which a Graph as an output is generated.

**FARE CALCULATOR**

It calculates the minimum fare between two cities using the shortest path algorithm. After having the graph data structure it generates the best path between source and destination.

**PROPOSED WORK**

**Purpose**

The purpose of this document is to specify requirements and to give guidelines for the development of above said project. In particular it gives guidelines on how to prepare the above said project.

This document is intended to be a practical guide for people who are developing this software.

**Scope**

The scope of this paper is to create a bash script which can be used by any application developer to calculate the shortest path using basic data structure in a dynamic environment. This problem has many useful applications. The routing protocols used in most of today’s computer networks are based on shortest path algorithms. Machine learning on graphs is a young but growing field.

Neural networks are a way to create functions that [no human could write](https://arxiv.org/pdf/1812.04948.pdf). They do this by harnessing the power of large datasets. On problems for which we have capable neural models, we can use example inputs and outputs to train the network to learn a function that transforms those inputs into those outputs, and hopefully generalizes to other unseen inputs. We need to be able to build neural networks that can learn functions on graphs. Those neural networks need the right [inductive biases](https://arxiv.org/pdf/1806.01261.pdf) so that they can reliably learn useful graph functions. With that foundation, we can build powerful neural graph systems.

**C++ CONCEPTS**

This project is a C++ based bash script. The major purpose of C++ programming is to introduce the concept of object orientation to the C programming language.

Object Oriented Programming is a paradigm that provides many concepts such as **inheritance, data binding, polymorphism, etc.** Object-oriented programming – As the name suggests uses [objects](https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/#objects) in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

**[Class](https://www.geeksforgeeks.org/c-classes-and-objects/)**: The building block of C++ that leads to Object-Oriented programming is a Class. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

We can say that a **Class in C++** is a blue-print representing a group of objects which shares some common properties and behaviours.

**[Object:](https://www.geeksforgeeks.org/c-classes-and-objects/)** An Object is an identifiable entity with some characteristics and behaviour. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

**[Encapsulation](https://www.geeksforgeeks.org/encapsulation-in-c/)**: In normal terms, Encapsulation is defined as wrapping up of data and information under a single unit. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them.

**[Abstraction](https://www.geeksforgeeks.org/abstraction-in-c/)**: Data abstraction is one of the most essential and important features of object-oriented programming in C++. Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

**[Polymorphism:](https://www.geeksforgeeks.org/polymorphism-in-c/)** The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

C++ supports operator overloading and function overloading.

* *Operator Overloading*: The process of making an operator to exhibit different behaviours in different instances is known as operator overloading.
* *Function Overloading*: Function overloading is using a single function name to perform different types of tasks.  
  Polymorphism is extensively used in implementing inheritance.

**[Inheritance](https://www.geeksforgeeks.org/inheritance-in-c/)**: The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programming.

* **Sub Class**: The class that inherits properties from another class is called Subclass or Derived Class.
* **Super Class**:The class whose properties are inherited by subclass is called Base Class or Super class.
* **Reusability**: Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

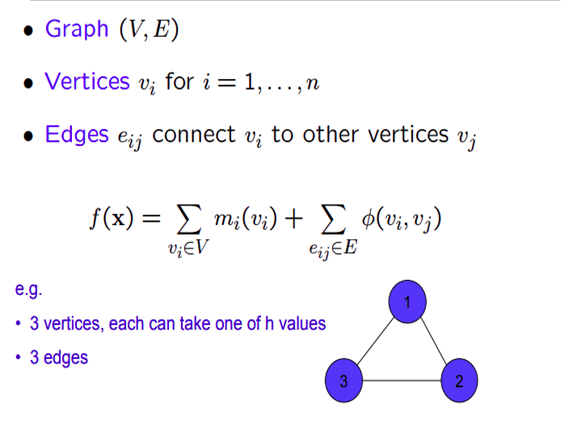
**DYNAMIC PROGRAMMING**

Dynamic programming approach is similar to divide and conquer in breaking down the problem into smaller and yet smaller possible sub-problems. But unlike, divide and conquer, these sub-problems are not solved independently. Rather, results of these smaller subproblems are remembered and used for similar or overlapping subproblems.Dynamic programming is used where we have problems, which can be divided into similar sub-problems, so that their results can be reused. Mostly, these algorithms are used for optimization. Before solving the in-hand sub-problem, a dynamic algorithm will try to examine the results of the previously solved sub-problems. The solutions of sub-problems are combined in order to achieve the best solution.

So we can say that −

* The problem should be able to be divided into smaller overlapping sub-problem.
* An optimum solution can be achieved by using an optimum solution of smaller sub-problems.
* Dynamic algorithms use Memoization.

Here we have used the concepts of dynamic programming to calculate the minimum fare between source and destination. Dynamic programming (usually referred to as DP ) is a very powerful technique to solve a particular class of problems. It demands very elegant formulation of the approach and simple thinking and the coding part is very easy. Dynamic Programming is mainly an optimization over plain [recursion](https://www.geeksforgeeks.org/recursion/). Wherever we see a recursive solution that has repeated calls for the same inputs, we can optimize it using Dynamic Programming. The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial. Several classes of graph optimization problems, which can be solved using dynamic programming, are known to have more efficient tailor-made algorithms. There are two key attributes that a problem must have in order for dynamic programming to be applicable: [optimal substructure](https://en.wikipedia.org/wiki/Optimal_substructure) and [overlapping sub-problems](https://en.wikipedia.org/wiki/Overlapping_subproblem). If a problem can be solved by combining optimal solutions to *non-overlapping* sub-problems, the strategy is called "[divide and conquer](https://en.wikipedia.org/wiki/Divide_and_conquer_algorithm)" instead. This is why [merge sort](https://en.wikipedia.org/wiki/Mergesort) and [quick sort](https://en.wikipedia.org/wiki/Quicksort) are not classified as dynamic programming problems.

 Fig: Dynamic Programming on Graphs

**DATABASE CONNECTIVITY**

MySQL Connector/C++ 1.1 is a MySQL database connector for C++ applications that connect to MySQL servers. Connector/C++ enables development of C++ applications that use the JDBC-based API. Connector/C++ is compatible with the JDBC 4.0 API. Connector/C++ does not implement the entire JDBC 4.0 API, but does feature these classes: Connection, DatabaseMetaData, Driver, PreparedStatement, ResultSet, ResultSetMetaData, Savepoint, Statement.

The JDBC 4.0 API defines approximately 450 methods for the classes just mentioned. Connector/C++ implements approximately 80% of these.

To establish a connection to the MySQL server, retrieve an instance of sql::Connection from a sql::mysql::MySQL\_Driver object. A sql::mysql::MySQL\_Driver object is returned by sql::mysql::get\_mysql\_driver\_instance().These methods can be used to check the connection state or reconnect:

* sql::Connection::isValid() checks whether the connection is alive
* sql::Connection::reconnect() reconnects if the connection has gone down.

**TESTING**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding. The purpose of product testing is to verify and validate the various work products viz. units, integrated unit, final product to ensure that they meet their requirements.

In software development phase and error can be injected at any stage during development, so, the different type of testing is required on different levels. Software Testing is the process of testing the software project. Effective software will contribute to the delivery of higher quality software products, more satisfied users, lower maintenance costs, more accurate, and reliable results.

The main types of software testing are:

**Component Testing**

Starting from the bottom the first test level is “Component Testing”, sometimes called Unit Testing. It involves checking that each feature specified in the “Component Design” has been implemented in the component. In theory an independent tester should do this, but in practice the developer usually does it, as they are the only people who understand how a component works. The problem with a component is that it performs only a small part of the functionality of a system, and it relies on cooperating with other parts of the system, which may not have been built yet. To overcome this, the developer either builds, or uses special software to trick the component into believing it is working in a fully functional system.

* **Interface Testing**

As the components are constructed and tested they are then linked together to check if they work with each other. It is fact that two components that have passed all their tests, when connected to each other produce one new component full of faults. These tests can be done by specialists, or by the developers. Interface testing is not focused on what the components are doing but on how they communicate with each other, as specified in the “System Design”. This involves stating:

1). what a component can expect from another component in terms of services.

2). How these services will be asked for.

3). How they will be given.

4). How to handle non-standard conditions, i.e. errors.

5). Tests are constructed to deal with each of these.

* **System Testing**

Once the entire system has been built then it has to be tested against the “System Specification” to check if it delivers the features required. It is still developer focused, although specialist developers known as system testers are normally employed to do it.

In essence System testing is not about checking the individual parts of the design, but about checking the system as a whole. In effect it is one giant component.

System testing can involve a number of specialist types of test to see if all the functional and non-functional requirements have been met.

* **Acceptance Testing**

Acceptance testing checks the system against the “Requirements”. It is similar to system testing in that the whole system is checked but the important difference is the change in focus. System testing checks that the system that was specified has been delivered. Acceptance testing checks that the system delivers what was requested. The customer and not the developer should always do acceptance testing. The customer knows what is required from the system to achieve value in the business and is the only person qualified to make that judgment. The forms of tests may follow those in system testing, but at all times they are informed by the business needs.

**Test Case Design**

Test case design focuses on a set of techniques for the creation of test cases that meet overall testing objectives. In the test case design phase, the engineer creates a series of test cases that are intended to “demolish” the software that has been built.

Any software product can be tested in one of two ways:

**Black box testing**:-

Black box testing is designed to uncover errors. They are used to demonstrate that software functions are operations; that input is properly accepted and output is correctly produced; and that integrity of external information is maintained. A black box examines some fundamental aspects of a system with little regard for the internal logical structure of the software.

**White box testing:-**

White box testing of software is predicated on close examination of procedural details. Providing test cases that exercises specific sets of conditions and/or loops tests logical paths through the software. The “state of program” may be examined at various points to determine if the expected or asserted status corresponds to the actual status.

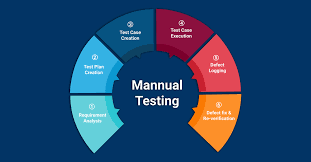
**MANUAL TESTING**

The testing method used to test this project is Manual Testing. Manual Testing is a process in which we compare the behaviour of a developed piece of code(software, module, API, features, etc.) against the expected behaviour(Requirements).

Manual Testing is a process of finding out the defects, bugs in a software program. A tester performs an end user role and verifies if all the features are working properly or not. Tester manually executes the test cases.

Manual Testing does not require knowledge of any testing tool.

One of the Software Testing Fundamentals is "**100% Automation is not possible**".This makes Manual Testing imperative.Test Suites or cases are designed during the testing phase and should have 100% test coverage.It also makes sure that reported defects are fixed by developers and re-testing has been performed by testers on the fixed defects. Basically, this testing checks the quality of the system and delivers bug-free products to the customer.



**Testing the Training Information System**

In case of this software, testing has been done in the following areas and manner:-

1. **Functional Testing**

According to the need of the problem, following testing plans have been planned on some amount on the test data. A document of prerequisites and main sequence of running scripts are created. An execution sequence is developed based on common usage scenarios. Comments and instructions are provided in each shell script. An initial backup is made to create a baseline.

1. **Security Testing**

Linux based systems are highly secured by default. Encrypted volumes are created for sensitive files. A host integrity testing facility is provided to verify that the running system has not been compromised or tampered. You may also easily restore program files for all of the software that is included with your distribution with the software management tools. No one except the person who has been granted permission can access the data.

1. **Performance Testing**

Based on the field conditions these testing for fine tuning can be carried out at a later date.

* Peak load testing
* Storage testing
* Performance time testing
* Recovery testing

1. **Integration Testing**

After testing each component separately, all the web pages and components were put together and tested thoroughly. New errors that were identified were removed. The consistency of database entries was checked and appropriate changes were made. After this testing, the project was submitted to the invigilator for approval.

**Goals of proposed system**

1. **Reliability:** - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.
2. **No Redundancy: -** In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.
3. **Immediate retrieval of information: -** The main objective of the proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.
4. **Immediate storage of information: -** In manual systems there are many problems to store the largest amount of information.
5. **Easy to Operate: -** The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

**HARDWARE AND SOFTWARE SPECIFICATION**

**Software Requirements:-**

* **C++**: It is a general purpose programming language created by Bjarne Stroutstrup as an extension of the C programming language, or "C with [Classes](https://en.wikipedia.org/wiki/Class_(programming))". The language has expanded significantly over time, and modern C++ now has object-oriented ,generic and functional features in addition to facilities for [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) [memory](https://en.wikipedia.org/wiki/Memory_(computing)) manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers including the Free SoftwareFoundation, LLVM, Microsoft,Intel,Oracle and IBM and so on it is available on many platforms.C++ was designed with a bias toward system programming and embedded, resource-constrained software and large systems, with performance, efficiency, and flexibility of use as its design highlights
* **Connector/C++**: MySQL Connector/C++ is a library for applications written in C or C++ that communicate with MySQL database servers. Version 8.0 of Connector/C++ implements three different APIs which can be used by applications.The new XDevAPI for applications written in C++
* **MySQL:** MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for mySQL however, is for the purpose of a web database.

**Hardware Requirements**

* Pentium IV Processor
* 1GB RAM
* 10 GB HDD
* Color Monitor
* Keyboard, Mouse

**Operating System**

Linux-Unix operating system.

**SYSTEM DEVELOPMENT LIFE CYCLE**

The **System development life cycle (SDLC)**, or **Software development process** in systems engineering, [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), is a process of creating or altering information systems, and the models and [methodologies](http://en.wikipedia.org/wiki/Methodologies) that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies).

Broadly, following are the different activities to be considered while defining system [software development process](http://en.wikipedia.org/wiki/Software_development_process) the system development life cycle for the said project:

* Problem Definition
* System Analysis
* Study of existing system
* Drawback of the existing system
* Proposed system
* System Requirement study
* Data flow analysis
* Feasibility study
* System design
* Input Design (Database & Forms)
* Updating
* Query /Report design
* Administration
* Testing
* Implementation
* Maintenance

**SYSTEM ANALYSIS:-** System Analysis is the study of sets of [interacting](http://en.wikipedia.org/wiki/Interaction) [entities](http://en.wikipedia.org/wiki/Entity), including computer systems analysis. This field is closely related to [requirements analysis](http://en.wikipedia.org/wiki/Requirement_analysis) or [operations research](http://en.wikipedia.org/wiki/Operations_research). It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made.

System development can generally be thought of having two major components: systems analysis and systems design. In System Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvement to the system.

**SYSTEM DESIGN:-** Systems design is the process of defining the architecture, components, modules, interfaces, and [data](http://en.wikipedia.org/wiki/Data) for a [system](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement). One could see it as the application of [systems theory](http://en.wikipedia.org/wiki/Systems_theory) to [product development](http://en.wikipedia.org/wiki/Product_development). There is some overlap with the disciplines of [systems analysis](http://en.wikipedia.org/wiki/Systems_analysis), [systems architecture](http://en.wikipedia.org/wiki/Systems_architecture) and [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering). If the broader topic of [product development](http://en.wikipedia.org/wiki/Product_development) "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing [systems](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement) of the user. Until the 1990s systems design had a crucial and respected role in the [data processing](http://en.wikipedia.org/wiki/Data_processing) industry. In the 1990s [standardization](http://en.wikipedia.org/wiki/Standardization) of hardware and software resulted in the ability to build [modular](http://en.wikipedia.org/wiki/Modularity_(programming)) systems. The increasing importance of software running on generic platforms has enhanced the discipline of [software engineering](http://en.wikipedia.org/wiki/Software_engineering).

[Object-oriented analysis and design](http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) methods are becoming the most widely used methods for computer systems design. The [UML](http://en.wikipedia.org/wiki/Unified_Modeling_Language) has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non software systems and organization.

**AGILE PROCESS MODEL**

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software products. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like −

* Planning
* Requirements Analysis
* Design
* Coding
* Unit Testing and
* Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

This project “Decision maker bash script” is based on the Agile process model.The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning.

The time to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change. Though the development team can decide to reduce the delivered functionality during a Time-box if necessary to deliver it on time. The central principle of the Agile model is the delivery of an increment to the customer after each Time-box.Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.



Fig: Agile Model in SDLC

**Principles of Agile model:**

* To establish close contact with the customer during development and to gain a clear understanding of various requirements, each Agile project usually includes a customer representative on the team. At the end of each iteration stakeholders and the customer representative review the progress made and re-evaluate the requirements.
* The Agile model relies on working software deployment rather than comprehensive documentation.
* Frequent delivery of incremental versions of the software to the customer representative in intervals of a few weeks.
* Requirement change requests from the customer are encouraged and efficiently incorporated.
* It emphasizes on having efficient team members and enhancing communications among them is given more importance. It is realized that enhanced communication among the development team members can be achieved through face-to-face communication rather than through the exchange of formal documents.
* It is recommended that the development team size should be kept small (5 to 9 peoples) to help the team members meaningfully engage in face-to-face communication and have a collaborative work environment.
* Agile development processes usually deploy Pair Programming. In Pair programming, two programmers work together at one work-station. One does coding while the other reviews the code as it is typed in. The two programmers switch their roles every hour or so.



FIG: AGILE MODEL WORKING

In this project, the project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirement. Each iteration involves a team working through a full software development life cycle including planning, requirement analysis, design, coding and testing before a working product is delivered.

**PHASES:**

Following are the phases in the Agile Model upon which our project is based:

* Requirement gathering
* Requirement designing
* Construction/Iteration
* Testing/Quality Assurance
* Deployment
* Feedback

**DESIGN**

**Context Diagrams:**

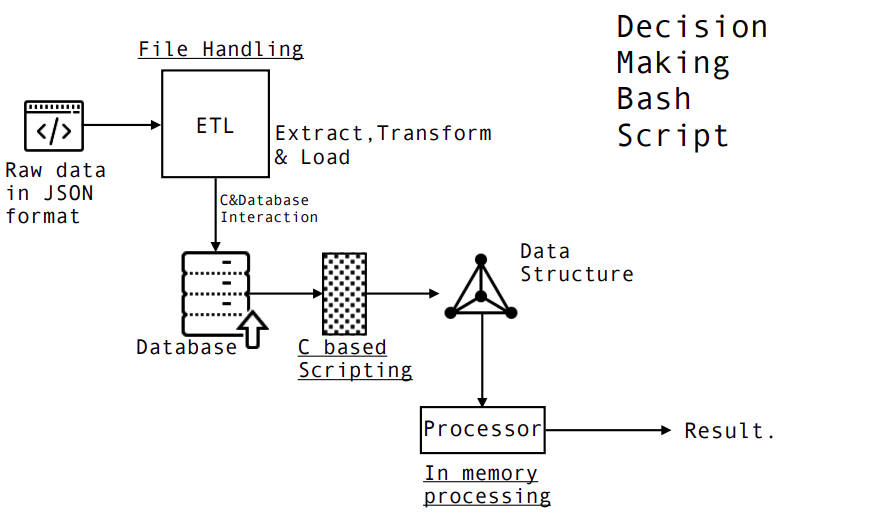
****

Fig: Context Diagram for Decision maker bash script

**Data Flow Diagrams:**

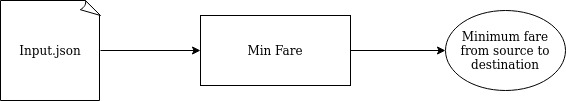


Fig: Level 1 DFD

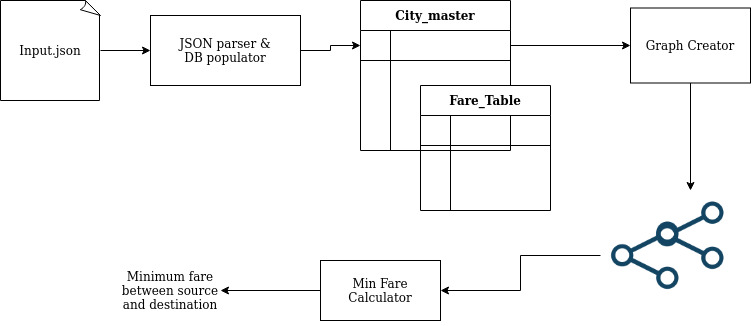


Fig: Level 2 DFD

**Entity Relationship Diagrams:**

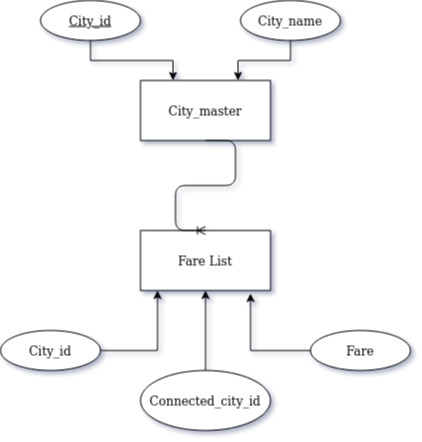
****

Fig: Entity relationship diagram for the database

**Snapshot of the project:**

****

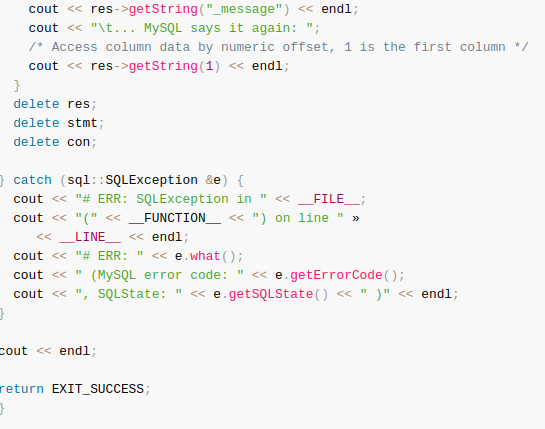
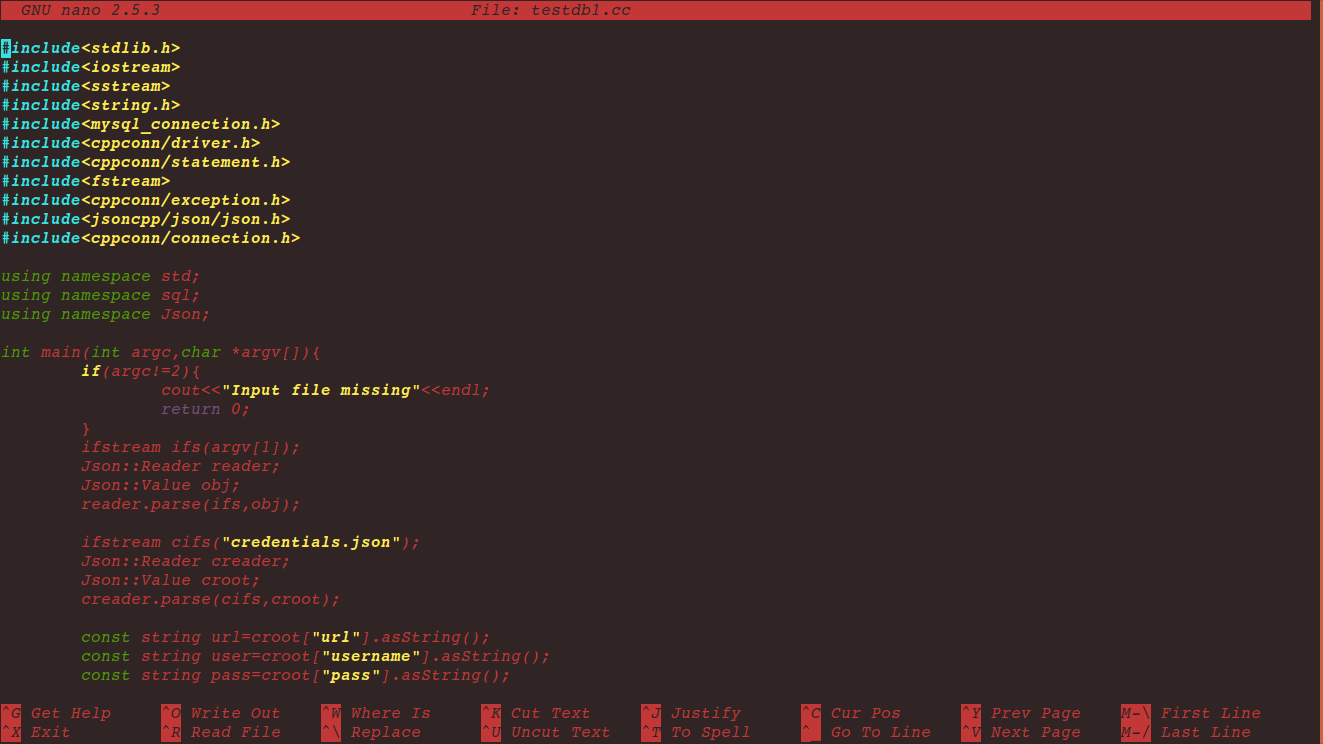
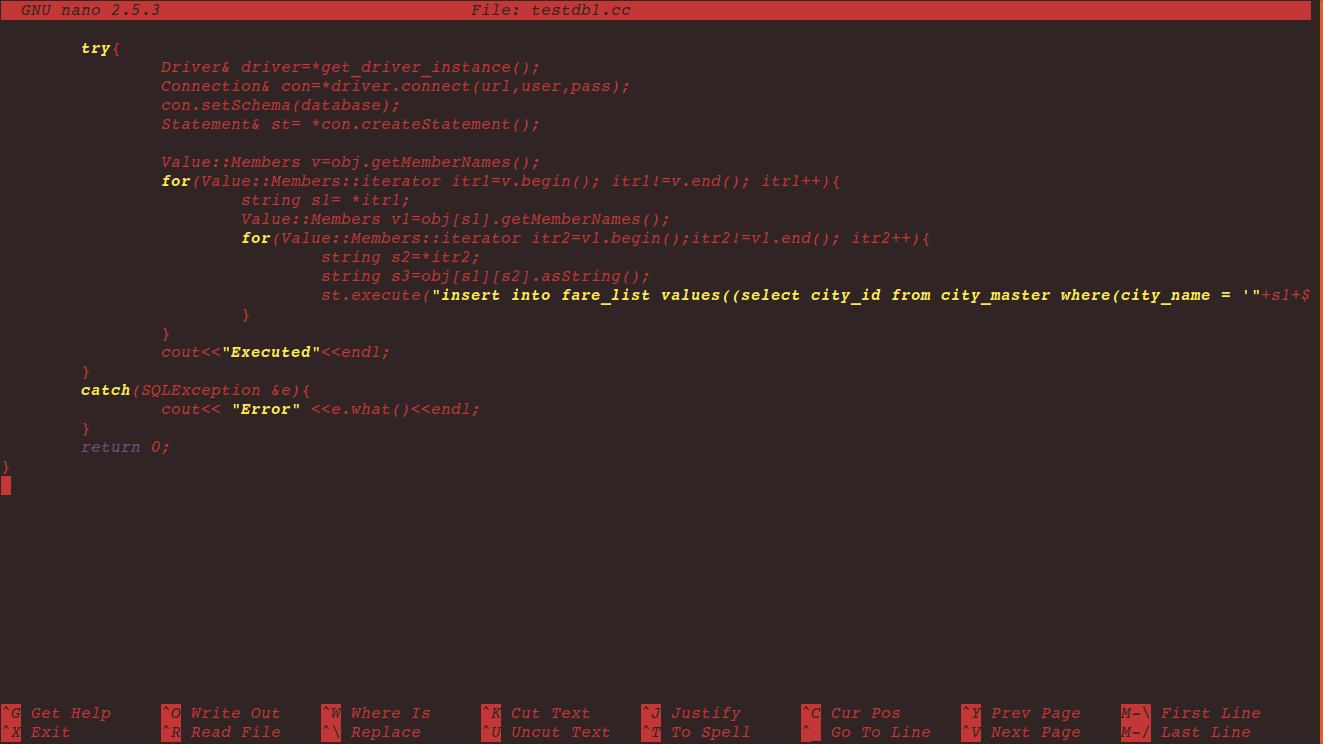
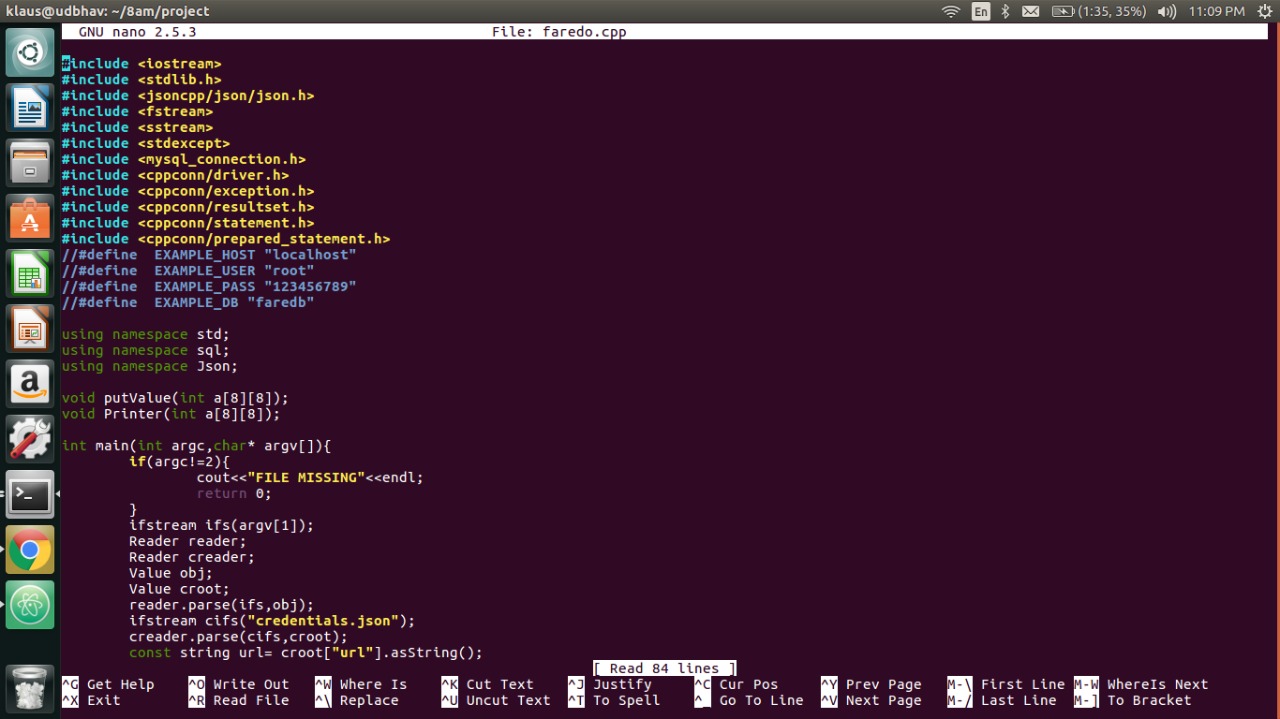
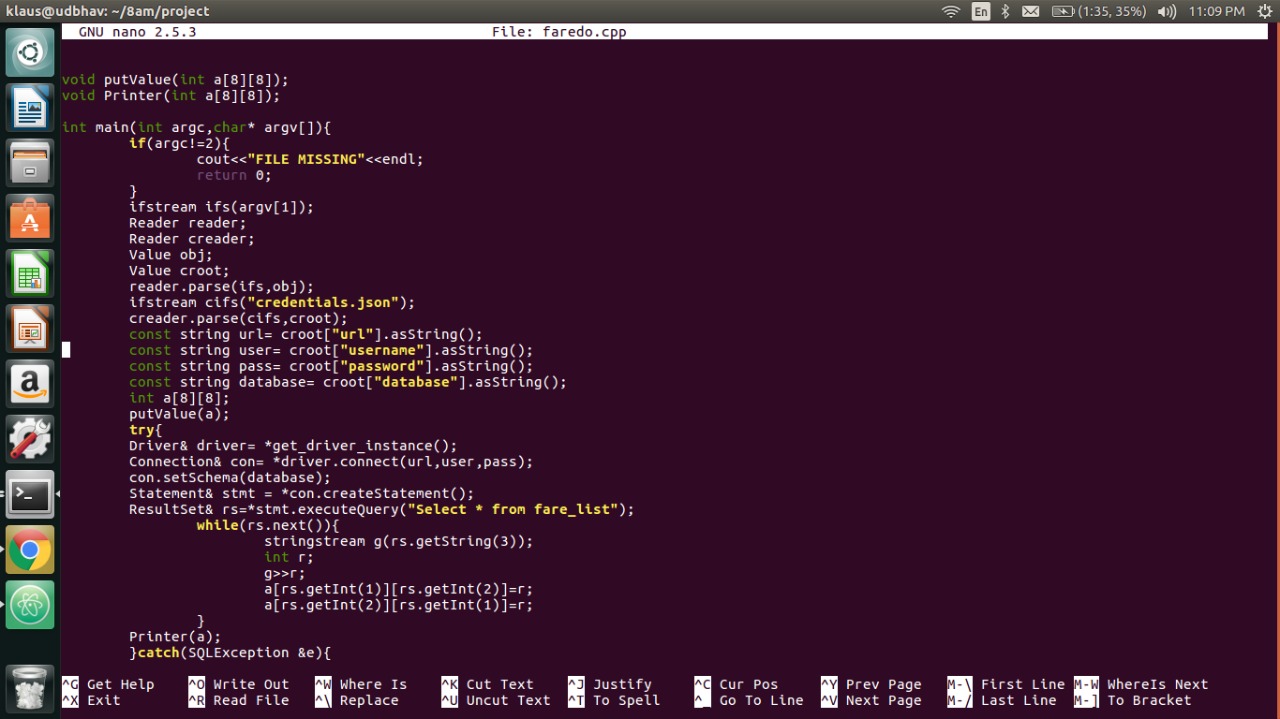
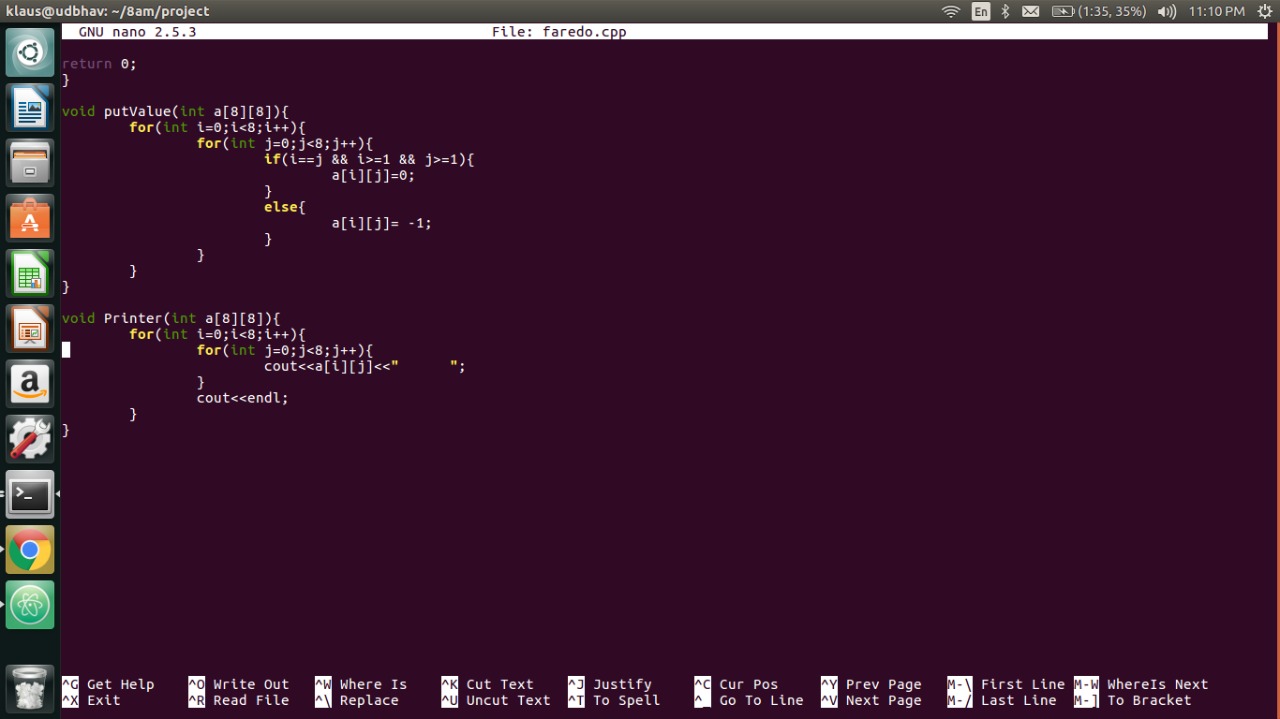


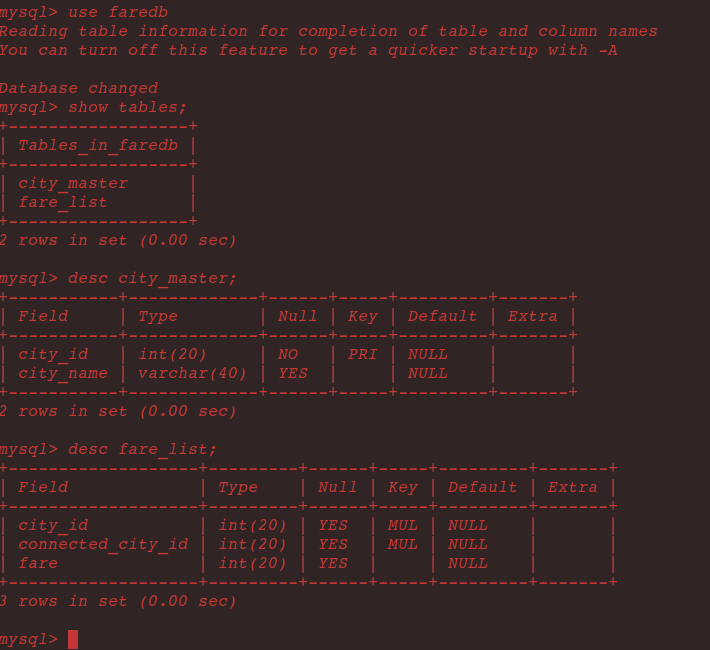
Fig: Database connectivity





****



****

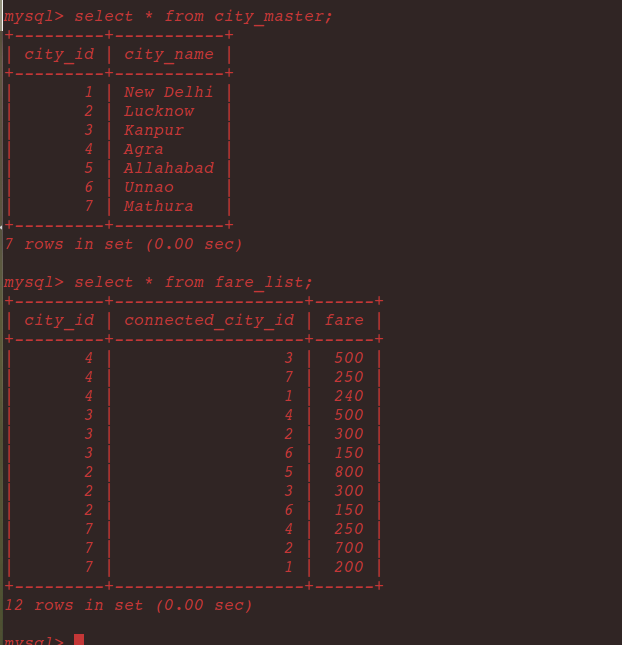
****

Fig: Database structure

**Result & Snapshot:**

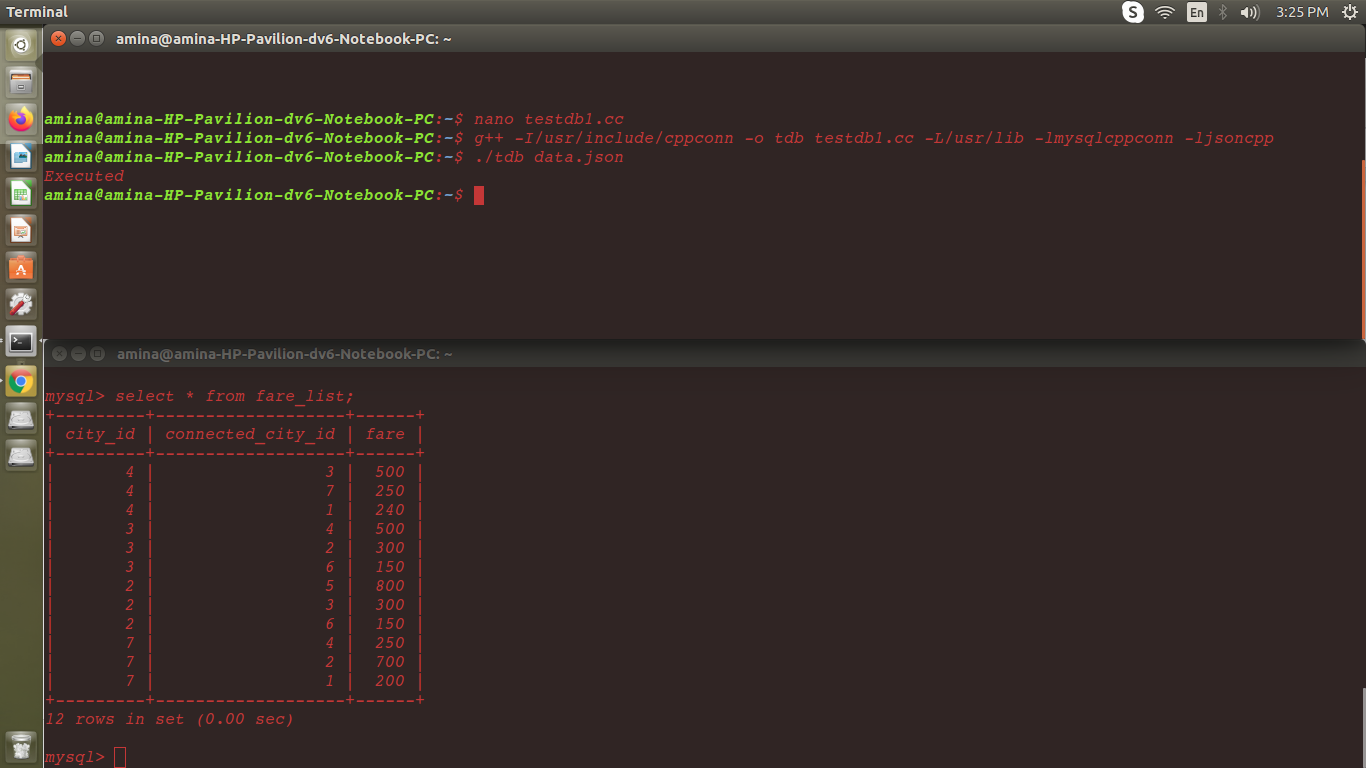
****

Fig: Inserting data into database using script

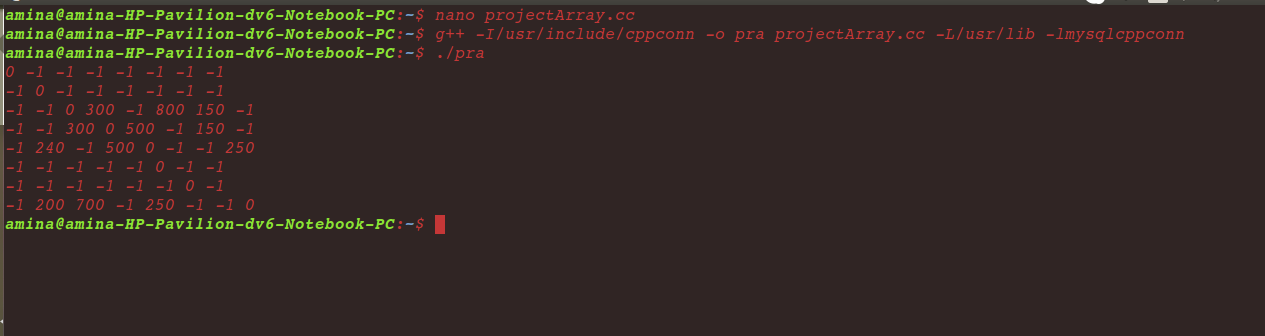
****

Fig: Populating data into a 2D array

**Conclusion**

**Future Scope:**

The scope of this paper is to create a bash script which can be used by any application developer to calculate the shortest path using basic data structure in a dynamic environment. This problem has many useful applications. The routing protocols used in most of today’s computer networks are based on shortest path algorithms. Machine learning on graphs is a young but growing field.

Neural networks are a way to create functions that [no human could write](https://arxiv.org/pdf/1812.04948.pdf). They do this by harnessing the power of large datasets. On problems for which we have capable neural models, we can use example inputs and outputs to train the network to learn a function that transforms those inputs into those outputs, and hopefully generalizes to other unseen inputs. We need to be able to build neural networks that can learn functions on graphs. Those neural networks need the right [inductive biases](https://arxiv.org/pdf/1806.01261.pdf) so that they can reliably learn useful graph functions. With that foundation, we can build powerful neural graph systems.

The shortest path algorithm can also be used in deep learning where the algorithm will work faster because it will use heuristics to make more educated guesses about which route is the smallest.

To overcome the limitations of this script a more efficient heuristic function can be developed. This newly designed script can be modified so that it always returns the shortest path possible.

**REFERENCES**

**Websites:**

* [www.google.co.in](http://www.google.co.in)
* [www.json.org/json-en.html](http://www.google.co.in)
* [www.geeksforgeeks.org](http://www.google.co.in)
* [www.wikipedia.com](http://www.wikipedia.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* [www.udemy.com](http://www.udemy.com)
* [www.w3schools.com](http://www.w3schools.com)
* [www.codeforces.com](http://www.codeforces.com)
* [www.mysql.com](http://www.mysql.com)
* <https://dev.mysql.com>
* <https://linux.tips/>
* <https://subscription.packtpub.com>
* <https://github.com/Microsoft/vcpkg>

**Books:**

* C++ PRIMER
* INTRODUCTION TO ALGORITHMS BY THOMAS H. CORMEN
* DATABASE SYSTEM CONCEPTS
* JAVASCRIPT-JSON-COOKBOOK
* DATA STRUCTURES AND ALGORITHMS MADE EASY BY NARASIMHA KARUMANCHI

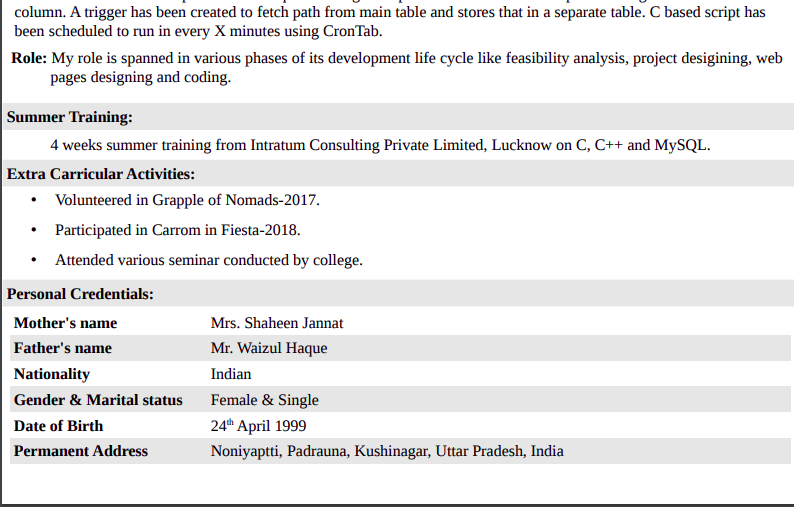
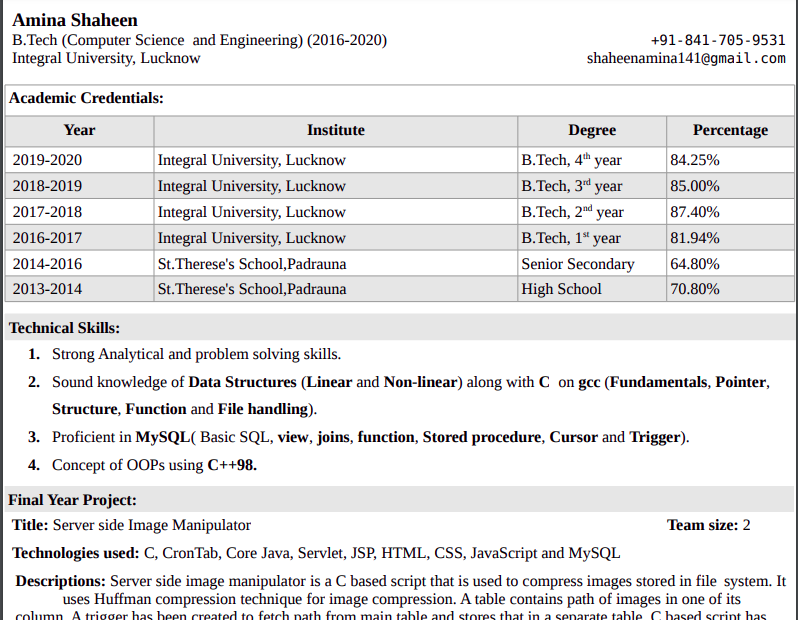
**BIO DATA** 

Fig: CV of Amina Shaheen

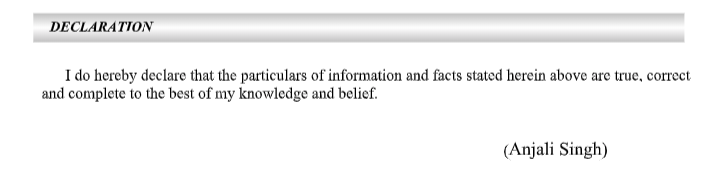
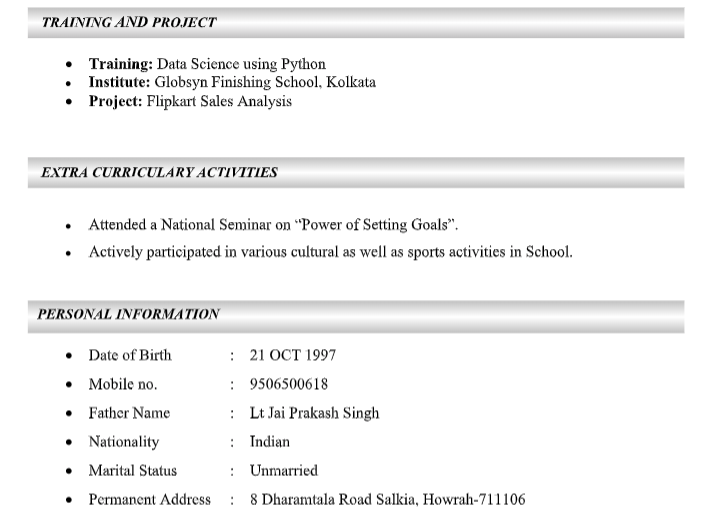
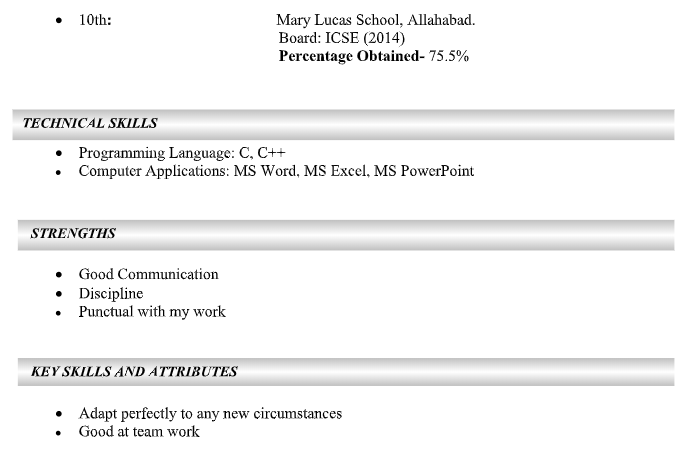
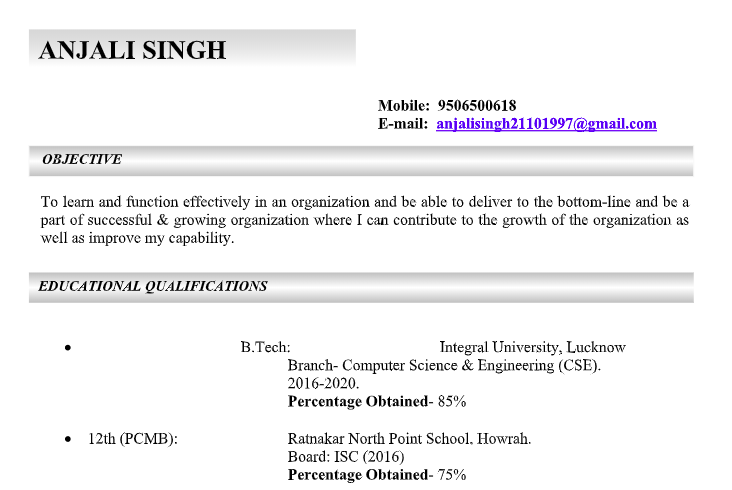


Fig: CV of Anjali Singh