**1.ACKNOWLEDGEMENT**

**Acknowledgement:-**

Fifth Semester Project is a major component of Academic Schedule of B.C.A. Hence I worked on **CineTicket Pro**.The conceptual Knowledge acquired by Management/ Computer student is best manifested in the project they undergo. The present project gives a perfect vent to my understanding of the practicalities of information of different Educational areas.I express my whole hearted gratitude towards **Sherwood College Of Professional Management** for having given me the opportunity to undergo my project in the field of Website development of great report and allowing me to gain invaluable experience. I express my heart-felt gratitude to **Mr.Sharad Nigam** for supervising me during the project period. I also express my special thanks to all the staff member who gave me their precious time and help me whenever required. I am also grateful to my parents who have always been supportive in giving me correct decision and advice.

I also express my sincere thanks to all the Respondents, without whose help the completion of this project report was not possible

[Student Name]

5th Semester

**2.Preface**

**Preface-**

I am glad to make this project **CineTicket Pro** and now I make this project report. In this project report, I include about my project, objective of project, lists of project activities etc.

Doing this project I have a great experience and I get knowledge about boutique system. These all experience and knowledge I am going to share by my project report.

During work on this project I met many technical problems. These technical problems are solved by our Teacher **Mr. Sharad Nigam.** He is very helpful for me in this project so I am very thankful to him.

**Thanking you.**

**3.Project Report**

# 3.1 INTRODUCTION

**3.1.1 BACKGROUND :-**

In the fast changing world, information technology and information management are going to play an important role. We are living in the computer age during past some year .The computer has gaining popularity. Computer revolution found its way into almost every aspect of human life and living. A computer is admirably suited to handle any information and hence is an information processor that is, it can receive data, perform some basic operations on that data and produces results according to a predetermined program.

This Software is used mainly for the multiplex cinema halls for booking the ticket and storing the information of customers.

The CineTicket Pro software is so designed as to ease the workload of multiplex cinema hall professionals. The main feature includes ticket booking, movie show details management and client management.

**3.1.2 OBJECTIVE :-**

Today’s world is computer world because most of work is doing with the help of computer. Dependency on computer is behind the few reasons. We cannot easily manage to store large number of data or information single handle. If we will be need some information or data in urgency then we cannot manage in manually these works are very difficult if we cannot use computer.

As this is generic software it can be used by a wide variety of multiplex cinema halls to automate the process of manually maintaining the records related to the subject of maintaining the movie details and customer data.

This software is basically updating the manual CineTicket Pro to automated ticket booking. So that organization can manage their record in efficient and organize them.

* The main objective is to automate non computer environment
* To save manpower.
* It will speed the processing of data and transaction.
* It will provide best security features such as provisions of passwords.
* **System Objective:-**

Today’s world is computer world because most of work is doing with the help of computer. Dependency on computer is behind the few reasons. We cannot easily manage to store large number of data or information single handle. If we will be need some information or data in urgency then we cannot manage in manually these works are very difficult if we cannot use computer.

* **System Context:-**

This section clearly depicts the environment and boundaries of the CineTicket Pro system and the entities with which it interacts. It helps us see how the system fits into the existing scheme of things. What the system will do by itself.

* **Functional Requirement:-**

This Software must request Username and Password for access to data, after authentication will allow access to the system. The Software must allow input of products data from administrator and secured access.

* **Non-Functional Requirement:-**

In this Software Input error will be returned in red with appropriate message box. System should automatically update after every transaction.

**3.1.3 PURPOSE AND SCOPE**

* **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 developing this software.

* **SCOPE :-**

As this is generic software it can be used by a wide variety of multiplex cinema halls to automate the process of manually maintaining the records related to the subject of maintaining the movie details and customer data.

**3.2 SURVEY OF TECHNOLOGIES**

**3.2.1 JAVA PROGRAMMING LANGUAGE :-**

**Java** is general-purpose [computer-programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [byte code](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). As of 2016, Java is one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity), particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems)(which has since been [acquired by Oracle Corporation](https://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The language derives much of its [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.

**3.2.2 MY SQL :-**

My SQL  is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). The My SQL development project has made its [source code](https://en.wikipedia.org/wiki/Source_code) available under the terms

the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) agreements. My SQL was owned and sponsored by a single [for-profit](https://en.wikipedia.org/wiki/Business) firm,the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQLAB](https://en.wikipedia.org/wiki/MySQL_AB" \o "MySQL AB),no owned by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation).

For proprietary use, several paid editions are available, and offer additional functionality.

**3.3 REQUIREMENTS & ANALYSIS**

**3.3.1 PROBLEM DEFINITION :-**

Cinema-going is one of the most popular out-of-home cultural activities, affecting a serious of social, economic and cultural phenomena in modern societies. Cinemas are considered to be an integral part of cities and they contribute to the definition of a local geography and identity. They also contribute to the preservation of the collective memory, since they constitute a significant social and cultural practice linked to a specific place, which acts as a common reference or landmark for many individuals. Through this project we present a comprehensive solution for ticket booking in multiplexes. Theater management system, and ticket selling software that is easy to understand, easy to use and offers the simplicity of fast point-and-click service to the employee and admin. This powerful software program is specifically designed for theater owners, to sell tickets. This intuitive visual interface makes day-to-day aspects of selling, exchanging, refunding, and reporting fast and easy for both the user and administrators. Theater Management controls all back-end and front-end functionalities like, movie details, ticket rate and show time, customer information and sales history saved in a database, etc. Theater admin Manages the report details like counter wise report, daily, weekly, monthly report and movie report etc.

**3.3.2 PLANNING AND SCHEDULING :-**

**PERT CHART :-**

A project plan needs to be created to ensure the timely completion of the project. As part of project analysis, we break the project down to a number of stages and use a Gantt chart and PERT chart to describe specific tasks and status.

The Work Breakdown Structure of our proposed system “E-Commerce” is shown below:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Task Name** | **Duration** | **Start** | **Finish** |
| 1 | Project Initiation | 2 days |  |  |
| 2 | Draft Project Plan | 3 days |  |  |
| 3 | Analysis Phase | 9 days |  |  |
| 4 | Plan User Interviews | 2 days |  |  |
| 5 | Schedule users Interviews | 4 days |  |  |
| 6 | Conducting users Interviews | 2 days |  |  |
| 7 | System Design | 14 days |  |  |
| 8 | Modules Design | 12 days |  |  |
| 9 | Data Structure Design | 3 days |  |  | |
| 10 | User Interface Design | 3 days |  |  | |
| 11 | Coding Phase | 38 days |  |  | |
| 12 | Testing Phase | 8 days |  |  | |
| 13 | Integration Testing | 5 days |  |  | |
| 14 | System Level Testing | 7 days |  |  | |
| 15 | Implementation | 4 days |  |  | |
| 16 | Post-Implementation Review | 2 days |  |  | |

**3.3.3 REQUIRMENTS SPECIFICATION :-**

* **Software Requirements :-**

1. Java/JDK
2. NetBeans
3. MySql
4. SQL YOG

* **Hardware Requirements :-**

1. Pentium IV Processor
2. 512 MB RAM
3. 40 GB HDD
4. Color Monitor
5. Keyboard, Mouse

**3.3.4 PRELIMINARY PRODUCT DESCRIPTION :-**

**MODULE DISCRIPTION :-**

This system will cover mainly two Modules, i.e.

1. User (EMPLOYEE).

2. Theatre (Admin).

* **Theatre Module will cover FOUR Sub modules**,

**a. Adding a Show.**

Theatre can add a show with timing and price of Ticket. This can be seen and booked by User.

**b. Deleting a Show.**

Theatre can delete show if it was performed

**c. Updating Show Details (Its Status).**

Theatre can Update show timing and Price

**d. Each and every Customer Detailing.**

Theatre can see each and every user details, which is registered (booked a ticket before)

* **EMPLOYEE Module will cover FOUR Sub modules,**

1. REGISTER CUSTOMER
2. Booking a Ticket. (House full if there is no Ticket)
3. Booking Details. (Booking Status)
4. Updating CUSTOMER Profile.

**3.3.5 CONCEPTUAL MODELS :-**

**SYSTEM DEVELOPMENT LIFE CYCLE :-**

The System development life cycle (SDLC), or Software development processing [systems engineering](http://en.wikipedia.org/wiki/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). These methodologies form the framework for planning and controlling the creation of an information system the [software development process](http://en.wikipedia.org/wiki/Software_development_process).

Broadly, following are the different activities to be considered while defining 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

**1.SYSTEM ANALYSIS :-**

Systems analysis is the study of sets of interacting entities, 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.

**1.2 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 organizations.

**ENTITY RELATION DIAGRAMS :-**

The Entity Relation Model or Entity Relation Diagram (ERD) is a data model or diagram for high-level description of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity relationship diagrams. Such models are typically used in the first stage of Management information system design; they are used for example, to describe information needs and/ or the type of information that is to be stored in the Database during the requirement analysis. The data modeling technique, however, can be used to describe any ontology (i.e. an overview and classification of used term and their relationships) for a certain universe of discourse (i.e. area of interest).

In the case of design a Management Information System that is based on a database, the conceptual data model is, a later stage( usually called logical design), mapped to a logical data model such as, relational data model; this is turn in mapped to a physical model during physical design. Note that sometimes, both of the phases are referred a “physical design”. There are number of convention for entity-relation diagrams (ERDs). The classical notation is describe in the remainder of this article, and mainly related to the conceptual modeling. There is a range of notation more typically employed in physical and logical database design.

web

contactno

MAN-AGES

**Theatre**

**Add Movie**

securityques

**Regist-ration**

MAN-AGES

Name

User Theatre

**Book Ticket**

BUY

**User**

Movdate Movname

Seat

MAN-AGES

**User\_Image**

ENTITY RELATIONSHIP DIAGRAM

**DATA FLOW DIAGRAM :-**

The data flow diagram shows the flow of data within any system. It is an important tool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It’s also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD.

In the DFD, four symbols are used and they are as follows.

1. A square defines a source (originator) or destination of system data.
2. An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.
3. A circle or a “bubble “(Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.
4. An open rectangle is a data store-data at rest, or a temporary

Repository of data.

**Context Level Data Flow Diagram :-**

This level shows the overall context of the system and its operating environment and shows the whole system as just one process. Online book store is shown as one process in the context diagram; which is also known as zero level DFD, shown below. The context diagram plays important role in understanding the system and determining the boundaries. The main process can be broken into sub-processes and system can be studied with more detail; this is where 1st level DFD comes into play.

EMPLOYEE

COUSTOMER

Request

ADMIN

Request fulfill

Zero Level Data Flow Diagram

**First Level DFD :-**

This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major high-level processes of the system and their interrelation. A process model will have one, and only one, level-1 diagram. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level 1

ADD/UPDATE/DELETE MOVIE

ADD/UPDATE CUSTOMER INFORMATION

ADMIN/EMPLOYEE

TICKET BOOKING AND

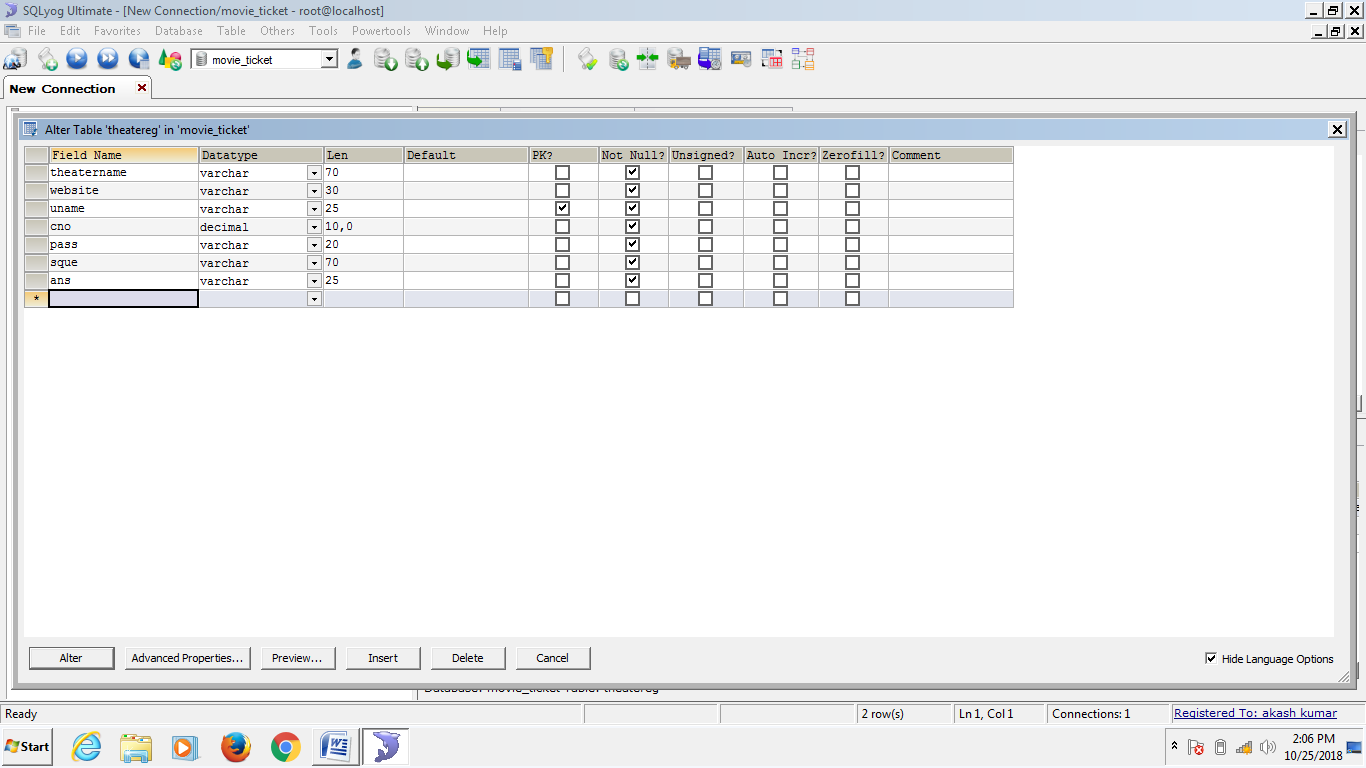
REPORT

ADD/UPDATE DELETE EMPLOYEE DETAILS

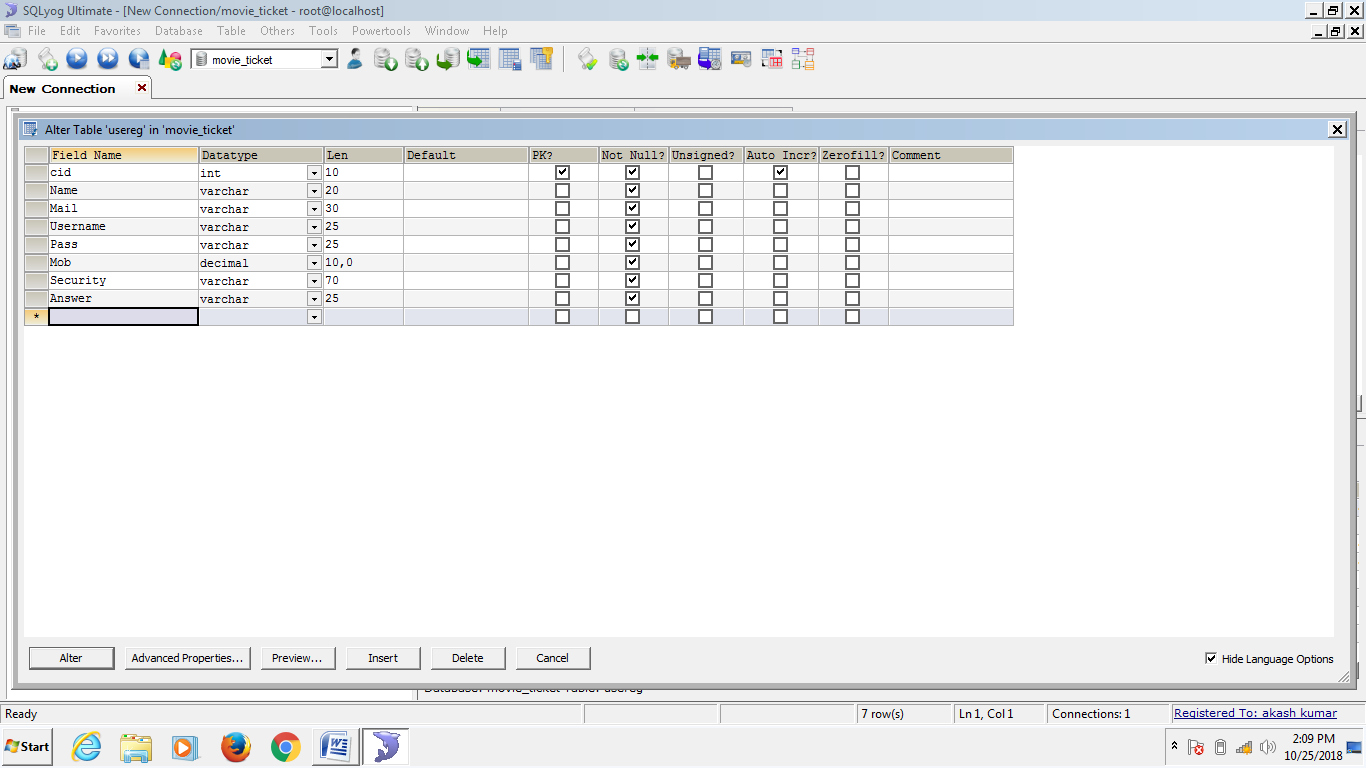
ONE LEVEL DATA FLOW DIAGRAM

**ScreenShot Of Data Table:-**

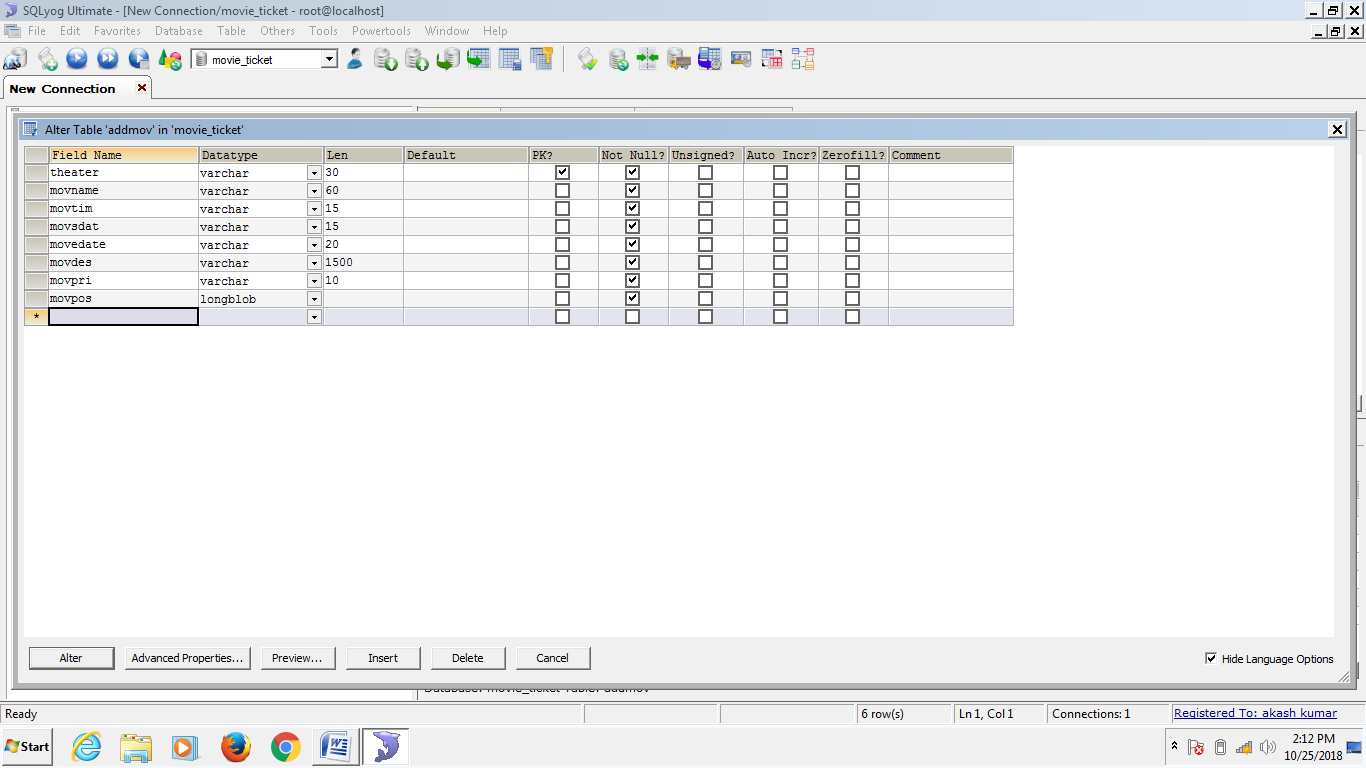
**THEATRE REGISTRATION:-**

****

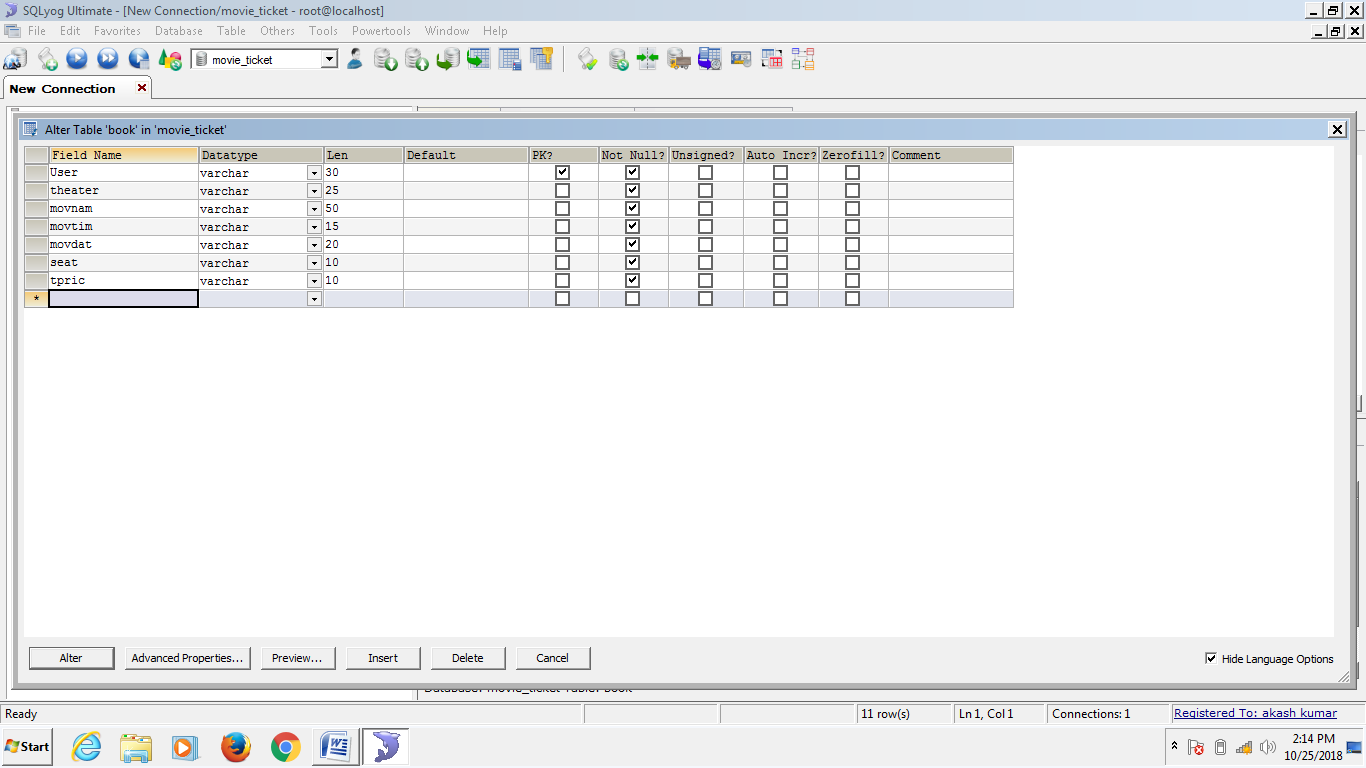
**USER REGISTRATION:-**

****

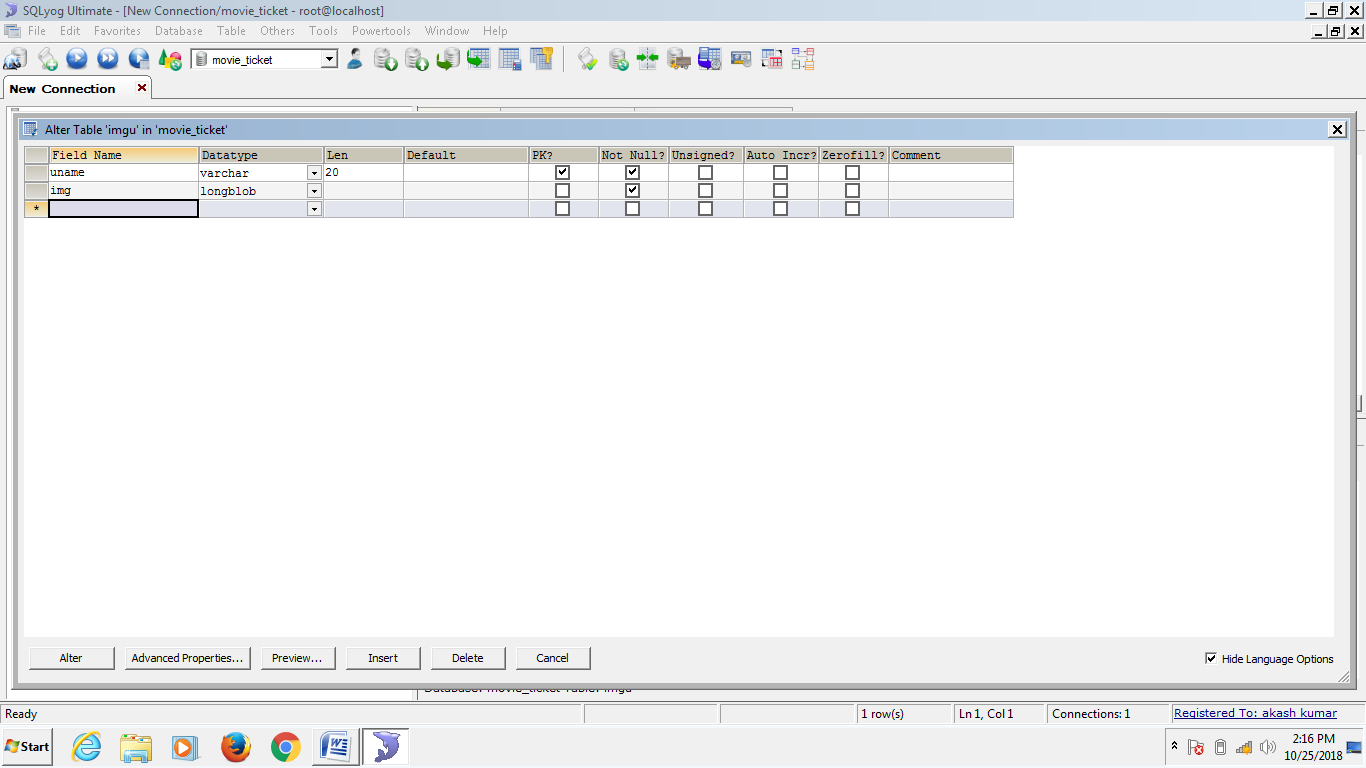
**ADD MOVIE:-**

****

**BOOK TICKET:-**

****

**USER IMAGE:-**

****

**Data Tables:-**

**THEATRE REGISTRATION :-**

|  |  |  |  |
| --- | --- | --- | --- |
| Name Column | Data Type | Constraints | Size |
| Theatername | Varchar | Not Null | 70 |
| Website | Varchar | Not Null | 30 |
| Username | Varchar | Primary Key | 25 |
| Contectno. | Decimal | Not Null | 10,0 |
| Password | Varchar | Not Null | 20 |
| Securityquestion | Varchar | Not Null | 70 |
| Answer | Varchar | Not Null | 25 |

**USER REGISTRATION :-**

|  |  |  |  |
| --- | --- | --- | --- |
| Name Column | Data Type | Constraints | Size |
| Customerid | Int | Primary key | 10 |
| Name | Varchar | Not Null | 20 |
| Mail | Varchar | Not Null | 30 |
| Username | Varchar | Not Null | 25 |
| Password | Varchar | Not Null | 25 |
| Mobile | Decimal | Not Null | 10,0 |
| Security | Varchar | Not Null | 70 |
| Answer | Varchar | Not Null | 25 |

**ADD MOVIE :-**

|  |  |  |  |
| --- | --- | --- | --- |
| Name Column | Data Type | Constraints | Size |
| Theatre | Varchar | Primary key | 30 |
| Movname | Varchar | Not Null | 60 |
| Movtimming | Varchar | Not Null | 15 |
| Movsdate | Varchar | Not Null | 15 |
| Movedate | Varchar | Not Null | 20 |
| Movdescription | Varchar | Not Null | 1500 |
| Movprice | Varchar | Not Null | 10 |
| Movposter | Longblob | Not Null | 10 |

**BOOK TICKET :-**

|  |  |  |  |
| --- | --- | --- | --- |
| Name Column | Data Type | Constraints | Size |
| User | Varchar | Primary key | 30 |
| Theatre | Varchar | Not Null | 25 |
| Movname | Varchar | Not Null | 50 |
| Movtime | Varchar | Not Null | 15 |
| Movdate | Varchar | Not Null | 20 |
| Seat | Varchar | Not Null | 10 |
| Tprice | Varchar | Not Null | 10 |

**USER IMAGE :-**

|  |  |  |  |
| --- | --- | --- | --- |
| Name Column | Data Type | Constraints | Size |
| Uname | Varchar | Primary key | 20 |
| Image | Longblob | Not Null | 10 |