KAMALA NIKETAN MONTESSORI SCHOOL (CBSE) – TRICHY

THEATRE BOOKING APPLICATION

DONE BY -

Harini.R

Parameshwari.S

Preity.M

ACKNOWLEDGEMENT

I sincerely thank my teachers, parents and friends for helping me finish this project efficiently. I especially thank Mrs. Mala Sivakumar and Mrs. Vasavi, my teacher, for giving me this opportunity.

I extend my hearty thanks to my parents who supported and guided me, and my friends who cooperated extremely well throughout the project, and without whom this would not have been possible.

INDEX

INTRODUCTION: NETBEANS

NetBeans is a software developmentplatform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The editor supports many languages from Java, C/C++, XML and HTML, to PHP, Groovy, Javadoc, JavaScript and JSP. Because the editor is extensible, you can plug in support for many other languages.

The NetBeans Team actively supports the product and seeks feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.[5]
A new version was released 8.2/october 3,2016.NetBeans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references.

MYSQL

MySQL (officially pronounced as /maɪ ˌɛskjuːˈɛl/ "My S-Q-L") is an opensourcerelational database management system (RDBMS).Its name is a combination of "My", the name of co-founder Michael Widenius's daughter,and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), FacebookTwitter, Flickr, and YouTube.

Java Database Connectivity

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the

JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection. JDBC represents statements using one of the following classes:

Statement – the statement is sent to the database server each and every time.

PreparedStatement – the statement is cached and then the execution path is pre-determined on the database server allowing it to be executed multiple times in an efficient manner.

CallableStatement – used for executing stored procedures on the database.

Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many rows were affected in the database. These statements do not return any other information.

Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

OBJECTIVE

The main aim of this Theatre Booking Application is to manage time and make ticket booking and billing processes quicker and more efficient. This application is primarily for booking tickets for movies in different movie theatres according to the users' convenience. It provides various bases on which users select movies they want to watch. This will help the users select quickly. It lets the user choose from movies with broader options by considering more perspectives, unlike the narrow-minded concepts of booking applications present nowadays. This application categorizes all the movies according to the users' wishes.

This booking application provides an environment no less than a real counter in a movie theatre, providing image posters of the movies, also showing available discounts, if any, where they're put up.

This application is even more helpful when it comes to other facilities; as it provides details about the different movie theatres and the different show timings they have. This is extremely effective as the users can choose to watch where they want to; a good quality theatre or a theatre right around the corner of their house;

according to their wish. Booking can be done whenever and wherever, and the tickets are delivered to the addresses the users provide with payment or they could be collected by the buyers at the venue itself, prior to when the movie is due.

Also, this application allows the users to choose where they wish to sit inside the cinema by letting them choose the seats from an aerial view of the theatre. This will help create a more comfortable environment for the watchers.

It includes different payment options like payments through credit or debit cards or payment by cash at the venue itself.

Also, the billing process is made quicker, easier and error-free, requiring no human labour, as the application itself is programmed to calculate total price, validating discount rates and taxes.

We hope our little application is innovative in its own way, making booking simpler, more effective and time-saving.

PROBLEM

We wouldn’t necessarily call them problems;there are few inconveniences associated with booking tickets via other processes; apart from online booking.

Here we’re listing out these inconveniences as per the main problem is that the users are not aware of where the different shows are available and their timings. They tend to go to the respective theatres themselves to buy tickets. They do not know right show timings which will pose an extreme waste of time and money.

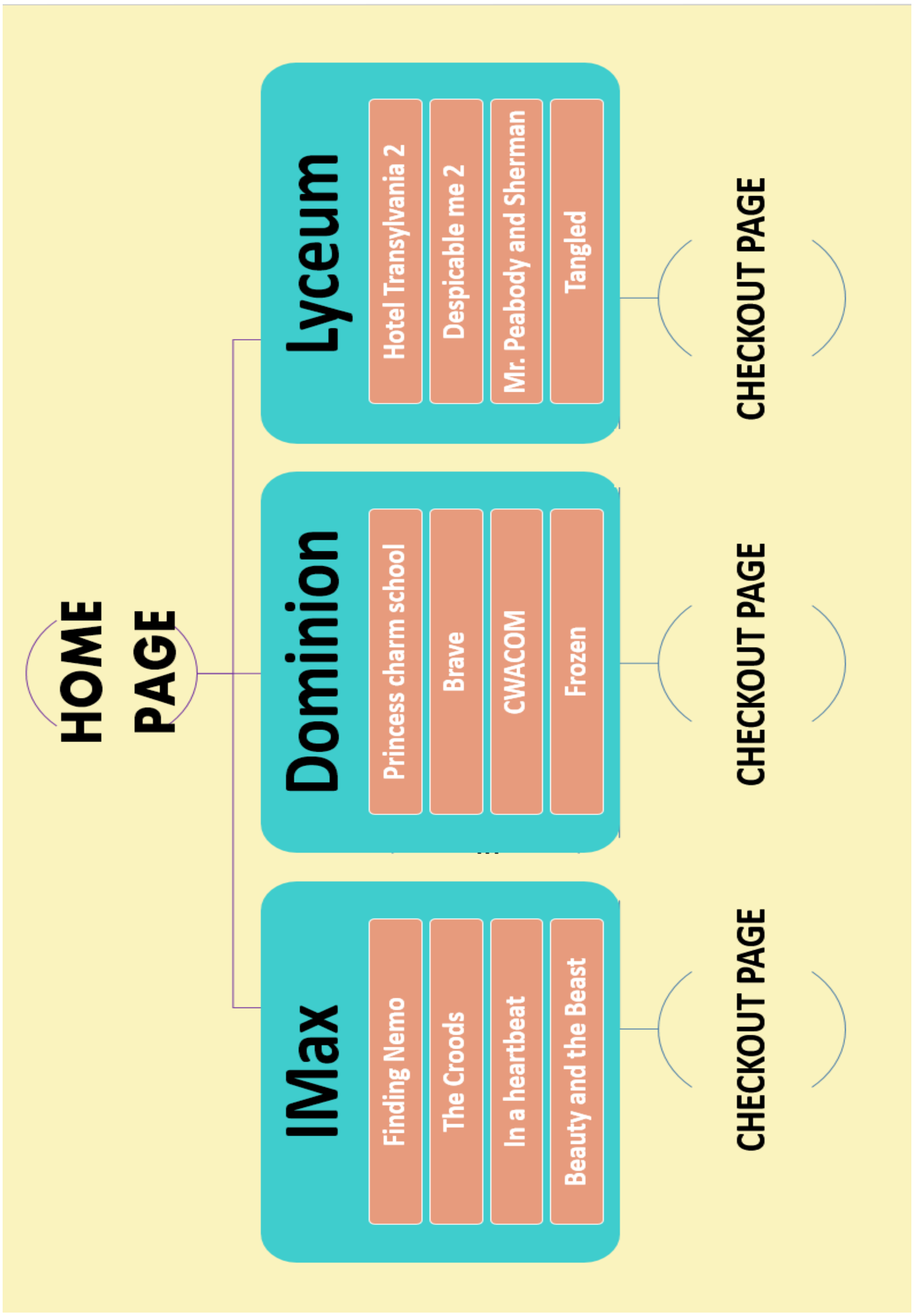
Secondly,they are forced to come to the venue prior to the show to receive their tickets which is a waste of time and not required.

They’re also forced to stand in long queues which is highly inconvenient for elderly people,women and children.

Last but not the least people are not given the privilege of choosing their own seats according to their wish. They’re forced to sit where they are allotted to sit.

These are a few inconveniences we recognised,for which we have developed this application, hoping to sort out these little problems; making ticket booking a convenient and easy process.

FRAME DETAIL



CODING

```
private void
iMAXActionPerformed(java.awt.event.ActionEvent evt)
  {
 new iMax().setVisible(true);
}

  private void
DOMINIONActionPerformed(java.awt.event.ActionEve
nt evt)
{
 new DOMINION().setVisible(true);
 }

  private void
LYCEUMActionPerformed(java.awt.event.ActionEvent
evt)
  {
new lyceum().setVisible(true);
}

  public static void main(String args[])
  {
```

```
private void
jButton1.Performed(java.awt.event.ActionEvent evt)
      {
       new BOOKTICKETS().setVisible(true);
      }
    private void
jButton2.Performed(java.awt.event.ActionEvent evt)
      {
       new BOOKTICKETS().setVisible(true);
      }
    private void
jButton3.Performed(java.awt.event.ActionEvent evt)
      {
       new BOOKTICKETS().setVisible(true);
      }
    private void
jButton4.Performed(java.awt.event.ActionEvent evt)
      {
       new BOOKTICKETS().setVisible(true);
      }
 public BEAUTY_AND_THE_BEAST()
  initComponents();
    }
 private void
jButton1.Performed(java.awt.event.ActionEvent evt)
        {
          new BOOKTICKETS().setVisible(true);
        }
      public static void main(String args[])
        {
          java.awt.EventQueue.invokeLater(new
Runnable()
            {
             public void run()
              {
               new
BEAUTY_AND_THE_BEAST().setVisible(true);
              }
         public NEMO()
         initComponents();
    }
       private void
jButton2.Performed(java.awt.event.ActionEvent evt)
        {
         new BOOKTICKETS().setVisible(true);
        }
  public static void main(String args[])
```

```
       {
     java.awt.EventQueue.invokeLater(new Runnable())
        {
          public void run()
           {
            new NEMO().setVisible(true);
           }
   public THE_CROODS()
     initComponents();
        }
     private void
jButton3.Performed(java.awt.event.ActionEvent evt)
       {
        new BOOKTICKETS().setVisible(true);
       }
     public static void main(String args[])
       {
      java.awt.EventQueue.invokeLater(new Runnable()
         {
          public void run()
           {
            new THE_CROODS().setVisible(true);
           }
 public IN_A_HEARTBEAT()
```

```
        initComponents();
            }
       private void
jButton4.Performed(java.awt.event.ActionEvent evt)
        {
         new BOOKTICKETS().setVisible(true);
        }
        public static void main(String args[])
         {
          java.awt.EventQueue.invokeLater(new
Runnable()
            {
             public void run()
              {
               new IN_A_HEARTBEAT().setVisible(true);
              }
         public static void main(String args[])
          {
      private void
jButton1.Performed(java.awt.event.ActionEvent evt)
       {
       new BOOKTICKETS().setVisible(true);
       }
```

```
    private void
jButton2.Performed(java.awt.event.ActionEvent evt)
     {
      new BOOKTICKETS().setVisible(true);
     }
    private void
jButton3.Performed(java.awt.event.ActionEvent evt)
     {
      new BOOKTICKETS().setVisible(true);
     }
    private void
jButton4.Performed(java.awt.event.ActionEvent evt)
     {
      new BOOKTICKETS().setVisible(true);
     }
  public PRINCESS_CHARM_SCHOOL()
      initComponents();
      }
     private void
jButton1.Performed(java.awt.event.ActionEvent evt)
     {
     new BOOKTICKETS().setVisible(true);
     }
 public static void main(String args[])
```

```
      {
       java.awt.EventQueue.invokeLater(new Runnable()
        {
          public void run()
           {
            new
PRINCESS_CHARM_SCHOOL().setVisible(true);
           }
   public THE_BRAVE()
       initComponents();
       }
      private void
jButton2.Performed(java.awt.event.ActionEvent evt)
       {
        new BOOKTICKETS().setVisible(true);
       }
      public static void main(String args[])
       {
        java.awt.EventQueue.invokeLater(new Runnable()
         {
          public void run()
           {
            new THE_BRAVE().setVisible(true);
 }
```

```
  public CLOUDY_WITH_A_CHANCE_OF_MEATBALLS()
      initComponents();
      }
     private void
jButton3.Performed(java.awt.event.ActionEvent evt)
      {
       new BOOKTICKETS().setVisible(true);
  }
     public static void main(String args[])
      {
       java.awt.EventQueue.invokeLater(new Runnable()
         {
   public void run()
       {
        new
CLOUDY_WITH_A_CHANCE_OF_MEATBALLS().setVisibl
e(true);
       }
   public FROZEN()
      initComponents();
      }
     private void
jButton4.Performed(java.awt.event.ActionEvent evt)
```

```
      {
       new BOOKTICKETS().setVisible(true);
      }
     public static void main(String args[])
      {
       java.awt.EventQueue.invokeLater(new Runnable()
        {
         public void run()
          {
           new FROZEN().setVisible(true);
          }
      public static void main(String args[])
       {
      private void
jButton1.Performed(java.awt.event.ActionEvent evt)
       {
        new BOOKTICKETS().setVisible(true);
       }
      private void
jButton2.Performed(java.awt.event.ActionEvent evt)
        {
         new BOOKTICKETS().setVisible(true);
        }
       private void
jButton3.Performed(java.awt.event.ActionEvent evt)
        {
         new BOOKTICKETS().setVisible(true);
        }
       private void
jButton4.Performed(java.awt.event.ActionEvent evt)
        {
         new BOOKTICKETS().setVisible(true);
        }
     public HOTEL_TRANSYLVANIA()
        initComponents();
        }
        private void
jButton2.Performed(java.awt.event.ActionEvent evt)
        {
         new BOOKTICKETS().setVisible(true);
        }
        public static void main(String args[])
        {
         java.awt.EventQueue.invokeLater(new
Runnable()
          {
 public void run()
```

```
            {
             new HOTEL_TRANSYLVANIA().setVisible(true);
            }
      public DESPICABLE_ME()
         initComponents();
         }
        private void
jButton2.Performed(java.awt.event.ActionEvent evt)
         {
          new BOOKTICKETS().setVisible(true);
         }
       public static void main(String args[])
         {
          java.awt.EventQueue.invokeLater(new
Runnable()
           {
            public void run()
 {
             new DESPICABLE_ME().setVisible(true);
            }
public MR_PEABODY_AND_SHERMAN()
         initComponents();
         }
        private void
jButton2.Performed(java.awt.event.ActionEvent evt)
         {
          new BOOKTICKETS().setVisible(true);
         }
       public static void main(String args[])
         {
          java.awt.EventQueue.invokeLater(new
Runnable()
           {
            public void run()
             {
              new
MR_PEABODY_AND_SHERMAN().setVisible(true);
             }
     public TANGLED()
        initComponents();
         }
        private void
jButton2.Performed(java.awt.event.ActionEvent evt)
         {
          new BOOKTICKETS().setVisible(true);
         }
        public static void main(String args[])
        {
         java.awt.EventQueue.invokeLater(new
Runnable()
         {
          public void run()
           {
new TANGLED().setVisible(true);
           }
    private void
jButton1ActionPerformed(java.awt.event.ActionEvent
evt)
     {
         String NAME=jTextField1.getText();
         String MOBILE_NUMBER=jTextField2.getText();
         int TOTAL_AMOUNT;
 int discount;
             int
NO_OF_SEATS=Integer.parseInt(jTextField3.getText());
        if(Silver.isSelected())
             {
                  discount=40;
                   TOTAL_AMOUNT=(320*NO_OF_SEATS)-
discount;
JOptionPane.showMessageDialog(this,"WELCOME
"+NAME+" YOUR SEATS HAVE BEEN BOOKED");
jTextField4.setText(Intger.toString(TOTAL_AMOUNT));
              try
                        {
Class.forName("java.sql.DriverManager");
  Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:
3306/theatrebooking","root","school");
                         Statement stmt = (Statement)
con.createStatement();
                         String query = "insert into details
values('"+NAME+"','"+MOBILE_NUMBER+"','"+NO_OF_
SEATS+"')";
     stmt.executeUpdate(query);
                       }
             catch(Exception e)
                       {
JOptionPane.showMessageDialog(this,e.getMessage());
                       }
              }


     else if(Gold.isSelected())
```

```
            {
                discount=40;
                 TOTAL_AMOUNT=(320*NO_OF_SEATS)-
discount;
JOptionPane.showMessageDialog(this,"WELCOME
"+NAME+" YOUR SEATS HAVE BEEN BOOKED");
jTextField4.setText(Intger.toString(TOTAL_AMOUNT));
                 try
                     {

Class.forName("java.sql.DriverManager");
  Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:
3306/theatrebooking","root","school");
                      Statement stmt = (Statement)
con.createStatement();
                      String query = "insert into details
values('"+NAME+"','"+MOBILE_NUMBER+"','"+NO_OF_
SEATS+"')";
                      stmt.executeUpdate(query);
                    }
                catch(Exception e)
                 {
JOptionPane.showMessageDialog(this,e.getMessage());
```

```
                    }
             }
 else if(Platinum.isSelected())
             {
                  discount=40;
                   TOTAL_AMOUNT=(320*NO_OF_SEATS)-
discount;
JOptionPane.showMessageDialog(this,"WELCOME
"+NAME+" YOUR SEATS HAVE BEEN BOOKED");
jTextField4.setText(Intger.toString(TOTAL_AMOUNT));
                   try
                     {

Class.forName("java.sql.DriverManager");
  Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:
3306/theatrebooking","root","school");
                     Statement stmt = (Statement)
con.createStatement();
                     String query = "insert into details
values('"+NAME+"','"+MOBILE_NUMBER+"','"+NO_OF_
SEATS+"')";
     stmt.executeUpdate(query);
              }
```

```
          catch(Exception e)
                {
JOptionPane.showMessageDialog(this,e.getMessage());
                }
           }
         }
```

BIBLIOGRAPHY

1. *https://en.wikipedia.org/wiki/Moana_(2016_film)*
2. *https://en.wikipedia.org/wiki/The_Croods*
3. *https://en.wikipedia.org/wiki/Brave_(2012_film)*
4. *https://en.wikipedia.org/wiki/Cloudy_with_a_Chance_of_Meatballs_2*
5. *https://en.wikipedia.org/wiki/Despicable_Me_2*
6. *http://cinematreasures.org/theaters/22220*
7. *Images.google.com*
8. *IP TEXTBOOK CLASS XII*