GYM MANAGEMENT SOFTWARE

# AIM:

To design and develop a gym management software to manage registration of customers, management of customer’s data, management of batches.

# INTRODUCTION:

In the era of rapid technological advancement, every task is being computerised with minimum or no manual intervention for effective management. Computerisation of various tasks has led to efficient delivery of services and rapid rise in revenue to businesses. It enables users to perform their tasks easily.

# SOFTWARE AND TOOLS USED:

Java , Servlets, JSP, mysql and JDBC.

# PROJECT DEVELOPMENT PROCESS:

Project is designed in 4 sprints. During the first sprint, Database is created and establishing the connection of java with database is done through JDBC. User Interface for creation of batch is designed using html. User interface for registration of participants is desgined using JSP.

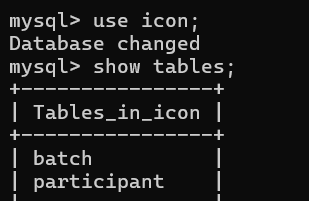
During the Second sprint, Methods were designed to insert the data into database, query the data from database(for both batches and participants). User interface to view the list of batches, list of participants and also individual batches and individual participants are created.

During the Third Sprint, Methods which define the functionality for editing and deleting the database are designed. User interface where all crud operations for both batches and participants can be performed at one place is designed.

During the Fourth Sprint, Decorative architecture of web pages is designed.

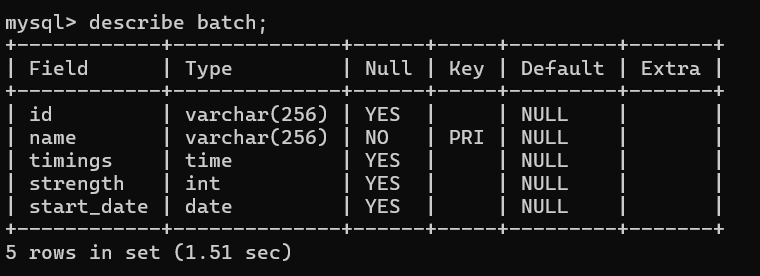
# PROJECT DESIGN:

Java is used to design a class for participant and batch. These classes define their attributes and structure. Another class Db is created to define attributes such as for PreparedStatement, Connection and methods to establish connection to database, inserting data into the database, querying the data, editing the data and deleting the data from database. Mysql is used to create a database icon and two tables “batch” and “participant” are created in database icon .



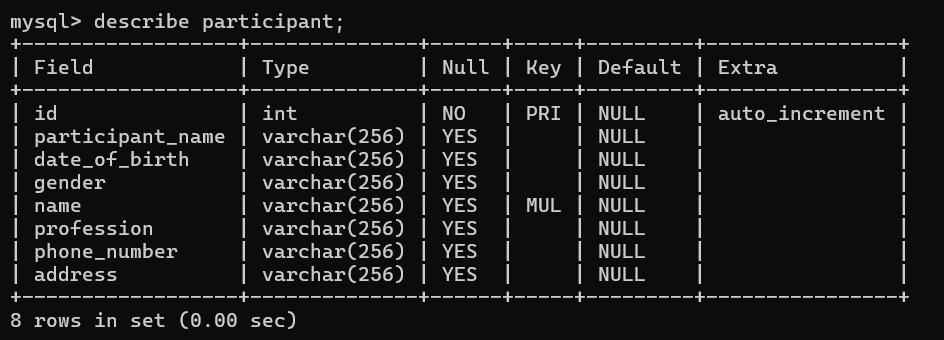
Created Table batch:

create table batch(id varchar(256), name varchar(256) primary key, timings time, strength int, start\_date date);



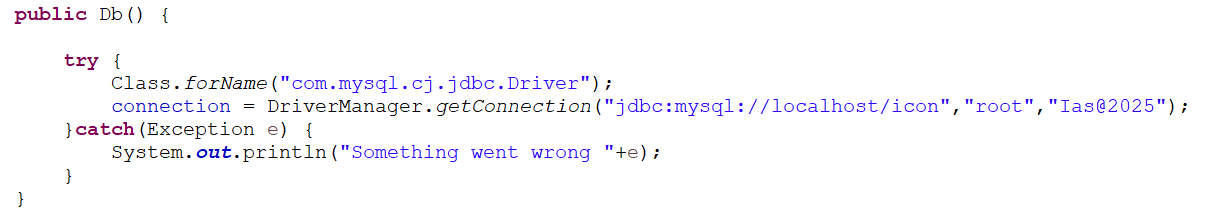
Created table participant:

create table participant(id int primary key auto\_increment, participant\_name varchar(256), date\_of\_birth varchar(256), gender varchar(256), name varchar(256), profession varchar(256), phone\_number varchar(256), address varchar(256), foreign key(name) references batch(name) );



**Methods and variables to establish database connection:**

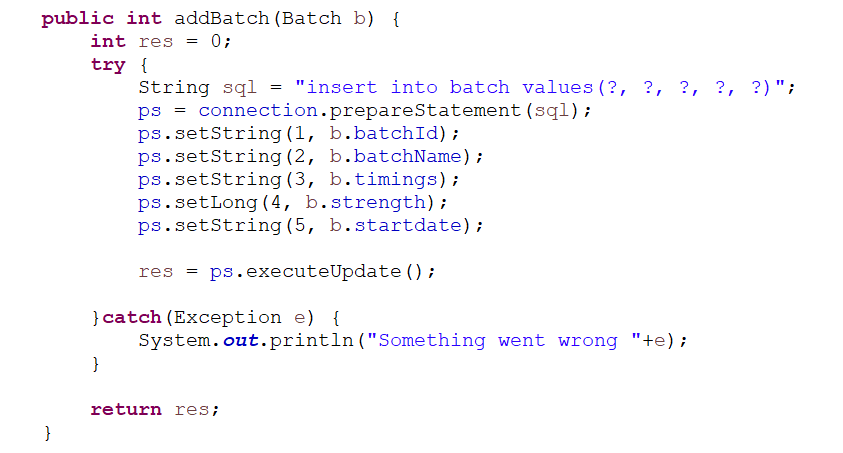
Variable for prepared statement is defined. Variable for connection is also defined. A non parameterized constructor is called to establish connection.



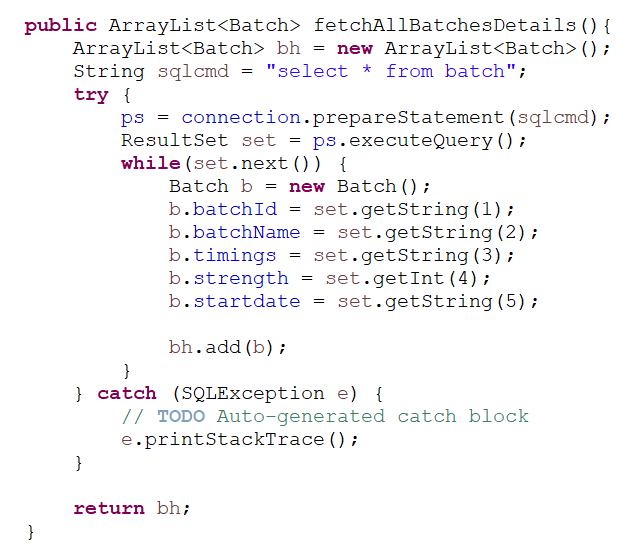
public void connection() is defined to close connection.

**Methods for Participants in Db Class:**

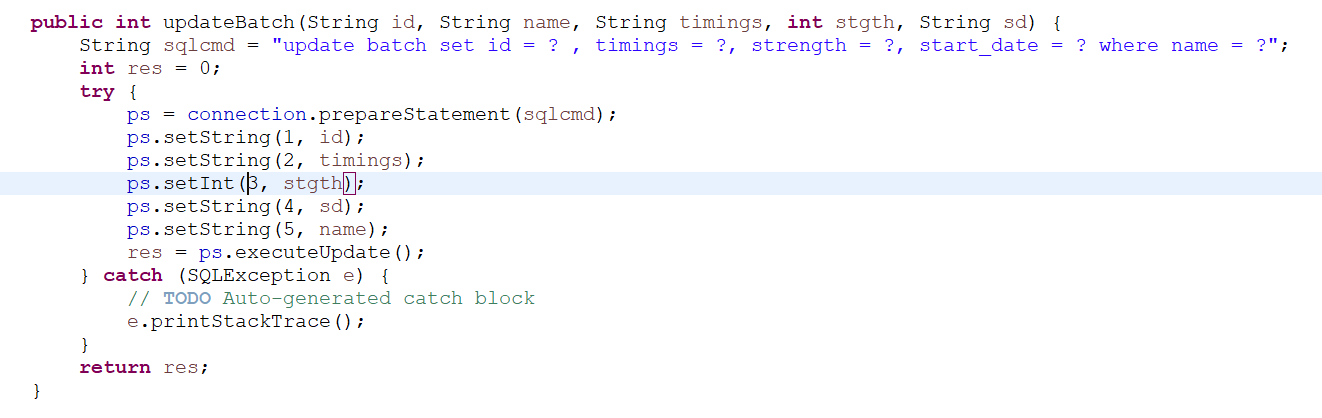
1. Public int addBatch(Batch b): This method adds a new batch into the batch table.



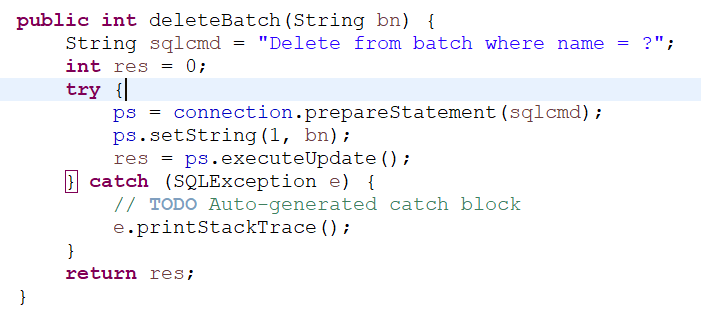
1. public ArrayList<String> fetchAllBatches(): This method is used to fetch the list of batch names from the batch tables.
2. public ArrayList<Batch> fetchBatchDetails(String bn): This method is used to fetch batch details of particular batch name from batch table.



1. public int updateBatch(String id, String name, String timings, int stgth, String sd): This method is used to update batch details in batch table.

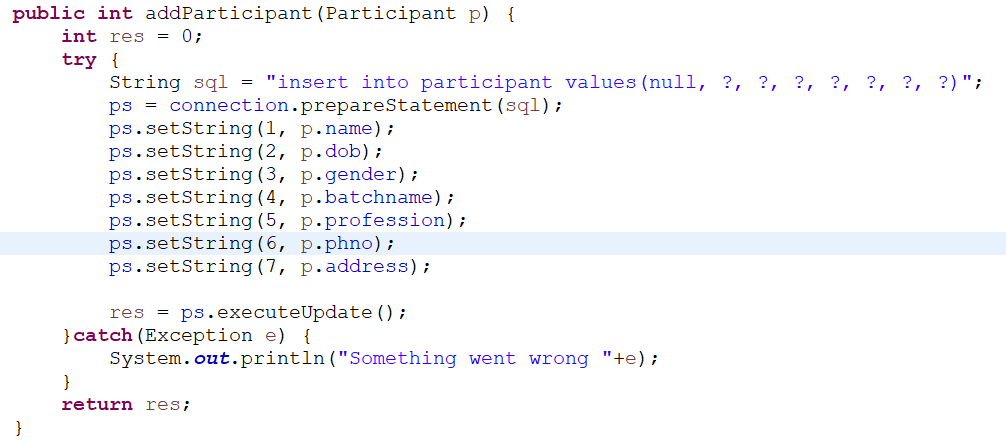


1. public int deleteBatch(String bn): This method is used to delete a batch with particular batch name from batch table.

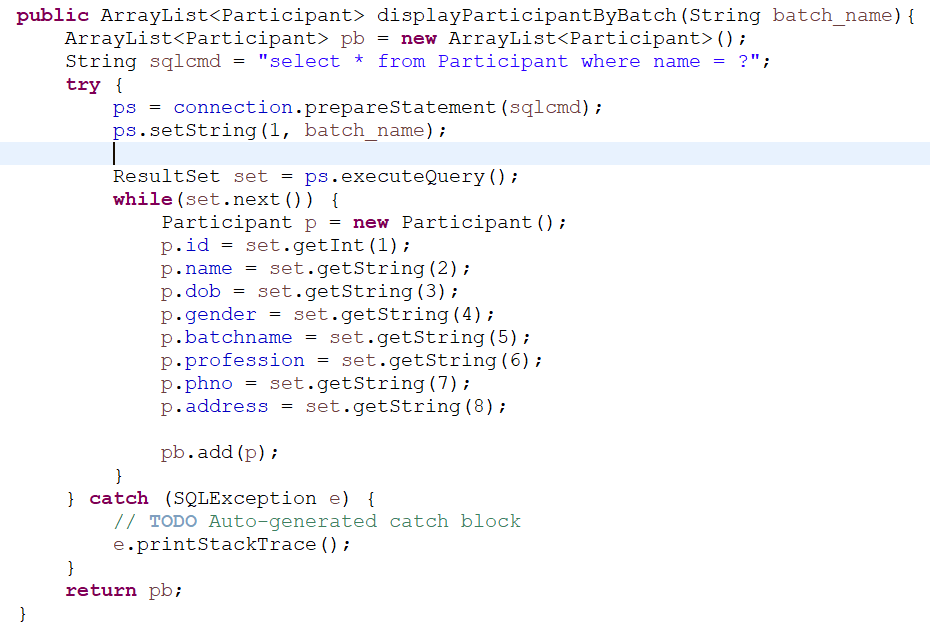


**Methods for participant in Db class:**

1. public int addParticipant(Participant p): This method is used to add participant to participant table.



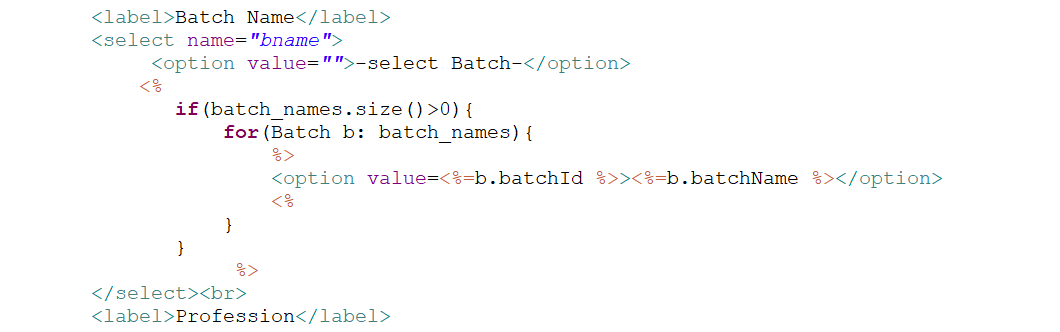
1. public ArrayList<Participant> displayParticipantByBatch(String batch\_name): This method is used to display participants in a particular batch.



1. public ArrayList<Participant> displayParticipantByName(String name): This method is used to display participant details of particular participant name.
2. public ArrayList<Participant> displayParticipantById(int id): This method is used to display participant details of particular participant ID.
3. Public ArrayList<Participant> displayAllParticipantDetails(): This method displays list of all the participants in participant table.
4. public int updateParticipant(int id, String name, String dob, String gender, String batch, String profession, String phno, String addr): This method is used to update participant details in participant table.
5. public int deleteParticipant(int id): This method is used to delete the participant from the participant table.

HTML Form with name “CreateBatch.html” is created to create batch. A servlet with name “BatchServlet.java” is created to process the input form parameters of the CreateBatch.html file and are stored in Batch table in icon database. The form parameters are sent through HTTP POST method.

A JSP File with name “Participant.jsp” is created to register the participants. In this file, a select dropdown menu is designed which has compatability to update batches from the batch table in database which are created using CreateBatch.html. The input form parameters are processed by Register.java servlet. The data is transferred through HTTP Post method.



A SearchBatch.jsp file is created to take input of batch name to display details of batch with that batch name. This jsp file is linked to SearchBatches.java servlet to process the input parameters.

SearchBatches.java

SearchBatch.jsp file

Now, Participants can be sorted in many ways. They can searched to display by batches, display by id and display by name. So, SearchParticipant.html file is created with a select dropdown box which has search parameter options like by batch or by ID or by name. This file is linked to DisplayParticipantDetails.jsp file. This DisplayParticipantDetails is included conditionally with three JSP pages SearchParticipantById.jsp, SearchParticipantByName.html and SearchParticipantByBatch.jsp through @include action tag. DisplayParticipantDetails.jsp file first requests the parameters from SearchParticipant.html and if search parameter option value is ID, it includes SearchParticipantById.jsp and similarly with other two parameters.

SearchParticipantById.jsp

DisplayParti-cipantDetails.jsp

SearchParticipant.ht-ml

SearchParticipantByName.html

SearchParticipantByBatch.jsp

SearchParticipantById.jsp takes participant id as input. SearchPariticipantById.java servlet request values from SearchParticipantById.jsp and responses participant details using displayParticipantById() method of Db class.

SearchPariticipantById.java

SearchParticipantById.jsp

SearchParticipantByName.html takes participant name as input. SearchParticipantByName.java servlet requests values from SearchParticipantByName.html and responses participant details using displayParticipantByName() method in Db class.

SearchPariticipantByName.java

SearchParticipantByName.html

SearchParticipantByBatch.jsp takes batch name as input. DisplayParticipantByBatch.java Servlet requests the input value from SearchParticipantByBatch.jsp and responses participant details using displayParticipantsByBatch() method of Db class.

DisplayPariticipantByBatch.java

SearchParticipantByBatch.jsp

BatchDetails.jsp is created to design a single one stop platform to view, edit and delete batch details. ParticipantDetails.jsp is created to design a single one stop platform to view, edit and delete participant details.

SearchBatches.java

UpdateBatch.java

EditBatch.java

BatchDetails.jsp

CancelBatch.java

SearchBatches.java servlet requests the batch name from BatchDetails.jsp and gets the information of batch using fetchBatchDetails(String name) from Db class.

EditBatch.java servlet requests the batch name from BatchDetails.jsp and displays the edit batch form to input the modified details. UpdateBatch.java request the values from edit batch form of EditBatch.java and updates them in database using UpdateBatch() method of Db class.

CancelBatch.java servlet requests the batch name from BatchDetails.jsp and deletes the batch using deleteBatch() method in Db class.

ParticipantsDetails.java

UpdateParticipant.java

ParticipantDetails.jsp

EditParticipant.java

DeleteParticipant.java

ParticipantsDetails.java servlet requests the participant id from ParticipantDetails.jsp and gets the information of participant using displayParticipantById() method from Db class.

EditParticipant.java servlet requests the participant id from ParticipantDetails.jsp and displays the edit participant form to input the modified details. UpdateParticipant.java request the values from edit participant form of EditParticipant.java and updates them in database using UpdateParticipant() method of Db class.

DeleteParticipant.java servlet requests the participant id from BatchDetails.jsp and deletes the batch using deleteParticipant() method in Db class.

# PROJECT DESCRIPTION:

Batches are created using form in CreateBatch.html. Participants can be registered using form in Participant.jsp where participants are allowed to choose batches which are created using CreateBatch.html.

Gym management will be able to search and view the details of individual participants and also list of participants entirely or by batches. Gym management will be able to view, edit and delete participant details at one place.

Gym management will also be able to view, edit and delete batch details at one place.

Participant will be able to view and edit their details. They can also view batch details.

Gym management

Participant

Participant.jsp

Participant Table

CreateBatch.html

Batch Table

BatchDetails.jsp

Participant Details.jsp

Delete🡪DeleteParticipant.java

Edit🡪EditParticipant.java

Info🡪ParticipantsDetails.java

Delete🡪CancelBatch.java

Edit🡪EditBatch.java

Info🡪SearchBatches.java

# CONCLUSION:

Thus, gym management software will empower the management to have robust performance and efficient functioning. They will be able to perform activities with minimal man power their by driving up their profits.

Display Batch Details:Create Select loop dropdown menu for batches created and search button to submit. Create a JSP for form to take input for batch search. Then link a servlet to this jsp page to display details of selected batch.- Completed

Display List Of Batches – Coding Done

Display List Of Participants

Display List Of Participants by Batch:

select participant.id,participant.participant\_name, participant.profession,participant.phone\_number, batch.name -> from participant Left Join batch on participant.name = batch.name Order By participant\_name;

Display Participant details