

hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
    <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/gymtracking</property>
    <property name="hibernate.connection.username">GymTracking</property>
    <property name="hibernate.connection.password">gym1234</property>
    <property name="show_sql">>true</property>
    <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <!-- Mapping files -->
    <mapping resource="Activity.hbm.xml"/>
    <mapping resource="User.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

User.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="com.hit.model.User" table="USERS">
    <id name="userName" type="java.lang.String">
        <column name="USER_NAME" />
    </id>
    <property name="password" type="java.lang.String">
        <column name="PASSWORD" />
    </property>
    <property name="weight" type="int">
        <column name="WEIGHT" />
    </property>
    <property name="height" type="int">
        <column name="HEIGHT" />
    </property>
</class>
</hibernate-mapping>
```

Activity.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="com.hit.model.Activity" table="ACTIVITY">
        <id name="activityId" type="int">
            <column name="ACTIVITY_ID" />
            <generator class="increment"></generator>
        </id>
        <property name="userName" type="java.lang.String">
            <column name="USER_NAME" />
        </property>
        <property name="activityName" type="java.lang.String">
            <column name="ACTIVITY_NAME" />
        </property>
        <property name="trainingDay" type="java.lang.String">
            <column name="TRAINING_DAY" />
        </property>
        <property name="numberOfSets" type="int">
            <column name="NUMBERS_OF_SETS" />
        </property>
    </class>
</hibernate-mapping>
```

```

        </property>
        <property name="numberOfReps" type="int">
            <column name="NUMBERS_OF_REPS" />
        </property>
    </class>
</hibernate-mapping>

```

com.hit.model

User.java

```

package com.hit.model;

/**
 * User class represent user properties.
 */

public class User {

    private String userName;
    private String password;
    private int weight;
    private int height;

    /**
     * default C'tor
     */
    public User() {
    }

    /**
     * C'tor that initialize this class members with given values.
     * @param userName represent user name.
     * @param password represent password for user.
     * @param weight represent weight for user.
     * @param height represent height for user.
     */
    public User(String userName, String password, int weight, int height){
        setUserName(userName);
        setPassword(password);
        setWeight(weight);
        setHeight(height);
    }

    /**
     * Get user name.
     * @return user name.
     */
    public String getUserName() {
        return userName;
    }

    /**
     * set user name.
     * @param userName to set.
     */
    public void setUserName(String userName) {
        this.userName = userName;
    }

    /**
     * Get password.
     * @return password of user.
     */
    public String getPassword() {

```

```

        return password;
    }

    /**
     * Set password.
     * @param password to set.
     */
    public void setPassword(String password) {
        this.password = password;
    }

    /**
     * Get weight.
     * @return weight of user.
     */
    public int getWeight() {
        return weight;
    }

    /**
     * Set weight.
     * @param weight to set.
     */
    public void setWeight(int weight) {
        this.weight = weight;
    }

    /**
     * Get height.
     * @return height of user.
     */
    public int getHeight() {
        return height;
    }

    /**
     * Set height.
     * @param height of set.
     */
    public void setHeight(int height) {
        this.height = height;
    }

    @Override
    public String toString() {
        return "User [userName=" + userName + ", password=" + password +
            ", weight=" + weight + ", height=" + height + "]" ;
    }
}

```

Activity.java

```

package com.hit.model;

/**
 * User class represent user properties.
 */
public class Activity {

    private String userName;
    private int activityId;
    private String activityName;
    private String trainingDay;
    private int numberOfSets;
}

```

```

private int numberOfReps;

/**
 * default C'tor
 */
public Activity() {
}

/**
 * C'tor that initialize this class members with given values.
 * @param userName represent user name.
 * @param activityName represent activity name.
 * @param trainingDay represent training day.
 * @param numbrtOfSets represent number of sets for activity.
 * @param numberOfReps represent number of repeats for activity.
 */
public Activity(String userName, String activityName, String trainingDay, int
numbrtOfSets, int numberOfReps) {
    setUserName(userName);
    setActivityName(activityName);
    setTrainingDay(trainingDay);
    setNumberOfSets(numbrtOfSets);
    setNumberOfReps(numberOfReps);
}

/**
 * Get username.
 * @return username.
 */
public String getUserName() {
    return userName;
}

/**
 * Set username.
 * @param userName to set.
 */
public void setUserName(String userName) {
    this.userName = userName;
}

/**
 * Get activity id.
 * @return activity id.
 */
public int getActivityId() {
    return activityId;
}

/**
 * Set activity id.
 * @param activityId to set.
 */
public void setActivityId(int activityId) {
    this.activityId = activityId;
}

/**
 * Get activity name.
 * @return activity name.
 */
public String getActivityName() {
    return activityName;
}

```

```

    }

    /**
     * Set activity name.
     * @param activityName to set.
     */
    public void setActivityName(String activityName) {
        this.activityName = activityName;
    }

    /**
     * Get Training Day.
     * @param trainingDay.
     */
    public String getTrainingDay() {
        return trainingDay;
    }

    /**
     * Set Training Day.
     * @param trainingDay to set.
     */
    public void setTrainingDay(String trainingDay) {
        this.trainingDay = trainingDay;
    }

    /**
     * Get number of sets.
     * @return number of activity sets.
     */
    public int getNumberOfSets() {
        return numberOfSets;
    }

    /**
     * Set number of sets.
     * @param numberOfSets to set.
     */
    public void setNumberOfSets(int numberOfSets) {
        this.numberOfSets = numberOfSets;
    }

    /**
     * Get number of repeats.
     * @return number of activity repeats.
     */
    public int getNumberOfReps() {
        return numberOfReps;
    }

    /**
     * Set number of repeats.
     * @param numberOfReps to set.
     */
    public void setNumberOfReps(int numberOfReps) {
        this.numberOfReps = numberOfReps;
    }

    @Override
    public String toString() {
        return "User [UserName="+ userName + " ,activityId=" + activityId + "
,activityName=" + activityName +
        ", numberOfSets=" + numberOfSets + ", numberOfReps=" + numberOfReps
+ "]" ;
    }
}

```

```
}
```

ExceptionGymDAO.java

```
package com.hit.model;

/**
 * Class that represent exception that belong to this project.
 */

public class ExceptionGymDAO extends Exception {

    /**
     * Default serial version id.
     */
    private static final long serialVersionUID = 1L;

    /**
     * Method that send a message that describe the exception to the user.
     * @param message is a string that represent the exception that occurred.
     */
    public ExceptionGymDAO(String message) {
        super("GymDAO Exception: " + message);
    }

    /**
     * An exception wrapper for the Hibernate framework.
     * @param message is a string that represent the exception that has just occured.
     * @param cause is a Throwable that represent error stack.
     */
    public ExceptionGymDAO(String message, Throwable cause) {
        super("GymDAO Exception: " + message, cause);
    }
}
```

IGymDAO.java

```
package com.hit.model;

import java.util.List;

/**
 * IGymDAO is a interface represent the methods for User and Activity.
 */

public interface IGymDAO {

    /**
     * User API function
     */

    /**
     * Add user for DataBase(User table).
     * @param userToAdd represent user to add to user table.
     * @return if user was added successfully.
     * @throws ExceptionGymDAO if adding failed.
     */
    public boolean addUser(User userToAdd) throws ExceptionGymDAO;

    /**
     * Delete user from the DataBase(User table).
     * @param UserNameToDelete represent user that we want to delete.
     * @return if user was deleted successfully.
     * @throws ExceptionGymDAO if delete failed.
     */
}
```

```

    public boolean deleteUser(User UserNameToDelete) throws ExceptionGymDAO;

    /**
     * Update user password,weight and height in the DataBase(User table).
     * @param userToUpdate represent User that we want to update his password,weight and
height.
     * @param passwordToChange represent String with new password.
     * @param weightToChange represent int with new weight.
     * @param heightToChange represent int with new height.
     * @return if user's data was updated successfully.
     * @throws ExceptionGymDAO if update failed.
     */
    public boolean updateUser(User userToUpdate,String passwordToChange,int
weightToChange,int heightToChange) throws ExceptionGymDAO;

    /**
     * Update password password in the DataBase(User table).
     * @param userToUpdate represent User that we want to update his password,weight and
height.
     * @param passwordToChange represent String with new password.
     * @return if user's data was updated successfully.
     * @throws ExceptionGymDAO if update failed.
     */
    public boolean updatePassword(User userToUpdate,String passwordToChange) throws
ExceptionGymDAO;

    /**
     * Update Weight And Height weight and height in the DataBase(User table).
     * @param userToUpdate represent User that we want to update his password,weight and
height.
     * @param weightToChange represent int with new weight.
     * @param heightToChange represent int with new height.
     * @return if user's data was updated successfully.
     * @throws ExceptionGymDAO if update failed.
     */
    public boolean updateWeightAndHeight(User userToUpdate,int weightToChange,int
heightToChange) throws ExceptionGymDAO;

    /**
     * Return all users from the DataBase(User table).
     * @return list of all Users.
     * @throws ExceptionGymDAO if getting the list from DataBase failed.
     */
    public List<User> findAllUsers() throws ExceptionGymDAO;

    /**
     * Return true or false if user is exist.
     * @param user represent user that we want to look for in the User table.
     * @return True or False if user exist.
     * @throws ExceptionGymDAO if user dont found.
     */
    public boolean isUserExist(User user) throws ExceptionGymDAO;

    /**
     * Find user from the DataBase(User table).
     * @param userName represent user that we look for.
     * @return the User if he exist.
     * @throws ExceptionGymDAO if searching failed.
     */
    public User searchUser(String userName) throws ExceptionGymDAO;

    /**
     * return true or false if user is exist in the gymActive list user table.
     * @param userName is a string that represent user name.
     * @param password is a string that represent password.

```

```

    * @return true or false if user is exist.
    * @throws ExceptionGymDAO if searching failed.
    */
    public boolean searchUserBool(String userName,String password) throws ExceptionGymDAO;

    /**
    * Return true or false if login successful.
    * @param userName represent user that we want to look for in the User table.
    * @param password represent a password for user.
    * @return True or False if login successful.
    * @throws ExceptionGymDAO if login failed.
    */
    public boolean login(String userName,String password) throws ExceptionGymDAO;

    /**
    * Return activity from DataBase(Activity table) ,with user's activity.
    * @param userId represent user that we want to get his activity form the DataBase.
    * @return list of user's activity.
    * @throws ExceptionGymDAO if getting user's activity from DataBase failed.
    */
    public List<Activity> findUserActivity(User user) throws ExceptionGymDAO;

    /**
    * Activity API function
    */

    /**
    * Add Activity to the DataBase(Activity table).
    * @param activityToAdd represent activity that we want to add.
    * @return if activity was added successfully.
    * @throws ExceptionGymDAO if adding failed.
    */
    public boolean addActivity(Activity activityToAdd) throws ExceptionGymDAO;

    /**
    * Delete Activity from the DataBase(Activity table).
    * @param activityToDelete represent activity that we want to delete.
    * @return if activity was deleted successfully.
    * @throws ExceptionGymDAO if the delete failed.
    */
    public boolean deleteActivity(Activity activityToDelete) throws ExceptionGymDAO;

    /**
    * Update activity in the Activity table.
    * @param activityToUpdate represent activity that we want to update.
    * @param changeSets is a int that represent the new sets.
    * @param changeReps is a int that represent the new repeats.
    * @param changeTrainingDay is a string that represent the new day.
    * @return if activity's data was updated successfully.
    * @throws ExceptionGymDAO if update failed.
    */
    public boolean updateActivity(Activity activityToUpdate,String changeTrainingDay, int
changeSets,int changeReps) throws ExceptionGymDAO;

    /**
    * Return true or false if user is activity.
    * @param activity represent activity that we want to look for in the Activity table.
    * @return True or False if activity exist.
    * @throws ExceptionGymDAO if activity dont found.
    */
    public boolean isActivityExist(Activity activity) throws ExceptionGymDAO;

```



```

/**
 * Return list from DataBase(Activity table) with all activity.
 * @return list of all activity.
 * @throws ExceptionGymDAO if getting all activity from DataBase failed.
 */
public List<Activity> findAllActivity() throws ExceptionGymDAO;

/**
 * Search activity in the DataBase(Activity table).
 * @param id represent id of activity that we want to look for.
 * @return activity that we look for by id.
 * @throws ExceptionGymDAO if searching failed.
 */
public Activity searchActivity(int id) throws ExceptionGymDAO;
}

```

HibernateGymDAO.java

```
package com.hit.model;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.hibernate.HibernateException;
```

```
import org.hibernate.Query;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
import org.hibernate.Transaction;
```

```
import org.hibernate.cfg.Configuration;
```

```

/**
 * HibernateGymDAO - implements IGymDAO interface, represent the methods for users and activity action.
 */

```

```
public class HibernateGymDAO implements IGymDAO {
```

```
    private static final HibernateGymDAO INSTANCE = new HibernateGymDAO();
```

```
    private SessionFactory factory;
```

```
    private HibernateGymDAO()
```

```
{
```

```
    if(INSTANCE != null){
```

```
        throw new IllegalStateException("HibernateGymDAO was already instantiated");
```

```

    }
    try {
        Configuration configuration = new Configuration().configure();
        factory = configuration.buildSessionFactory();
    }
    catch(Exception e) {
        System.out.println("HibernateGymDAO Error: "+e.getMessage());
    }
}

public static HibernateGymDAO getInstance()
{
    return INSTANCE;
}

/**
 *Checks whether a user has been added successfully
 */
@Override
public boolean addUser(User userToAdd) throws ExceptionGymDAO {
    Session session = null;
    boolean result = false;
    Transaction transaction = null;

    if(!isUserExist(userToAdd)){
        try {
            session = factory.openSession();
            transaction = session.beginTransaction();
            session.save(userToAdd);
            transaction.commit();
            result = true;
        }
        catch(HibernateException exception) {
            rollbackTrasaction(transaction);
            throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
        }
    }
}

```

```

        }
        finally {
            closeSession(session);
        }
    }
    return result;
}

/**
 * Checks whether user deletion was successful
 */
@Override
public boolean deleteUser(User UserNameToDelete) throws ExceptionGymDAO {
    Session session = null;
    boolean result = false;
    Transaction transaction = null;

    if (isUserExist(UserNameToDelete)) {
try {
    session = factory.openSession();
    transaction = session.beginTransaction();
    List<Activity> listOfActivities = findUserActivity(UserNameToDelete);
    for (Activity activity : listOfActivities) {
        session.delete(activity);
    }
    session.delete(UserNameToDelete);
    transaction.commit();
    result = true;
}
catch(HibernateException exception) {
    rollbackTrasaction(transaction);
    throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
}
finally {
    closeSession(session);
}

```

```

        }

return result;

    }

    /**
     * Checks whether a user(password&weight&height) update was successful
     */
    @Override
    public boolean updateUser(User userToUpdate, String passwordToChange, int weightToChange, int
heightToChange)
        throws ExceptionGymDAO {
        Session session = null;
        boolean result = false;
        Transaction transaction = null;

        if (isUserExist(userToUpdate)) {
try {
    session = factory.openSession();
    transaction = session.beginTransaction();
    userToUpdate.setPassword(passwordToChange);
    userToUpdate.setWeight(weightToChange);
    userToUpdate.setHeight(heightToChange);
    session.update(userToUpdate);
    transaction.commit();
    result = true;
}
catch(HibernateException exception) {
    rollbackTrasaction(transaction);
    throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
}
finally {
    closeSession(session);
}

        }

return result;

```

```

    }

    /**
     * Checks whether the user password update was successful
     */
    @Override
    public boolean updatePassword(User userToUpdate,String passwordToChange) throws ExceptionGymDAO
    {
        Session session = null;
        boolean result = false;
        Transaction transaction = null;

        if (isUserExist(userToUpdate)) {
        try {
            session = factory.openSession();
            transaction = session.beginTransaction();
            userToUpdate.setPassword(passwordToChange);
            session.update(userToUpdate);
            transaction.commit();
            result = true;
        }
        catch(HibernateException exception) {
            rollbackTrasaction(transaction);
            throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
        }
        finally {
            closeSession(session);
        }
        }

        return result;
    }

    /**
     * Checks whether the user weight&height update was successful
     */
    @Override

```

```

        public boolean updateWeightAndHeight(User userToUpdate,int weightToChange,int heightToChange)
        throws ExceptionGymDAO{

            Session session = null;

            boolean result = false;

            Transaction transaction = null;

            if (isUserExist(userToUpdate)) {
try {
            session = factory.openSession();

            transaction = session.beginTransaction();

            userToUpdate.setWeight(weightToChange);

            userToUpdate.setHeight(heightToChange);

            session.update(userToUpdate);

            transaction.commit();

            result = true;
        }
        catch(HibernateException exception) {
            rollbackTrasaction(transaction);

            throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
        }
        finally {
            closeSession(session);
        }
    }

    return result;
}

/**
 * Returns a list of all users
 */
@Override
public List<User> findAllUsers() throws ExceptionGymDAO {

    Session session = null;

    Transaction transaction = null;

    List users = new ArrayList<User>();

```

```

        try {
            session = factory.openSession();
            transaction = session.beginTransaction();
            users = session.createQuery("from User").list();
            transaction.commit();
        }

catch(HibernateException exception) {
    rollbackTrasaction(transaction);
    throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
}

finally {
    closeSession(session);
}

    return users;
}

/**
 * Checks if the user exists
 */
@Override
public boolean isUserExist(User user) throws ExceptionGymDAO {
    Session session = null;
    boolean exist = false;

    try {
        session = factory.openSession();
        Object object = session.get(user.getClass(), user.getUserName());
        exist = object != null ;
    }

    catch(HibernateException exception) {
        throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
    }

    finally {
        closeSession(session);
    }

    return exist;
}

```

```

}

/**
 * Searching for user by username
 */
@Override
public User searchUser(String userName) throws ExceptionGymDAO {
    Session session = null;
    Transaction transaction = null;
    User user = null ;
    try {
        session = factory.openSession();
transaction = session.beginTransaction();
user = (User)session.get(User.class, userName);
if(user != null) {
    transaction.commit();
}
    }
    catch(HibernateException exception) {
rollbackTrasaction(transaction);
throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
}
    finally {
        closeSession(session);
    }

    return user;
}

/**
 * Checks if the password is correct for the correct user
 */
@Override
public boolean searchUserBool(String userName,String password) throws ExceptionGymDAO{
    boolean userExist = false;
    User user = searchUser(userName);

```



```

        if(user != null) {
            if(user.getPassword().equals(password)) {
                userExist = true;
            }
        }
        return userExist;
    }
}

```

```

/**
 * Checks integrity of login
 */
@Override
public boolean login(String userName, String password) throws ExceptionGymDAO {
    Session session = null;
    Transaction transaction = null;

    try {
        session = factory.openSession();
        transaction = session.beginTransaction();

        Query query = session.createQuery("from User user where user.userName= :userName AND user.password= :password");
        query.setParameter("userName", userName);
        query.setParameter("password", password);
        transaction.commit();
        List<User> users = query.list();
        return (users != null && !users.isEmpty());
    }
    catch(HibernateException exception) {
        rollbackTrasaction(transaction);
        throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
    }
    finally {
        closeSession(session);
    }
}

```

```

/**
 * Returns a list of all user activities
 */
@Override
public List<Activity> findUserActivity(User user) throws ExceptionGymDAO {
    Session session = null;
    Transaction transaction = null;
    List<Activity> listOfActivities = new ArrayList<Activity>();

    if(isUserExist(user)) {
        try {
            session=factory.openSession();
            transaction = session.beginTransaction();
            Query query = session.createQuery("From Activity WHERE userName = :userName
");
            query.setParameter("userName", user.getUserName() );
            listOfActivities = query.list();
        }
        catch(HibernateException exception) {
            rollbackTrasaction(transaction);
            throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
        }
        finally {
            closeSession(session);
        }
    }

    return listOfActivities;
}

/**
 * Checks whether a new activity added successful
 */
@Override
public boolean addActivity(Activity activityToAdd) throws ExceptionGymDAO {

```

```

        Session session = null;

        boolean result = false;

        Transaction transaction = null;

        if(!isActivityExist(activityToAdd)) {
            try {
                session = factory.openSession();

                transaction = session.beginTransaction();

                session.save(activityToAdd);

                transaction.commit();

                result = true;
            }
            catch(HibernateException exception) {
                rollbackTrasaction(transaction);

                throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
            }
            finally {
                closeSession(session);
            }
        }

        return result;
    }

    /**
     * Checks whether activity deleted successful
     */
    @Override
    public boolean deleteActivity(Activity activityToDelete) throws ExceptionGymDAO {
        Session session = null;

        boolean result = false;

        Transaction transaction = null;

        if (isActivityExist(activityToDelete)) {
            try {
                session = factory.openSession();

                transaction = session.beginTransaction();

```

```

        session.delete(activityToDelete);

        transaction.commit();

        result = true;
    }

    catch(HibernateException exception) {

        rollbackTrasaction(transaction);

        throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
    }

    finally {

        closeSession(session);
    }

    }

return result;

    }

/**
 * Checks whether an activity(trainingDay,sets, reps) update was successful
 */
@Override

    public boolean updateActivity(Activity activityToUpdate,String changeTrainingDay, int changeSets, int
changeReps) throws ExceptionGymDAO {

        Session session = null;

        boolean result = false;

        Transaction transaction = null;

        if (isActivityExist(activityToUpdate)) {

try {

        session = factory.openSession();

        transaction = session.beginTransaction();

        activityToUpdate.setTrainingDay(changeTrainingDay);

        activityToUpdate.setNumberOfSets(changeSets);

        activityToUpdate.setNumberOfReps(changeReps);

        session.update(activityToUpdate);

        transaction.commit();

        result = true;

    }

```

```

        catch(HibernateException exception) {
            rollbackTrasaction(transaction);
            throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
        }
        finally {
            closeSession(session);
        }
    }

    return result;
}

/**
 * Checks whether the activity exists
 */
@Override
public boolean isActivityExist(Activity activity) throws ExceptionGymDAO {
    Session session = null;
    boolean exist = false;
    Object object = null;

    try {
        session = factory.openSession();
        object = session.get(activity.getClass(), activity.getActivityId());
        exist = object != null;
    }
    catch(HibernateException exception) {
        throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
    }
    finally {
        closeSession(session);
    }

    return exist;
}

/**
 * Returns a list of all activities

```

```
*/
```

```
@Override
```

```
public List<Activity> findAllActivity() throws ExceptionGymDAO {
```

```
    Session session = null;
```

```
    Transaction transaction = null;
```

```
    List activitys = new ArrayList<Activity>();
```

```
    try {
```

```
        session = factory.openSession();
```

```
        transaction = session.beginTransaction();
```

```
        activitys = session.createQuery("from Activity").list();
```

```
        transaction.commit();
```

```
    }
```

```
catch(HibernateException exception) {
```

```
    rollbackTrasaction(transaction);
```

```
    throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
```

```
}
```

```
finally {
```

```
    closeSession(session);
```

```
}
```

```
    return activitys;
```

```
}
```

```
/**
```

```
 * searches for activity by ID
```

```
*/
```

```
@Override
```

```
public Activity searchActivity(int id) throws ExceptionGymDAO {
```

```
    Session session = null;
```

```
    Transaction transaction = null;
```

```
    Activity activity = null;
```

```
    try {
```

```
        session = factory.openSession();
```

```
        transaction = session.beginTransaction();
```

```
        activity = (Activity)session.get(Activity.class, id);
```

```
    }
```

```

catch(HibernateException exception) {
    rollbackTrasaction(transaction);
    throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
}
finally {
    closeSession(session);
}

    return activity;
}

/**
 * roll back Trasaction
 * @param transaction
 */
private void rollbackTrasaction(Transaction transaction) {
    if(transaction != null) {
        transaction.rollback();
    }
}

/**
 * close Session
 * @param session
 * @throws ExceptionGymDAO
 */
private void closeSession(Session session) throws ExceptionGymDAO {
    if(session != null && session.isOpen()) {
        try {
            session.close();
        }
        catch(HibernateException exception) {
            throw new ExceptionGymDAO(exception.getMessage(), exception.getCause());
        }
    }
}

```

```
}
```

HibernateGymDAOTest.java

```
package com.hit.model;

import java.util.List;

/**
 * Testing done to IGymDAO & HibernateGymDAO
 * @param args
 * @throws ExceptionGymDAO
 */
public class HibernateGymDAOTest {

    public static void main(String[] args) throws ExceptionGymDAO {
        User user1 = new User("AlexeySh", "a12345", 72, 177);
        System.out.println(user1.getUserName());
        Activity activ1 = new Activity("AlexeySh", "benst", "Monday", 3, 8);
        Activity activ2 = new Activity("AlexeySh", "legs", "Sunday", 4, 7);
        HibernateGymDAO hibernate = HibernateGymDAO.getInstance();
        hibernate.addUser(user1);
        hibernate.addActivity(activ1);
        hibernate.addActivity(activ2);

        List<Activity> usersActivity = hibernate.findUserActivity(user1);
        System.out.println(user1);
        for(Activity activity : usersActivity)
        {
            System.out.println(activity.toString());
        }

        System.out.println(123);
    }
}
```

com.hit.controller

GymTrackingController.java

```
package com.hit.controller;

import java.io.IOException;
import java.util.Iterator;
import java.util.Iterator;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
```



```
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
import com.hit.model.Activity;
```

```
import com.hit.model.ExceptionGymDAO;
```

```
import com.hit.model.HibernateGymDAO;
```

```
import com.hit.model.IGymDAO;
```

```
import com.hit.model.User;
```

```
/**
```

```
 * Servlet implementation class GymTrackingController
```

```
 */
```

```
@WebServlet("/GymTrackingController/*")
```

```
public class GymTrackingController extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private IGymDAO gymTracking = HibernateGymDAO.getInstance();
```

```
/**
```

```
 * @see HttpServlet#HttpServlet()
```

```
 */
```

```
public GymTrackingController() {
```

```
    super();
```

```
}
```

```
/**
```

```
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
```

```
 */
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {
```

```
    String path = request.getPathInfo();
```

```
    String theCommand = request.getParameter("command");
```

```
    List<Activity> userActivity = null;
```

```
    //RequestDispatcher dispatcher = null;
```

```
    User newUser = null;
```

```
    HttpSession session = request.getSession();
```

```
    String activityID,newTrainingDay;
```

```
int intActivityID,newNumberOfSets,newNumberOfReps;
```

```
switch(theCommand)
```

```
{
```

```
//delete activity
```

```
case "delete":
```

```
    activityID = request.getParameter("activityId");
```

```
    intActivityID = Integer.parseInt(activityID);
```

```
    try {
```

```
        Activity activityToDelete = gymTracking.searchActivity(intActivityID);
```

```
        gymTracking.deleteActivity(activityToDelete);
```

```
        newUser = (User)session.getAttribute("loginUser");
```

```
        userActivity = gymTracking.findUserActivity(newUser);
```

```
        session.setAttribute("userActivity" , userActivity);
```

```
        request.setAttribute("message", "YOUR ACTIVITY HAS BEEN SUCCEFULLY  
DELETED!");
```

```
        request.getRequestDispatcher("/userActivities.jsp").forward(request,response);
```

```
    }
```

```
    catch (ExceptionGymDAO e) {
```

```
        e.printStackTrace();
```

```
        request.getRequestDispatcher("/error.jsp").forward(request, response);
```

```
    }
```

```
    break;
```

```
//send activity id to update
```

```
case "sendToUpdate":
```

```
    activityID = request.getParameter("activityId");
```

```
    session.setAttribute("activityId", activityID);
```

```
    request.getRequestDispatcher("/updateActivity.jsp").forward(request, response);
```

```
    break;
```

```
//update activity
```

```
case "updateActivity":
```

```

        newTrainingDay = request.getParameter("myRadio");
        newNumberOfSets = Integer.parseInt(request.getParameter("newNumberOfSets"));
        newNumberOfReps = Integer.parseInt(request.getParameter("newNumberOfReps"));
        intActivityID = Integer.parseInt((String)session.getAttribute("activityId"));

        try {
            newUser = (User) session.getAttribute("loginUser");
            Activity activityToUpdate = gymTracking.searchActivity(intActivityID);

            gymTracking.updateActivity(activityToUpdate,newTrainingDay,newNumberOfSets, newNumberOfReps);

            userActivity = gymTracking.findUserActivity(newUser);
            session.setAttribute("userActivity", userActivity);
            request.setAttribute("message", "YOUR ACTIVITY HAS BEEN SUCCEFULLY
UPDATED!");

            request.getRequestDispatcher("/userActivities.jsp").forward(request, response);

        }
        catch (ExceptionGymDAO e) {
            e.printStackTrace();
            request.getRequestDispatcher("/error.jsp").forward(request, response);
        }

        break;

//logout
case "logout":
    session.invalidate();
    request.setAttribute("message", "LOGOUT SUCCESSFUL!");
    request.getRequestDispatcher("/logout.jsp").forward(request, response);
    break;

}

//response.getWriter().append("Served at: ").append(request.getContextPath());
}

/**

```

```
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
```

```
*/
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {
```

```
    String userName,password,repPassword;
```

```
    String oldPassword,newPassword,repPassword;
```

```
    String activityName,trainingDay;
```

```
    int weight,height;
```

```
    int numberOfSets,numberOfReps;
```

```
    int newWeight,newHeight;
```

```
    List<Activity> userActivity = null;
```

```
    User newUser = null;
```

```
    String theCommand = request.getParameter("command");
```

```
    HttpSession session = request.getSession();
```

```
    switch(theCommand)
```

```
    {
```

```
        //login
```

```
        case "login":
```

```
            userName = request.getParameter("userName");
```

```
            password = request.getParameter("password");
```

```
            try {
```

```
                if(userName.equals("") || password.equals("")) {
```

```
                    request.setAttribute("message","You forgot to fill password/username");
```

```
                    request.getRequestDispatcher("/login.jsp").forward(request, response);
```

```
                }
```

```
                else if(password.length() < 6) {
```

```
                    request.setAttribute("message","The Password less than 6 characters");
```

```
                    request.getRequestDispatcher("/login.jsp").forward(request, response);
```

```
                }
```

```
                boolean userExist = gymTracking.searchUserBool(userName,password);
```

```
                if(userExist) {
```

```
                    User user = gymTracking.searchUser(userName);
```

```
                    userActivity = gymTracking.findUserActivity(user);
```

```
                    session.setAttribute("userActivity", userActivity);
```

```

        session.setAttribute("loginUser", user);
        request.getRequestDispatcher("/userMain.jsp").forward(request,
response);
    }
    else {
        request.setAttribute("message", "User/Password incorrect");
        request.getRequestDispatcher("/login.jsp").forward(request, response);
    }
}

catch (ExceptionGymDAO e) {
    e.printStackTrace();
    request.getRequestDispatcher("/error.jsp").forward(request, response);
}

break;

```

//registration

case "registration":

```

    Iterator<User> iterator = null;
    userName = request.getParameter("userName");
    password = request.getParameter("password");
    rePassword = request.getParameter("rePassword");
    weight = Integer.parseInt(request.getParameter("weight"));
    height = Integer.parseInt(request.getParameter("height"));

    if(password.length() < 6) {
        request.setAttribute("message", "The Password less than 6 characters");
        request.getRequestDispatcher("/registration.jsp").forward(request, response);
    }

    else if(!(password.equals(rePassword))) {
        request.setAttribute("message", "Passwords are not the same, please try again");
        request.getRequestDispatcher("/registration.jsp").forward(request, response);
    }

    else if (password.equals("") || userName.equals("")) {
        request.setAttribute("message", "You need to fill both User name and Password");
        request.getRequestDispatcher("/registration.jsp").forward(request, response);
    }

```

```

        }
    else {
        try {
            User user = gymTracking.searchUser(userName);
            if(user != null) {
                request.setAttribute("message","The username is already exist");

request.getRequestDispatcher("/registration.jsp").forward(request, response);

            }
            else {
                user = new User(userName,password,weight,height);
                gymTracking.addUser(user);
                session.setAttribute("loginUser", user);
                request.setAttribute("message", "WELCOME TO OUR GymTracking
WEB! PLEASE ENTER YOUR DETAILS");

                request.getRequestDispatcher("/login.jsp").forward(request,
response);

            }
        }
        catch (ExceptionGymDAO e) {
            e.printStackTrace();
            request.getRequestDispatcher("/error.jsp").forward(request, response);
        }
    }
}

```

break;

//update user when he change password, weight and height.

case "updateUser" :

```

    newUser = (User)session.getAttribute("loginUser");
    oldPassword = request.getParameter("oldPassword");
    newPassword = request.getParameter("newPassword");
    repPassword = request.getParameter("rePassword");
    newWeight = Integer.parseInt(request.getParameter("newWeight"));
    newHeight = Integer.parseInt(request.getParameter("newHeight"));

    if((oldPassword == "") || (newPassword == "") || (repPassword == "")) {

```

```

        request.setAttribute("message", "Your need to fill all the fields");
        request.getRequestDispatcher("/updateUser.jsp").forward(request, response);
    }

    if((oldPassword == "") || (newPassword == "") || (repPassword == "")) {
        request.setAttribute("message", "Your need to fill all the fields");
        request.getRequestDispatcher("/updateUser.jsp").forward(request, response);
    }

    else if(oldPassword.length()<6 || newPassword.length() <6) {
        request.setAttribute("message", "The Passwords less than 6 characters");
        request.getRequestDispatcher("/updateUser.jsp").forward(request, response);
    }

    else if (!(oldPassword.equals(newUser.getPassword()))) {
        request.setAttribute("message", "Your old password is incorrect");
        request.getRequestDispatcher("/updateUser.jsp").forward(request, response);

    }

    else if(!(newPassword.equals(repPassword))) {
        request.setAttribute("message", "Your password repeat dosent match to your
new password");

        request.getRequestDispatcher("/updateUser.jsp").forward(request, response);
    }

    else if(newPassword.equals(newUser.getPassword())) {
        request.setAttribute("message", "You entered your current password, please
enter a different one");

        request.getRequestDispatcher("/updateUser.jsp").forward(request, response);
    }

    else {
        try {
            gymTracking.updateUser(newUser,newPassword,newWeight,newHeight);
            request.setAttribute("message", "YOUR PASSWORD & WEIGHT & HEIGHT
SUCCEFULLY CHANGED!");

            request.getRequestDispatcher("/userMain.jsp").forward(request,
response);

        }

        catch (ExceptionGymDAO e) {
            e.printStackTrace();
            request.getRequestDispatcher("/error.jsp").forward(request, response);
        }
    }

```

```
}
```

```
break;
```

```
//add new activity
```

```
case "addActive" :
```

```
    newUser = (User)session.getAttribute("loginUser");
```

```
    activityName = request.getParameter("mySelect");
```

```
    trainingDay = request.getParameter("myRadio");
```

```
    numberOfSets = Integer.parseInt(request.getParameter("numberOfSets"));
```

```
    numberOfReps = Integer.parseInt(request.getParameter("numberOfReps"));
```

```
    try {
```

```
        List<Activity> userActivityCheck= gymTracking.findUserActivity(newUser);
```

```
    boolean flag = true;
```

```
    for(Activity activity : userActivityCheck) {
```

```
        String nameActivityTOCheck = activity.getActivityName();
```

```
        String dayCheck = activity.getTrainingDay();
```

```
        if(nameActivityTOCheck.equals(activityName))
```

```
        {
```

```
            if(dayCheck.equals(trainingDay))
```

```
            {
```

```
                flag = false;
```

```
            }
```

```
        }
```

```
    }
```

```
    if(flag) {
```

```
        Activity newActivity = new
```

```
Activity(newUser.getUserName(),activityName,trainingDay,numberOfSets,numberOfReps);
```

```
        gymTracking.addActivity(newActivity);
```

```
        userActivity = gymTracking.findUserActivity(newUser);
```

```
        session.setAttribute("userActivity", userActivity);
```

```
        request.setAttribute("message", "ACTIVITY SUCCEFULLY ADDED!");
```

```
        request.getRequestDispatcher("/userActivities.jsp").forward(request, response);
```

```
    }
```

```
    else {
```



```

        request.setAttribute("message", "You have this activity in your Training
Table!");

        request.getRequestDispatcher("/addActive.jsp").forward(request,
response);

    }

}

catch (ExceptionGymDAO e) {
    e.printStackTrace();
    request.getRequestDispatcher("/error.jsp").forward(request, response);
}

break;

//when user change password
case "updatePassword" :
    newUser = (User)session.getAttribute("loginUser");
    oldPassword = request.getParameter("oldPassword");
    newPassword = request.getParameter("newPassword");
    repPassword = request.getParameter("rePassword");

    if((oldPassword == "") || (newPassword == "") || (repPassword == "")) {
        request.setAttribute("message", "Your need to fill all the fields");
        request.getRequestDispatcher("/updatePassword.jsp").forward(request,
response);
    }

    else if(oldPassword.length()<6 || newPassword.length() <6) {
        request.setAttribute("message", "The Passwords less than 6 characters");
        request.getRequestDispatcher("/updatePassword.jsp").forward(request,
response);
    }

    else if (!(oldPassword.equals(newUser.getPassword()))) {
        request.setAttribute("message", "Your old password is incorrect");
        request.getRequestDispatcher("/updatePassword.jsp").forward(request,
response);
    }

    else if (!(newPassword.equals(repPassword))) {
        request.setAttribute("message", "Your password repeat dosent match to your
new password");

```

```

        request.getRequestDispatcher("/updatePassword.jsp").forward(request,
response);
    }
    else if(newPassword.equals(newUser.getPassword())) {
        request.setAttribute("message", "You entered your current password, please
enter a different one");
        request.getRequestDispatcher("/updatePassword.jsp").forward(request,
response);
    }
    else {
        try {
            gymTracking.updatePassword(newUser, newPassword);
            request.setAttribute("message", "YOUR PASSWORD SUCCSEFULLY
CHANGED!");
            request.getRequestDispatcher("/userMain.jsp").forward(request,
response);
        }
        catch (ExceptionGymDAO e) {
            e.printStackTrace();
            request.getRequestDispatcher("/error.jsp").forward(request, response);
        }
    }

    break;

//user change weight and height
case "updateWeightAndHeight" :
    newUser = (User)session.getAttribute("loginUser");
    newWeight = Integer.parseInt(request.getParameter("newWeight"));
    newHeight = Integer.parseInt(request.getParameter("newHeight"));

    try {
        gymTracking.updateWeightAndHeight(newUser, newWeight, newHeight);
        request.setAttribute("message", "YOUR WEIGHT & HEIGHT SUCCSEFULLY
CHANGED!");
        request.getRequestDispatcher("/userMain.jsp").forward(request, response);
    }
    catch (ExceptionGymDAO e) {

```

```

        e.printStackTrace();
        request.getRequestDispatcher("/error.jsp").forward(request, response);
    }

    break;

    }

    //doGet(request, response);
}
}

```

ServletListenerCounter.java

```

package com.hit.controller;

import javax.servlet.annotation.WebListener;
import javax.servlet.http.HttpSessionEvent;
import javax.servlet.http.HttpSessionListener;

/**
 * Application Lifecycle Listener implementation class ServletListenerCounter
 *
 */
@WebListener
public class ServletListenerCounter implements HttpSessionListener {

    private static int sessionCounter = 0;

    public static int getSessionCounter() {
        return sessionCounter;
    }

    /**
     * @see HttpSessionListener#sessionCreated(HttpSessionEvent)
     */
    @Override
    public void sessionCreated(HttpSessionEvent se) {

```

```

        sessionCounter++;
    }

    /**
     * @see HttpSessionListener#sessionDestroyed(HttpSessionEvent)
     */
    @Override
    public void sessionDestroyed(HttpSessionEvent se) {
        sessionCounter--;
    }
}

```

com.hit.tag
HelloUserTag.java

```
package com.hit.tag;
```

```

import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import java.io.*;

```

```
public class HelloUserTag extends SimpleTagSupport {
```

```
    private String nameOfUser;
```

```

    /**
     * Get name of user
     * @return name of user
     */
    public String getNameOfUser() {
        return nameOfUser;
    }

```

```

    /**
     * Set name of user
     * @param nameOfUser of set

```

```

    */

    public void setNameOfUser(String nameOfUser) {

        this.nameOfUser = nameOfUser;

    }

    /**
     * prints the name of the user if he connected
     */
    public void doTag() throws JspException, IOException {

        JspWriter out = getJspContext().getOut();

        if(nameOfUser != null)

        {

            out.println("Hello and welcome!" + nameOfUser);

        }

        else

        {

            out.println("Hello,you are not logged yet");

        }

    }

}

```

com.hit.wp.model
BGBean.java

```

package com.hit.wp.model;

public class BGBean {
    int r,g,p;

    /**
     * get color r
     * @return r color
     */
    public int getR() {
        return r;
    }

    /**
     * set color r
     * @param r to set
     */
    public void setR(int r) {
        this.r=r;
    }

    /**
     * get color g
     * @return g color
     */
    public int getG() {

```

```

        return g;
    }

    /**
     * set color g
     * @param g to set
     */
    public void setG(int g) {
        this.g=g;
    }

    /**
     * get color p
     * @return p color
     */
    public int getP() {
        return r;
    }

    /**
     * set color p
     * @param p to set
     */
    public void setP(int p) {
        this.p=p;
    }

    public BGBean() {
        r=255;
        g=255;
        p=255;
    }
}

```

WebContent

styleForTable.css

```

table {
    width: 100%;
    border-collapse: collapse;
}
/* Zebra striping */
tr:nth-of-type(odd) {
    background: #eee;
}
th {
    background: #333;
    color: white;
    font-weight: bold;
}
td, th {
    padding: 6px;
    border: 1px solid #ccc;
    text-align: left;
}
@media
only screen and (max-width: 760px),
(min-device-width: 768px) and (max-device-width: 1024px) {

    /* Force table to not be like tables anymore */
    table, thead, tbody, th, td, tr {
        display: block;
    }
}

```

```

/* Hide table headers (but not display: none;, for accessibility) */
thead tr {
    position: absolute;
    top: -9999px;
    left: -9999px;
}

tr { border: 1px solid #ccc; }

td {
    /* Behave like a "row" */
    border: none;
    border-bottom: 1px solid #eee;
    position: relative;
    padding-left: 50%;
}

td:before {
    /* Now like a table header */
    position: absolute;
    /* Top/left values mimic padding */
    top: 6px;
    left: 6px;
    width: 45%;
    padding-right: 10px;
    white-space: nowrap;
}

/*
Label the data
*/
td:nth-of-type(1):before { content: "Type"; }
td:nth-of-type(2):before { content: "Sets"; }
td:nth-of-type(3):before { content: "Reps"; }
td:nth-of-type(4):before { content: "Day"; }
td:nth-of-type(5):before { content: ""; }
td:nth-of-type(6):before { content: ""; }
}

```

tag.tld

```

<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-jsp.taglibrary_2_0.xsd"
    version="2.0">

    <tlib-version>1.0</tlib-version>
    <short-name>tag</short-name>

    <tag>
        <description>My Hello Tag</description>
        <name>helloTag</name>
        <tag-class>com.hit.tag.HelloUserTag</tag-class>
        <body-content>empty</body-content>
        <attribute>
            <description>name of user</description>
            <name>nameOfUser</name>
            <required>false</required>
        </attribute>
    </tag>

</taglib>

```

login.jsp

```
<%@ page language="java" contentType="text/html; charset=windows-1255" pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
  <script src="js/view/popper/popper.min.js"></script>
  <script src="js/view/bootstrap/bootstrap.min.js"></script>
  <title>Login</title>
</head>

<body>
<!-- Content of the login --%>
<div class="container" align="center">
  <div>
    <h1>Sign In</h1>
  </div>
  <div>
    <!-- Form of login --%>
    <form action="/WebGymTracking/GymTrackingController" method="post">
      <br>
      
      <h4 style="color:red; font-weight:bold;"><%if(request.getAttribute("message") != null)
        out.write((String)request.getAttribute("message")); %></h4>
      <input type="hidden" name="command" value="Login">
      <!-- Using JSP TAG --%>
      <%@ taglib uri="/WEB-INF/tag.tld" prefix="test" %>
      <test:helloTag/>
      <br>
      <h2>Please sign in</h2>
      <br>

      <!-- UserName input --%>
      <div class="input-group mb-3">
        <div class="input-group-append">
          
        </div>
        <input type="text" class="form-control" name="userName" placeholder="Username">
      </div>

      <!-- Password input --%>
      <div class="input-group mb-2">
        <div class="input-group-append">
          
        </div>
        <input type="password" class="form-control" name="password"
placeholder="Password">
      </div>

      <br>
      <input type="submit" name="command" value="Login" class="btn btn-success"><br>
      <a href="registration.jsp" class="badge badge-warning" role="button" aria-
pressed="true">Creat your GymTracking account</a><br>
    </form>
  </div>
</div>
```



```
</body>
</html>
```

registration.jsp

```
<%@ page language="java" contentType="text/html; charset=windows-1255" pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>Registration</title>
</head>

<body>
<!-- Registration content --%>
<div class="container" align="center">
    <div>
        <h2>Registration</h2>
        
    </div>

    <div>
        <h3>Enter your details:</h3>
    </div>
    <div>
        <!-- Form of registration --%>
        <form action="/WebGymTracking/GymTrackingController" method="post">
            <h4 style="color:red; font-weight:bold;"><%if(request.getAttribute("message") !=
null)>
                out.write((String)request.getAttribute("message")); %></h4>
            <input type="hidden" name="command" value="registration">

            <!-- Input to UserName --%>
            <div class="form-group">
                <label for="userName">Username:</label>
                <input type="text" class="form-control" name="userName" placeholder="Enter
username">
            </div>

            <!-- Input to Password --%>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" class="form-control" name="password" placeholder="Enter
password">
            </div>

            <!-- Input to Re-Password --%>
            <div class="form-group">
                <label for="rePassword">Re-Password:</label>
                <input type="password" class="form-control" name="rePassword" placeholder="Enter
password">
            </div>

            <!-- Input to Weight --%>
            <div class="form-group">
                <label for="weight">Weight:</label>
```

```

        <div align="center">
            <input type="number" size="3" step="1" min="30" max="150" placeholder="90"
id="inputWeight" >
        </div>
            <input type="range" id="sliderWeight" class="custom-range" name="weight"
value="90" min="30" max="150" />
        </div>

        <!-- Input to Height -->
        <div class="form-group">
            <label for="height">Height:</label>
            <div align="center">
                <input type="number" size="3" step="1" min="100" max="250" placeholder="150"
id="inputHeight" >
            </div>
            <input type="range" id="sliderHeight" class="custom-range" name="height"
value="150" min="100" max="250" />
        </div>

        <input type="submit" class="btn btn-outline-warning" name="command"
value="registration" ><br>
        <a href="Login.jsp" class="badge badge-light" role="button" aria-pressed="true">Back
to login</a><br>
    </form>
    <script>
        var sliderWeight = document.getElementById("sliderWeight");
        var sliderHeight = document.getElementById("sliderHeight");
        var inputWeight = document.getElementById("inputWeight");
        var inputHeight = document.getElementById("inputHeight");

        ///value updation from input to slider
        //function input update to slider
        function sliderWeightInput(){//input udate slider Weight
            sliderWeight.value = inputWeight.value;
        }
        function sliderHeightInput(){//input update slider Height
            sliderHeight.value = (inputHeight.value);
        }

        //calling function on change of inputs to update in slider
        inputWeight.addEventListener("change",sliderWeightInput);
        inputHeight.addEventListener("change",sliderHeightInput);

        ///value updation from slider to input
        //functions to update from slider to inputs
        function inputWeightSliderWeight(){//slider update inputs
            inputWeight.value = sliderWeight.value;
        }
        function inputHeightSliderHeight(){//slider update inputs
            inputHeight.value = sliderHeight.value;
        }
        sliderWeight.addEventListener("change",inputWeightSliderWeight);
        sliderHeight.addEventListener("change",inputHeightSliderHeight);
    </script>
    </div>
</div>

</body>
</html>

```

```

<%@ page language="java" import="com.hit.model.*,com.hit.controller.*" contentType="text/html;
charset=windows-1255" pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>User Main</title>
</head>

<body>
<!-- toolbar of user -->
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="userActivities.jsp">MyActivity</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="search.jsp">Search</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="about.jsp">About</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <%User user = (User)session.getAttribute("loginUser");
out.println(user.getUserName()); %>
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                    <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                    <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                    <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
                </div>
            </li>
            <li class="nav-item">
                <form action="/WebGymTracking/GymTrackingController" method="get">
                    <input type="hidden" name="command" value="Logout" >
                    <input class="btn btn-danger" type="submit" value="Logout">
                </form>
            </li>
        </ul>
    </div>
</nav>

<!-- MAIN content -->
<div align="center">
<br>

```

```

<h4 style="color:green; font-weight:bold;"><%if(request.getAttribute("message") != null)
    out.write((String)request.getAttribute("message")); %></h4>
    <br>
    <!-- Select options --%>
    <a href="addActive.jsp" class="btn btn-outline-primary btn-lg btn-block" role="button"
aria-pressed="true">Add New Activity</a><br>
    <a href="userActivities.jsp" class="btn btn-outline-info btn-lg btn-block" role="button"
aria-pressed="true">My Activities Table</a><br>
    <a href="BMICalculator.jsp" class="btn btn-outline-warning btn-lg btn-block"
role="button" aria-pressed="true">BMI Calculator</a><br>
    <form action="/WebGymTracking/GymTrackingController" method="get">
        <input type="hidden" name="command" value="Logout" >
        <input class="btn btn-outline-danger btn-lg btn-block" type="submit" value="Logout">
    </form>
    <br>
    <!-- Share a site to that user Facebook page --%>
    <iframe
src="https://www.facebook.com/plugins/share_button.php?href=http://www.WebGymTracking.com/&Layo
ut=button&size=small&mobile_iframe=true&width=60&height=20&appId" width="60" height="20"
style="border:none;overflow:hidden" scrolling="no" frameborder="0"
allowTransparency="true"></iframe>
</div>
</body>

</html>

```

addActive.jsp

```

<%@ page language="java" import="com.hit.model.*,com.hit.controller.*,java.util.*"
contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>Add Activity</title>
</head>

<body>
<!-- toolbar of user --%>
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="userActivities.jsp">MyActivity</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="search.jsp">Search</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="about.jsp">About</a>
            </li>

```

```

        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                <%User user = (User)session.getAttribute("loginUser");
                out.println(user.getUserName()); %>
            </a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
            </div>
        </li>
        <li class="nav-item">
            <form action="/WebGymTracking/GymTrackingController" method="get">
                <input type="hidden" name="command" value="logout" >
                <input class="btn btn-danger" type="submit" value="Logout">
            </form>
        </li>
    </ul>
</div>
</nav>

<!-- The content of the Add Active page -->
<div class="container">
<br>
<div align="center">
    
</div>
<p>Here you can add your activity, you must choose several sets and reps.</p>
<h5 style="color:red; font-weight:bold;"><%if(request.getAttribute("message") != null)
    out.write((String)request.getAttribute("message")); %></h5>
<h4>add new Activity:</h4>
    <!-- Form of adding -->
    <form action="/WebGymTracking/GymTrackingController" method="post">
        <div>
            <!-- Select type -->
            <label for="mySelect">Select a new active from the list:</label>
            <select class="selectpicker" data-style="btn-primary" id="mySelect"
name="mySelect">
                <optgroup label="Back">
                    <option value="Deadlift with rod">Deadlift with rod</option>
                    <option value="Narrow rowing in the machine">Narrow rowing in the
machine</option>
                    <option value="Horizontal bar">Horizontal bar</option>
                </optgroup>
                <optgroup label="Shoulders">
                    <option value="A standing shoulder press with a rod">A standing shoulder press
with a rod</option>
                    <option value="A shrug of the shoulders with weights">A shrug of the shoulders
with weights</option>
                </optgroup>
                <optgroup label="Chest">
                    <option value="Bench press">Bench press</option>
                    <option value="Breast butterfly with weights">Breast butterfly with
weights</option>
                    <option value="Pushups">Pushups</option>
                </optgroup>
                <optgroup label="Hands">
                    <option value="Bends and elbow">Bending the elbows with standing rod</option>
                    <option value="One-Arm Dumbbell Row">One-Arm Dumbbell Row</option>
                    <option value="Kick Back">Kick Back</option>
                    <option value="Pulling elbows in sitting with weights">Pulling elbows in sitting
with weights</option>
                </optgroup>
            </select>
        </div>
    </form>

```

```

        </optgroup>
        <optgroup label="Stomach">
        <option value="Static stomach/Plank">Static stomach/Plank</option>
        <option value="Crunches">Crunches</option>
        </optgroup>
        <optgroup label="Legs">
        <option value="Deadlift">Deadlift</option>
        <option value="Squat">Squat</option>
        <option value="Foot-pressing exercise against a machine">Foot-pressing exercise
against a machine</option>
        </optgroup>
    </select>
</div>

<br>
<!-- Choose a day -->
<div class="radio">
    <label for="myRadio">Choose the day you want to practice:</label><br>
    <input type="radio" name="myRadio" id="myRadio" value="Sunday"
checked><label>Sunday</label><br>
        <input type="radio" name="myRadio" id="myRadio"
value="Monday"><label>Monday</label><br>
        <input type="radio" name="myRadio" id="myRadio"
value="Tuesday"><label>Tuesday</label><br>
        <input type="radio" name="myRadio" id="myRadio"
value="Wednesday"><label>Wednesday</label><br>
        <input type="radio" name="myRadio" id="myRadio"
value="Thursday"><label>Thursday</label><br>
        <input type="radio" name="myRadio" id="myRadio"
value="Friday"><label>Friday</label><br>
        <input type="radio" name="myRadio" id="myRadio"
value="Saturday"><label>Saturday</label><br>
    </div>
<br>

<!-- Quantity Sets -->
<div class="form-group">
    <label for="numberOfSets">Sets:</label>
    <div align="center">
        <input type="number" size="2" step="1" min="1" max="50" placeholder="5"
id="inputSets" >
    </div>
    <input type="range" id="sliderSets" class="custom-range" name="numberOfSets"
value="5" min="1" max="50"/>
    </div>

    <!-- Quantity Reps -->
    <div class="form-group">
        <label for="numberOfReps">Reps:</label>
        <div align="center">
            <input type="number" size="2" step="1" min="5" max="100" placeholder="20"
id="inputRets" >
        </div>
        <input type="range" id="sliderReps" class="custom-range" name="numberOfReps"
value="20" min="5" max="100" />
    </div>

    <div align="center">
        <input type="submit" class="btn btn-primary" name="command" value="addActive"><br>
        <a href="userMain.jsp" class="badge badge-info" role="button" aria-
pressed="true">Back to Main</a>
    </div>

</form>
<script>

```

```

var sliderSets = document.getElementById("sliderSets");
var sliderReps = document.getElementById("sliderReps");
var inputSets = document.getElementById("inputSets");
var inputReps = document.getElementById("inputRets");

//value updation from input to slider
//function input update to slider
function sliderSetsInput(){//input udate slider Sets
    sliderSets.value = inputSets.value;
}
function sliderRepsInput(){//input update slider Reps
    sliderReps.value = (inputReps.value);
}

//calling function on change of inputs to update in slider
inputSets.addEventListener("change",sliderSetsInput);
inputReps.addEventListener("change",sliderRepsInput);

//value updation from slider to input
//functions to update from slider to inputs
function inputSetsSliderSets(){//slider update inputs
    inputSets.value = sliderSets.value;
}
function inputRepsSliderReps(){//slider update inputs
    inputReps.value = sliderReps.value;
}
sliderSets.addEventListener("change",inputSetsSliderSets);
sliderReps.addEventListener("change",inputRepsSliderReps);
</script>

```

```

</div>
</body>
</html>

```

userActivities.jsp

```

<%@ page language="java" import="com.hit.model.*,com.hit.controller.*,java.util.*"
contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/styleForTable.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>User Activity</title>
</head>

<body>
<!-- toolbar of user --%>
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">

```



```

        <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="userActivities.jsp">MyActivity</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="search.jsp">Search</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="about.jsp">About</a>
    </li>
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            <%User user = (User)session.getAttribute("loginUser");
            out.println(user.getUserName()); %>
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
            <a class="dropdown-item" href="updateUser.jsp">Update User</a>
            <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
            <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
        </div>
    </li>
    <li class="nav-item">
        <form action="/WebGymTracking/GymTrackingController" method="get">
            <input type="hidden" name="command" value="logout" >
            <input class="btn btn-danger" type="submit" value="logout">
        </form>
    </li>
</ul>
</div>
</nav>

<div class="jumbotron" align="center">
    <h2 class="display-4">Training Table</h2>
    
    <p class="lead">Here you can see all your activities, <b><%
out.println(user.getUserName()); %></b>.
    <br>You can update or delete your activities.
    <br>Number of sessions are: <b><% out.println(ServletListenerCounter.getSessionCounter());
%></b></p>
    <hr class="my-4">
    <h5 style="color:green; font-weight:bold;"><%if(request.getAttribute("message") != null)
        out.write((String)request.getAttribute("message")); %></h5>
    <a class="btn btn-primary btn-lg" href="addActive.jsp" role="button">Add Activity</a>
    <a class="btn btn-info btn-lg" href="userMain.jsp" role="button">Back to Main</a>
    <br>
    <a class="btn btn-outline-warning btn-lg" href="trainingTableByDay.jsp" role="button">Show
my training table by day</a>
</div>

<!-- Table contents -->
<div class="container">
    <table>
        <thead class="thead-dark">
            <tr>
                <th scope="col">#</th>
                <th scope="col">Type</th>
                <th scope="col">Sets</th>
                <th scope="col">Reps</th>
                <th scope="col">Day</th>
                <th scope="col"></th>
                <th scope="col"></th>
            </tr>

```



```

        </tr>
    </thead>
    <%
        List<Activity> userActivity =
        (List<Activity>)session.getAttribute("userActivity");//Gets activity list of user
        request.setAttribute("activityList", userActivity);
        if(userActivity.isEmpty()) {
            out.println("<tbody><tr><th scope='row'></th><td colspan='6'><b>You have no
activities, start adding.</b></td></tr></tbody>");
        }
        else {
            int i=1;
            for(Activity activity : userActivity)//Prints the user's activities
            {
                out.println(
                    "<tbody><tr><th scope='row'>" + i + "</th>" +
                    "<td>" + activity.getActivityName() +
                    "</td><td>" + activity.getNumberOfSets() +
                    "</td><td>" + activity.getNumberOfReps() +
                    "</td><td>" + activity.getTrainingDay() +
                    "</td><td>" +
                    "<form action='/WebGymTracking/GymTrackingController' method='get'><input
type='hidden' name='command' value='sendToUpdate'><input type='hidden'
value='" + activity.getActivityId() +
                    "' name='activityId'><input type='submit' value='update' class='btn btn-
success'></form>" +
                    "</td><td>" +
                    "<form action='/WebGymTracking/GymTrackingController' method='get'><input
type='hidden' name='command' value='delete'><input type='hidden'
value='" + activity.getActivityId() +
                    "' name='activityId'><input type='submit' value='delete' class='btn btn-
danger'></form>" +
                    "</td></tr><tbody>");
                i++;
            }
        }
    <%>
</table>
<br>
</div>

</body>
</html>

```

BMICalculator.jsp

```

<%@ page language="java" import="com.hit.model.*" contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <script type="text/javascript" src="js/fusioncharts/fusioncharts.js"></script>
    <script type="text/javascript" src="js/fusioncharts/fusioncharts.theme.fusion.js"></script>
    <title>BMI Calculator</title>

    <style>
    #checkbox-container {
        margin-top: 10px;
        width: 400px;
    }
    b {

```

```
font-weight: bold;
color : blue;
}
</style>
```

```
<!-- Calculation and presentation of a BMI graph -->
<script type="text/javascript">
<%User user = (User)session.getAttribute("loginUser");%>
var w = <%=user.getWeight()%>;
var h = <%=user.getHeight()%>;
var bmiValue = Math.round(w / Math.pow(h, 2) * 10000);
```

```
FusionCharts.ready(function() {
    var showCheckBox = document.getElementById('showCB'),
        posCheckBox = document.getElementById('positionCB'),
        cpuGauge = new FusionCharts({
            type: 'hlineargauge',
            renderAt: 'chart-container',
            width: '270',
            height: '150',
            dataFormat: 'json',
            dataSource: {
                "chart": {
                    "theme": "fusion",
                    "caption": "Your BMI",
                    "subcaptionFontBold": "0",
                    "lowerLimit": "0",
                    "upperLimit": "100",
                    "chartBottomMargin": "20",
                    "captionPadding": "10",
                    "valueAbovePointer": "0",
                    "valueFontSize": "13",
                    "valueFontBold": "0",
                    "gaugeFillMix": "{light-10},{light-70},{dark-10}",
                    "gaugeFillRatio": "40,20,40"
                },
                "colorRange": {
                    "color": [{
                        "minValue": "0",
                        "maxValue": "18.5",
                        "code": "#66e6f0"
                    },
                    {
                        "minValue": "18.9",
                        "maxValue": "24.9",
                        "code": "#62B58F"
                    },
                    {
                        "minValue": "25",
                        "maxValue": "29.9",
                        "code": "#FFC533"
                    },
                    {
                        "minValue": "30",
                        "maxValue": "100",
                        "code": "#F2726F"
                    }
                ]
            },
            "pointers": {
                "pointer": [{
                    "value": bmiValue
                }]
            }
        })
    showCheckBox.onclick = function() {
        posCheckBox.checked = !posCheckBox.checked;
    }
    posCheckBox.onclick = function() {
        showCheckBox.checked = !showCheckBox.checked;
    }
});
```

```

    })
    .render();

//Set event listener for check boxes
showCheckBox.addEventListener && showCheckBox.addEventListener("click", showValue);
posCheckBox.addEventListener && posCheckBox.addEventListener("click", setPosition);

//Function to show hide value
function showValue(evt, obj) {
    //Using showValue attribute to show hide pointer value
    if (showCB.checked) {
        cpuGauge.setChartAttribute('showValue', 1);
        posCheckBox.disabled = false;
    }
    else {
        cpuGauge.setChartAttribute('showValue', 0);
        posCheckBox.disabled = true;
    }
}

//Function to change value pointer
function setPosition(evt, obj) {
    //Using valueAbovePointer attribute to toggle position of pointer value
    (posCheckBox.checked) ? cpuGauge.setChartAttribute('valueAbovePointer', 1):
        cpuGauge.setChartAttribute('valueAbovePointer', 0);
}

});
</script>

</head>
<body>
<!-- toolbar of user --%>
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
        <li class="nav-item active">
            <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="userActivities.jsp">MyActivity</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="search.jsp">Search</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="about.jsp">About</a>
        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                <%out.println(user.getUserName()); %>
            </a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update
Weight And Height</a>
                <a class="dropdown-item" href="updatePassword.jsp">Update
Password</a>
            </div>
        </li>
    </ul>
    </div>
</nav>

```

```

        </div>
    </li>
    <li class="nav-item">
        <form action="/WebGymTracking/GymTrackingController" method="get">
            <input type="hidden" name="command" value="logout" >
            <input class="btn btn-danger" type="submit" value="logout">
        </form>
    </li>
</ul>
</div>
</nav>

<!-- User profile and graph content -->
<div align="center">
<h1>BMI Coalculator</h1>

<div>
    <h2><b style="color:black">Your profile is: </b></h2>
    <br>
    <%%out.println(bmiValue);%>

    <h4>UserName: <b><% out.println(user.getUserName()); %></b></h4>
    <h4>Weight: <b><% out.println(user.getWeight()); %>Kg</b></h4>
    <h4>Height: <b><% out.println(user.getHeight()); %>Cm</b></h4>
    <br>
    <!-- Graph using with FusionCharts -->
    <div id="chart-container">FusionCharts will render here</div>
    <div id="checkbox-container">
        <span style="color:#66e6f0">Low </span><span style="color:#62B58F">Normal </span>
        <span style="color:#FFC533">Overweight </span><span
style="color:#F2726F">Obese</span><br>
        <input type="checkbox" id="showCB" name="padding" value="1" checked>Show Value<br>
        <input type="checkbox" id="positionCB" name="padding" value="2">Place Value Above
Pointer
    </div>
    <br>
    <a href="userMain.jsp" class="badge badge-info">Back To Main</a>
</div>
</div>

</body>
</html>

```

trainingTableByDay.jsp

```

<%@ page language="java" import="com.hit.model.*,com.hit.controller.*,java.util.*"
contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>Training Table By Day</title>
</head>

<body>
<!-- toolbar of user -->
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>

```

```

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="userActivities.jsp">MyActivity</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="search.jsp">Search</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="about.jsp">About</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <%User user = (User)session.getAttribute("loginUser");
out.println(user.getUserName()); %>
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                    <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                    <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                    <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
                </div>
            </li>
            <li class="nav-item">
                <form action="/WebGymTracking/GymTrackingController" method="get">
                <input type="hidden" name="command" value="Logout" >
                <input class="btn btn-danger" type="submit" value="Logout">
                </form>
            </li>
        </ul>
    </div>
</nav>

<div class="jumbotron" align="center">
    <h2 class="display-4">Training Table</h2>
    
    <p class="lead">Here you can see all your activities, <b><%
out.println(user.getUserName()); %></b>.
    <br>You can update or delete your activities.
    <br>Number of sessions are: <b><% out.println(ServletListenerCounter.getSessionCounter());
%></b></p>
    <hr class="my-4">
    <h5 style="color:green; font-weight:bold;"><%if(request.getAttribute("message") != null)
out.write((String)request.getAttribute("message")); %></h5>
    <a class="btn btn-primary btn-lg" href="addActive.jsp" role="button">Add Activity</a>
    <a class="btn btn-info btn-lg" href="userMain.jsp" role="button">Back to Main</a>
    <br>
    <a class="btn btn-outline-warning btn-lg" href="userActivities.jsp" role="button">Show my
training table</a>
</div>

<!-- Table contents -->
<div class="container">
    <table class="table">

```

```

        <thead class="thead-dark">
        <tr>
            <th scope="col">Days</th>
            <th scope="col">Quantity</th>
        </tr>
        </thead>
        <%
            List<Activity> userActivity =
(List<Activity>)session.getAttribute("userActivity");//Gets activity list of user
            request.setAttribute("activityList", userActivity);
            if(userActivity.isEmpty()) {
                out.println("<tbody><tr><th scope='row'></th><td colspan='2'><b>You have no
activities, start adding.</b></td></tr></tbody>");
            }
            else {
                String nameOfDays[] =
{"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
                int days[] = new int[7]; //An array of days
                for(Activity activity : userActivity) //Count of days
                {
                    String day = activity.getTrainingDay();
                    if(day.equals("Sunday"))
                        days[0]++;
                    else if(day.equals("Monday"))
                        days[1]++;
                    else if(day.equals("Tuesday"))
                        days[2]++;
                    else if(day.equals("Wednesday"))
                        days[3]++;
                    else if(day.equals("Thursday"))
                        days[4]++;
                    else if(day.equals("Friday"))
                        days[5]++;
                    else if(day.equals("Saturday"))
                        days[6]++;
                }

                for(int j=0 ; j < 7 ; j++) //Prints the amount of training on any given day
                {
                    out.println(
                        "<tbody><tr><th scope='row'>" + nameOfDays[j] + "</th>" +
                        "<td>" + days[j] +
                        "</td></tr></tbody>");
                }

            }
        <%>
    </table>

    <br>
    <div align="center">
        <a class="btn btn-warning btn-lg" href="graphActivity.jsp" role="button">To show graph
activity order by days</a>
    </div>
    <br>
</div>

</body>
</html>

```

graphActivity.jsp

```

<%@ page language="java" import="com.hit.model.*,com.hit.controller.*,java.util.*"
contentType="text/html; charset=windows-1255"

```

```

    pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <script type="text/javascript" src="js/fusioncharts/fusioncharts.js"></script>
    <script type="text/javascript" src="js/fusioncharts/fusioncharts.theme.fusion.js"></script>
    <title>Graph Activity</title>
</head>

<body>
<!-- toolbar of user -->
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="userActivities.jsp">MyActivity</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="search.jsp">Search</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="about.jsp">About</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <%User user = (User)session.getAttribute("loginUser");
out.println(user.getUserName()); %>
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                    <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                    <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                    <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
                </div>
            </li>
            <li class="nav-item">
                <form action="/WebGymTracking/GymTrackingController" method="get">
                    <input type="hidden" name="command" value="Logout" >
                    <input class="btn btn-danger" type="submit" value="Logout">
                </form>
            </li>
        </ul>
    </div>
</nav>

<script type="text/javascript">

```

```

<%List<Activity> userActivity = (List<Activity>)session.getAttribute("userActivity");//Gets
activity list of user
request.setAttribute("activityList", userActivity);
int days[] = new int[7];//An array of days
for(Activity activity : userActivity)//Count of days
{
    String day = activity.getTrainingDay();
    if(day.equals("Sunday"))
        days[0]++;
    else if(day.equals("Monday"))
        days[1]++;
    else if(day.equals("Tuesday"))
        days[2]++;
    else if(day.equals("Wednesday"))
        days[3]++;
    else if(day.equals("Thursday"))
        days[4]++;
    else if(day.equals("Friday"))
        days[5]++;
    else if(day.equals("Saturday"))
        days[6]++;
}
%>
var sunday = <%=days[0] %>,
monday = <%=days[1] %>,
tuesday = <%=days[2] %>,
wednesday = <%=days[3] %>,
thursday = <%=days[4] %>,
friday = <%=days[5] %>
saturday = <%=days[6] %>;
FusionCharts.ready(function(){//function of the graph
    var fusioncharts = new FusionCharts({
        type: 'column2d',
        renderAt: 'chart-container',
        width: '100%',
        height: '100%',
        dataFormat: 'json',
        dataSource: {
            // Chart Configuration
            "chart": {
                "caption": "Activity Graph",
                "subCaption": "Order by days",
                "xAxisName": "Days",
                "yAxisName": "Quantity Activity",
                "numberSuffix": "",
                "theme": "fusion",
            },
            // Chart Data
            "data": [{
                "label": "Sun",
                "value": sunday
            }, {
                "label": "Mon",
                "value": monday
            }, {
                "label": "Tue",
                "value": tuesday
            }, {
                "label": "Wed",
                "value": wednesday
            }, {
                "label": "Thu",
                "value": thursday
            }, {
                "label": "Fri",

```



```

        "value": friday
      }, {
        "label": "Sat",
        "value": saturday
      }
    ]
  });
  fusioncharts.render();
});
</script>

<!-- Displays the graph by days -->
<div align="center">
  <h3>Your graph :</h3>
  
  <br>
  <p> Here you can see your graph activity order by days:</p>
  <!-- Graph using with FusionCharts -->
  <div id="chart-container">FusionCharts XT will load here!</div>
  <br>
  <a href="userActivities.jsp" class="badge badge-secondary">Back To User Activities</a>
</div>

</body>
</html>

```

search.jsp

```

<%@ page language="java" import="com.hit.model.*" contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
  <script src="js/view/popper/popper.min.js"></script>
  <script src="js/view/bootstrap/bootstrap.min.js"></script>
  <title>Search</title>
  <style>
    h4{
      color:red
    }
    em {
      font-style: normal;          /* Removes italics */
      text-decoration: underline; /* Makes underline */
    }
  </style>
</head>

<body>
<!-- toolbar of user -->
<nav class="navbar navbar-dark bg-dark">
  <h3 style="color:white;">GymTracking</h3>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
      </li>

```

```

<li class="nav-item">
    <a class="nav-link" href="userActivities.jsp">MyActivity</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="search.jsp">Search</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="about.jsp">About</a>
</li>
<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        <%User user = (User)session.getAttribute("loginUser");
        out.println(user.getUserName()); %>
    </a>
    <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
        <a class="dropdown-item" href="updateUser.jsp">Update User</a>
        <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
        <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
    </div>
</li>
<li class="nav-item">
    <form action="/WebGymTracking/GymTrackingController" method="get">
    <input type="hidden" name="command" value="logout" >
    <input class="btn btn-danger" type="submit" value="logout">
    </form>
</li>
</ul>
</div>
</nav>

```

```

<div align="center">
    <br>
    <!-- Button trigger modal -->
    <h4><em>Back</em></h4>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#DeadliftWithRod">Deadlift with rod</button><br><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#NarrowRowingInTheMachine">Narrow rowing in the machine</button><br><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#HorizontalBar">Horizontal bar</button><br><br>
    <h4><em>Shoulders</em></h4><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#AStandingShoulderPressWithARod">A standing shoulder press with a rod</button><br><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#AShrugOfTheShouldersWithWeights">A shrug of the shoulders with
weights</button><br><br>
    <h4><em>Chest</em></h4><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#BenchPress">Bench press</button><br><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#BreastButterfly with weights">Breast butterfly with weights</button><br><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#Pushups">Pushups</button><br><br>
    <h4><em>Hands</em></h4><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#BendingTheElbowsWithStandingRod">Bending the elbows with standing rod</button><br><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#One-ArmDumbbellRow">One-Arm Dumbbell Row</button><br><br>
    <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#KickBack">Kick Back</button><br><br>

```

```

        <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#PullingElbowsInSittingWithWeights">Pulling elbows in sitting with
weights</button><br><br>
        <h4><em>Stomach</em></h4><br>
        <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#StaticStomachPlank">Static stomach/Plank</button><br><br>
        <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#Crunches">Crunches</button><br><br>
        <h4><em>Legs</em></h4><br>
        <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#Deadlift">Deadlift</button><br><br>
        <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#Squat">Squat</button><br><br>
        <button type="button" class="btn btn-outline-light text-dark" data-toggle="modal" data-
target="#Foot-pressingExerciseAgainstAMachine">Foot-pressing exercise against a
machine</button><br><br>
        <a href="userMain.jsp" class="badge badge-info" role="button" aria-pressed="true">Back
to Main</a>
</div>

<!-- Explanations of the Modals -->
<div class="modal fade" id="DeadliftWithRod" tabindex="-1" role="dialog" aria-
labelledby="DeadliftWithRod" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="DeadliftWithRod">Deadlift with rod</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                Deadlift is a weight training exercise in which an Olympic pole is lifted from
                the floor in a position where the legs are bent and the back straight.
                Deadlift is a complex exercise that improves the grip of the hands, and works mainly on
                the muscles of the
                back of the back,the large buttock muscle, the large mitochondrial muscle, the muscles
                of the hamstring,
                the quadriceps and the soles of the sole.
                <br> <br> <b> Tip: </b> 1 set of 5 reps once a week.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>

<div class="modal fade" id="NarrowRowingInTheMachine" tabindex="-1" role="dialog" aria-
labelledby="NarrowRowingInTheMachine" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="NarrowRowingInTheMachine">Narrow rowing in the machine</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                One of the base exercises whose primary purpose is to work on the broad top (Latissimus
                Dorsi).
                The rowing begins and ends on the floor at each repetition [like deadlift]
                and does not end when the weight is suspended in the air with the hands straight.
            </div>
        </div>
    </div>
</div>

```

```

        <br> <br> <b> Tip:</b> Perform 6-1 high intensity repetitions, with 3-5 minutes rest
between the sets.
    <br> <b> Note: </b>Stretching is recommended at the end of the exercise.
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
    <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
</div>
</div>
</div>
</div>
</div>

<div class="modal fade" id="HorizontalBar" tabindex="-1" role="dialog" aria-
labelledby="HorizontalBar" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="HorizontalBar">Horizontal bar</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                Stress is a fitness facility consisting of a horizontal bar that is far from the ground.
                The purpose of the facility is to allow the user to hang on it and lift it with his help,
                while activating the muscles of the hands, back and abdomen.
                The accepted style of tension is to hang on to it while holding on to
                the hands and without touching the ground, and making ups and downs without the legs'
                help, so that
                the chin's lift passes the rod, and the descent is up to the alignment of the hands.
            <br> <br> <b> Tip:</b> 3 sets of 6 reps with a minute to 2 minutes rest with maximum
weight.
            <br> <b>Perform 6 repetitions of tension and feel easy for you? </b>
            <br>Add weight in the form of a handweight between your legs.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>

<div class="modal fade" id="AStandingShoulderPressWithARod" tabindex="-1" role="dialog" aria-
labelledby="AStandingShoulderPressWithARod" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="AStandingShoulderPressWithARod">A standing shoulder press
with a rod</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                A shoulder press is designed for deep work on the front and middle heads of the delta
muscle.
                Through it you increase the mass and strength of all the muscles around the shoulder
joint.
            <br> <br> <b> Tip:</b> Perform 2-4 sets of 7-12 repetitions.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
  </div>
</div>

<div class="modal fade" id="AShrugOfTheShouldersWithWeights" tabindex="-1" role="dialog" aria-
labelledby="AShrugOfTheShouldersWithWeights" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="AShrugOfTheShouldersWithWeights">A shrug of the shoulders
with weights</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        A shoulder press is designed for deep work on the front and middle heads of the delta
muscle.
        Through it you increase the mass and strength of all the muscles around the shoulder
joint.
        <br> <br> <b> Tip:</b> Perform 2-4 sets of 7-12 repetitions.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
      </div>
    </div>
  </div>
</div>
</div>

```

```

<div class="modal fade" id="BenchPress" tabindex="-1" role="dialog" aria-
labelledby="BenchPress" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="BenchPress">Bench press</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        The bench press is a great exercise for anyone looking to gain muscle size and strength
in the upper body.
        This exercise is known as a compound exercise, meaning it involves multiple joints, and
therefore
        multiple major muscle groups to execute.
        <br> <br> <b> Tip:</b> Perform 3-5 sets of 7-12 repetitions.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
      </div>
    </div>
  </div>
</div>
</div>

```

```

<div class="modal fade" id="BreastButterflyWithWeights" tabindex="-1" role="dialog" aria-
labelledby="BreastButterflyWithWeights" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="BreastButterflyWithWeights">Breast butterfly with
weights</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
    </div>
  </div>
</div>

```

```

        </button>
    </div>
    <div class="modal-body">
        A breast butterfly in the machine is an effective exercise to sharpen the chest muscles.
        Exercise contributes to the chest appearance "rough".
        <br> <br> <b> Tip:</b> Perform 3-4 sets of 8-15 repetitions.
        <br>Before that, regular breast training should be performed.
        <br>The exercise is suitable for trainees at all levels.
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
    </div>
</div>
</div>
</div>

```

```

<div class="modal fade" id="Pushups" tabindex="-1" role="dialog" aria-labelledby="Pushups"
aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="Pushups">Pushups</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                Push-ups is a physical exercise exercise, based on applying the entire body's strength to
                the elbows and legs.
                The gymnast straightens his elbows while pushing the body upwards.Push-ups are designed
                to strengthen
                the muscles of the hands (triceps), back, shoulder (delta muscle) and chest (large chest
                muscle).
                <br> <br> <b> Tip:</b> Perform 3 sets of 15 repetitions. rest of 30-60 seconds between
                each set.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>

```

```

<div class="modal fade" id="BendingTheElbowsWithStandingRod" tabindex="-1" role="dialog" aria-
labelledby="BendingTheElbowsWithStandingRod" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="BendingTheElbowsWithStandingRod">Bending the elbows with
                standing rod</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                his exercise works on the biceps of the hand. The muscle participates in all
                movements that combine hand-pulling towards the body.
                In addition, this is the most effective exercise toincrease the strength and volume of
                the hand.
                <br> <br> <b> Tip: </b> 2-3 set of 8-12 reps.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
            </div>
        </div>
    </div>
</div>

```



```
        <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
    </div>
</div>
</div>
</div>
```

```
<div class="modal fade" id="One-ArmDumbbellRow" tabindex="-1" role="dialog" aria-
labelledby="One-ArmDumbbellRow" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="One-ArmDumbbellRow">One-Arm Dumbbell Row</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                his exercise targets the upper and lower back, shoulders, biceps, and hips while
improving core stability.
                Five different joint actions take place in this compound exercise.Beginners can use light
weights as they build strength.
                This is also a good exercise to do as part of a circuit training routine.
                <br> <br> <b> Tip: </b> 2-3 set of 8-12 reps.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>
```

```
<div class="modal fade" id="KickBack" tabindex="-1" role="dialog" aria-labelledby="KickBack"
aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="KickBack">Kick Back</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                Purpose of the exercise: Strengthening The triceps muscle (the back of the arm).
                The rear hand muscle is a very large muscle in the hand and it assembles most of the hand
mass.
                It is located at the back of the hand and is responsible for the movement of the elbow
alignment.
                <br> <br> <b> Tip: </b> 3 set of 20 reps once a week.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>
```

```
<div class="modal fade" id="PullingElbowsInSittingWithWeights" tabindex="-1" role="dialog"
aria-labelledby="PullingElbowsInSittingWithWeights" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="PullingElbowsInSittingWithWeights">Pulling elbows in
sitting with weights</h5>
```

```

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>
    <div class="modal-body">
        The best aim of the exercise is tosharpen it and increase the strength of the biceps muscle.
        Full shoulder flexion should beallowed to allow full hand pulling. It should be noted that the brace
        (if any) will not be too high and will prevent the weight of the hand from being fully lowered.
        <br> <br> <b> Tip: </b> 3-4 set of 8-12 reps once a week.
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
    </div>
</div>
</div>
</div>
```

```

<div class="modal fade" id="StaticStomachPlank" tabindex="-1" role="dialog" aria-labelledby="StaticStomachPlank" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="StaticStomachPlank">Static stomach/Plank</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                A static stomach is a power exercise that involves staying in a static position.
                The most common plank is a front plank, which stays in a push-up position when the body weight is carried on the arms, elbows and toes.
                The plank strengthens the abdominal muscles, back and shoulders.
                <br> <br> <b> Tip:</b> Perform 4 sets. 60 seconds in the first set, 50 seconds in the second, 40 seconds in the third,
                30 seconds in the fourth. rest of 45 seconds between each set.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>
```

```

<div class="modal fade" id="Crunches" tabindex="-1" role="dialog" aria-labelledby="Crunches" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="Crunches">Crunches</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                Crunches is a physical exercise exercise, based on activating the mass of the upper body on the abdominal muscle.
                Crunches are designed to strengthenthe abdominal muscles. Exercise is one of the popular exercises because it can be done anywhere and without special equipment.
                <br> <br> <b> Tip:</b> Perform 55 sets of 40 repetitions. rest of 30-60 seconds between each set.
            </div>
        </div>
    </div>
</div>
```



```

    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
    </div>
</div>
</div>
</div>

<div class="modal fade" id="Deadlift" tabindex="-1" role="dialog" aria-labelledby="Deadlift"
aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="Deadlift">Deadlift</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <u>Deadlift</u> is a weight training exercise in which an <u>Olympic</u> pole is lifted from the floor
                in a position where the legs
                are bent and the back straight. Deadlift is a complex exercise that improves the grip of
                the hands, and
                works mainly on the muscles of the back of the back, the large buttock muscle, the large
                <u>mitochondrial</u> muscle,
                the muscles of the <u>hamstring</u>, the <u>quadriceps</u> and the soles of the sole.
                <br> <br> <b> Tip: </b> 1 set of 5 <u>reps</u> once a week.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>

<div class="modal fade" id="Squat" tabindex="-1" role="dialog" aria-labelledby="Squat" aria-
hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="Squat">Squat</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                A squat is a complex <u>multi</u>-joint exercise that, when performed, activates the ankle
                joint, knee, and thigh.
                As a result of this activity, the body mobilizes a large number of muscles, in fact,
                almost all
                the muscles in the body . The exercise shapes the muscles of the legs and buttocks.
                <br> <br> <b> Tip: </b> 4 set of 6 <u>reps</u>.
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
            </div>
        </div>
    </div>
</div>
</div>

<div class="modal fade" id="Foot-pressingExerciseAgainstAMachine" tabindex="-1" role="dialog"
aria-labelledby="Foot-pressingExerciseAgainstAMachine" aria-hidden="true">

```

```

<div class="modal-dialog modal-dialog-centered" role="document">
  <div class="modal-content">
    <div class="modal-header">
      <h5 class="modal-title" id="Foot-pressingExerciseAgainstAMachine">Foot-pressing
exercise against a machine</h5>
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
    <div class="modal-body">
      The exercise is performed using power machines. In this exercise there are three
agonists, the buttocks, the quadriceps and
      the sinus muscles, and three joints involved in the exercise, the hip joint, the knee
joint, and the ankle joint.
      <br> <br> <b> Tip: </b> 4 set of 15 reps.
    </div>
    <div class="modal-footer">
      <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
      <a class="btn btn-primary" href="addActive.jsp" role="button">Add Active</a>
    </div>
  </div>
</div>
</div>
</body>
</html>

```

about.jsp

```

<%@ page language="java" import="com.hit.model.*" contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
  <script src="js/view/popper/popper.min.js"></script>
  <script src="js/view/bootstrap/bootstrap.min.js"></script>
  <jsp:useBean id="bg" class="com.hit.wp.model.BGBean" scope="session"/>
  <jsp:setProperty property="*" name="bg" />
  <title>About</title>
</head>

<!-- Using jsp:useBean and jsp:getProperty --%>
<body style="background: rgb(<jsp:getProperty name="bg" property="r"/>,
                           <jsp:getProperty name="bg" property="g"/>,
                           <jsp:getProperty name="bg" property="p"/>)">

<!-- toolbar of user --%>
<nav class="navbar navbar-dark bg-dark">
  <h3 style="color:white;">GymTracking</h3>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="userActivities.jsp">MyActivity</a>

```

```

        </li>
        <li class="nav-item">
            <a class="nav-link" href="search.jsp">Search</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="about.jsp">About</a>
        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                <%User user = (User)session.getAttribute("loginUser");
                out.println(user.getUserName()); %>
            </a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
            </div>
        </li>
        <li class="nav-item">
            <form action="/WebGymTracking/GymTrackingController" method="get">
                <input type="hidden" name="command" value="logout" >
                <input class="btn btn-danger" type="submit" value="Logout">
            </form>
        </li>
    </ul>
</div>
</nav>

<!-- Information about developers --%>
<div align="center">
    <br>
    <h3>This page created and desinged by : </h3>
    <br><h5>&#169;<a class="btn btn-link" href="https://www.linkedin.com/in/sapir-abuksis/"
role="button">Sapir Abukasis</a> &#38;
    <a class="btn btn-link" href="https://www.linkedin.com/in/alexey-shpigelman/"
role="button">Alexey Shpigelman</a></h5>
    <br>
    <a href="userMain.jsp" class="badge badge-info">Back To Main</a>
</div>

</body>
</html>

```

updateActivity.jsp

```

<%@ page language="java" import="com.hit.model.*" contentType="text/html; charset=windows-
1255"
    pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>Update Activity</title>
</head>

<body>
<!-- toolbar of user --%>
<nav class="navbar navbar-dark bg-dark">

```

```

        <h3 style="color:white;">GymTracking</h3>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item active">
                    <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="userActivities.jsp">MyActivity</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="search.jsp">Search</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="about.jsp">About</a>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        <%User user = (User)session.getAttribute("loginUser");
out.println(user.getUserName()); %>
                    </a>
                    <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                        <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                        <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                        <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
                    </div>
                </li>
                <li class="nav-item">
                    <form action="/WebGymTracking/GymTrackingController" method="get">
                        <input type="hidden" name="command" value="Logout" >
                        <input class="btn btn-danger" type="submit" value="Logout">
                    </form>
                </li>
            </ul>
        </div>
</nav>

```

```

<div class="container">
    <!-- Active update form --%>
    <form action="/WebGymTracking/GymTrackingController" method="get">
        <h4>Updating the training</h4>
        <br>
        <!-- Choose a day to change --%>
        <div class="radio">
            <label for="myRadio">Choose the day you want to practice:</label><br>
            <input type="radio" name="myRadio" id="myRadio" value="Sunday"
checked><label>Sunday</label><br>
            <input type="radio" name="myRadio" id="myRadio"
value="Monday"><label>Monday</label><br>
            <input type="radio" name="myRadio" id="myRadio"
value="Tuesday"><label>Tuesday</label><br>
            <input type="radio" name="myRadio" id="myRadio"
value="Wednesday"><label>Wednesday</label><br>
            <input type="radio" name="myRadio" id="myRadio"
value="Thursday"><label>Thursday</label><br>

```

```

        <input type="radio" name="myRadio" id="myRadio"
value="Friday"><label>Friday</label><br>
        <input type="radio" name="myRadio" id="myRadio"
value="Saturday"><label>Saturday</label><br>
    </div>
    <br>

    <!-- Change the quantity of sets -->
    <div class="form-group">
        <label for="numberOfSets">Sets:</label>
        <div align="center">
            <input type="number" size="2" step="1" min="1" max="50" placeholder="5"
id="inputSets" >
        </div>
        <input type="range" id="sliderSets" class="custom-range" name="newNumberOfSets"
value="5" min="1" max="50"/>
    </div>

    <!-- Change the quantity of reps -->
    <div class="form-group">
        <label for="numberOfReps">Reps:</label>
        <div align="center">
            <input type="number" size="2" step="1" min="5" max="100" placeholder="20"
id="inputReps" >
        </div>
        <input type="range" id="sliderReps" class="custom-range" name="newNumberOfReps"
value="20" min="5" max="100" />
    </div>

    <input class="btn btn-primary" type="submit" name="command"
value="updateActivity"><br>
    <a href="userActivities.jsp" class="badge badge-light" role="button" aria-
pressed="true">Return to table</a><br>
</form>
<script>
var sliderSets = document.getElementById("sliderSets");
var sliderReps = document.getElementById("sliderReps");
var inputSets = document.getElementById("inputSets");
var inputReps = document.getElementById("inputReps");

//value updation from input to slider
//function input update to slider
function sliderSetsInput(){//input update slider Sets
    sliderSets.value = inputSets.value;
}
function sliderRepsInput(){//input update slider Reps
    sliderReps.value = (inputReps.value);
}

//calling function on change of inputs to update in slider
inputSets.addEventListener("change",sliderSetsInput);
inputReps.addEventListener("change",sliderRepsInput);

//value updation from slider to input
//functions to update from slider to inputs
function inputSetsSliderSets(){//slider update inputs
    inputSets.value = sliderSets.value;
}
function inputRepsSliderReps(){//slider update inputs
    inputReps.value = sliderReps.value;
}
sliderSets.addEventListener("change",inputSetsSliderSets);
sliderReps.addEventListener("change",inputRepsSliderReps);
</script>
</div>

```

```
</body>
</html>
```

updatePassword.jsp

```
<%@ page language="java" import="com.hit.model.*" contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>Update Password</title>
</head>

<body>
<!-- toolbar of user --%>
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="userActivities.jsp">MyActivity</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="search.jsp">Search</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="about.jsp">About</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <%User user = (User)session.getAttribute("loginUser");
out.println(user.getUserName()); %>
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                    <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                    <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                    <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
                </div>
            </li>
            <li class="nav-item">
                <form action="/WebGymTracking/GymTrackingController" method="get">
                    <input type="hidden" name="command" value="Logout" >
                    <input class="btn btn-danger" type="submit" value="Logout">
                </form>
            </li>
        </ul>
```

```

    </div>
</nav>

<div class="container">
<h4>Please update your Password:</h4>
    <!-- Password update form -->
    <form action="/WebGymTracking/GymTrackingController" method="post">
    <h4 style="color:red; font-weight:bold;"><%if(request.getAttribute("message") != null)
        out.write((String)request.getAttribute("message")); %></h4>
        <input type="hidden" name="command" value="updatePassword">

            <!-- Input for old password -->
            <div class="form-group">
                <label for="password">Old Password:</label>
                <input type="password" class="form-control" name="oldPassword"
placeholder="Enter your old password">
            </div>

            <!-- Input for new password -->
            <div class="form-group">
                <label for="newPassword">New Password:</label>
                <input type="password" class="form-control" name="newPassword"
placeholder="Enter your new password">
            </div>

            <!-- Input for new re-password -->
            <div class="form-group">
                <label for="rePassword">Re-Password:</label>
                <input type="password" class="form-control" name="rePassword" placeholder="Re-
enter your new password">
                <br>
            </div>

            <div align="center">
                <input type="submit" class="btn btn-light" name="command" value="updatePassword">
            </div>
        <br>
        <a href="userMain.jsp" class="badge badge-info" role="button" aria-
pressed="true">Back to Main</a>
    </div>

    </form>
</div>
</body>
</html>

```

updateWeightAndHeight.jsp

```

<%@ page language="java" import="com.hit.model.*" contentType="text/html; charset=windows-1255"
    pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
    <script src="js/view/popper/popper.min.js"></script>
    <script src="js/view/bootstrap/bootstrap.min.js"></script>
    <title>Update Weight And Height</title>
</head>

<body>
<!-- toolbar of user -->
<nav class="navbar navbar-dark bg-dark">

```



```

        <h3 style="color:white;">GymTracking</h3>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="userActivities.jsp">MyActivity</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="search.jsp">Search</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="about.jsp">About</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <%User user = (User)session.getAttribute("loginUser");
out.println(user.getUserName()); %>
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                    <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                    <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                    <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
                </div>
            </li>
            <li class="nav-item">
                <form action="/WebGymTracking/GymTrackingController" method="get">
                    <input type="hidden" name="command" value="Logout" >
                    <input class="btn btn-danger" type="submit" value="Logout">
                </form>
            </li>
        </ul>
    </div>
</nav>

<div class="container">
    <h3>Please update your Weight and Height:</h3>
    <!-- Form of update the weight and height of the user -->
    <form action="/WebGymTracking/GymTrackingController" method="post">
        <%
int valueWeight = user.getWeight();//The weight of the user
int valueHeight = user.getHeight();//The height of the user
%>

        <!-- Input to change weight -->
        <div class="form-group">
            <label for="weight">Weight:</label>
            <div align="center">
                <input type="number" size="2" step="1" min="30" max="150" placeholder="<%=
valueWeight %>" id="inputWeight" >
            </div>
            <input type="range" id="sliderWeight" class="custom-range" name="newWeight"
value="<%= valueWeight %>" min="30" max="150" />
        </div>

        <!-- Input to change height -->

```



```

        <div class="form-group">
            <label for="height">Height:</label>
            <div align="center">
                <input type="number" size="2" step="1" min="100" max="250" placeholder="<%=
valueHeight %>" id="inputHeight" >
            </div>
            <input type="range" id="sliderHeight" class="custom-range" name="newHeight"
value="<%= valueHeight %>" min="100" max="250" />
        </div>

        <div align="center">
            <input type="submit" class="btn btn-light" name="command"
value="updateWeightAndHeight" ><br>
            <a href="userMain.jsp" class="badge badge-info" role="button" aria-pressed="true">Back
to Main</a>
        </div>
    </form>
</script>
var sliderWeight = document.getElementById("sliderWeight");
var sliderHeight = document.getElementById("sliderHeight");
var inputWeight = document.getElementById("inputWeight");
var inputHeight = document.getElementById("inputHeight");

//value updation from input to slider
//function input update to slider
function sliderWeightInput(){//input udate slider Weight
    sliderWeight.value = inputWeight.value;
}
function sliderHeightInput(){//input update slider Height
    sliderHeight.value = (inputHeight.value);
}

//calling function on change of inputs to update in slider
inputWeight.addEventListener("change",sliderWeightInput);
inputHeight.addEventListener("change",sliderHeightInput);

//value updation from slider to input
//functions to update from slider to inputs
function inputWeightSliderWeight(){//slider update inputs
    inputWeight.value = sliderWeight.value;
}
function inputHeightSliderHeight(){//slider update inputs
    inputHeight.value = sliderHeight.value;
}
sliderWeight.addEventListener("change",inputWeightSliderWeight);
sliderHeight.addEventListener("change",inputHeightSliderHeight);
</script>
</div>

</body>
</html>

```

updateUser.jsp

```

<%@ page language="java" import="com.hit.model.*" contentType="text/html; charset=windows-1255"
pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/bootstrap.min.css">
<script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
<script src="js/view/popper/popper.min.js"></script>
<script src="js/view/bootstrap/bootstrap.min.js"></script>
<script type="text/javascript" src="js/sliderUser.js"></script>
<title>Update User</title>
</head>

<body>
<!-- toolbar of user -->
<nav class="navbar navbar-dark bg-dark">
    <h3 style="color:white;">GymTracking</h3>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="true" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="userMain.jsp">Home<span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="userActivities.jsp">MyActivity</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="search.jsp">Search</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="about.jsp">About</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <!-- User user = (User)session.getAttribute("loginUser");
out.println(user.getUserName()); -->
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                    <a class="dropdown-item" href="updateUser.jsp">Update User</a>
                    <a class="dropdown-item" href="updateWeightAndHeight.jsp">Update Weight And
Height</a>
                    <a class="dropdown-item" href="updatePassword.jsp">Update Password</a>
                </div>
            </li>
            <li class="nav-item">
                <form action="/WebGymTracking/GymTrackingController" method="get">
                    <input type="hidden" name="command" value="logout" >
                    <input class="btn btn-danger" type="submit" value="logout">
                </form>
            </li>
        </ul>
    </div>
</nav>

<div class="container">
    <h2>Please update your information:</h2>
    <!-- User update form -->
    <form action="/WebGymTracking/GymTrackingController" method="post">
        <%
int valueWeight = user.getWeight();//The weight of the user
int valueHeight = user.getHeight();//The height of the user
%>

```

```

<h4 style="color:red; font-weight:bold;"><%if(request.getAttribute("message") != null)
    out.write((String)request.getAttribute("message")); %></h4>
<input type="hidden" name="command" value="updateUser">

    <!-- Input for old password -->
    <div class="form-group">
        <label for="password">Old Password:</label>
        <input type="password" class="form-control" name="oldPassword"
placeholder="Enter your old password">
    </div>

    <!-- Input for new password -->
    <div class="form-group">
        <label for="newPassword">New Password:</label>
        <input type="password" class="form-control" name="newPassword"
placeholder="Enter your new password">
    </div>

    <!-- Input for new re-password -->
    <div class="form-group">
        <label for="rePassword">Re-Password:</label>
        <input type="password" class="form-control" name="rePassword"
placeholder="Re-enter your new password">
    </div>
    <br>

    <!-- Input to change weight -->
    <div class="form-group">
        <label for="weight">Weight:</label>
        <div align="center">
            <input type="number" size="2" step="1" min="30" max="150" placeholder="<%=
valueWeight %>" id="inputWeight" >
        </div>
        <input type="range" id="sliderWeight" class="custom-range" name="newWeight"
value="<%= valueWeight %>" min="30" max="150" />
    </div>

    <!-- Input to change height -->
    <div class="form-group">
        <label for="height">Height:</label>
        <div align="center">
            <input type="number" size="2" step="1" min="100" max="250" placeholder="<%=
valueHeight %>" id="inputHeight" >
        </div>
        <input type="range" id="sliderHeight" class="custom-range" name="newHeight"
value="<%= valueHeight %>" min="100" max="250" />
    </div>

    <div align="center">
        <input type="submit" class="btn btn-light" name="command"
value="updateUser" ><br>
        <a href="userMain.jsp" class="badge badge-info" role="button" aria-
pressed="true">Back to Main</a>
    </div>

</form>
<script>
var sliderWeight = document.getElementById("sliderWeight");
var sliderHeight = document.getElementById("sliderHeight");
var inputWeight = document.getElementById("inputWeight");
var inputHeight = document.getElementById("inputHeight");

//value updation from input to slider
//function input update to slider

```

```

function sliderWeightInput(){//input udate slider Weight
    sliderWeight.value = inputWeight.value;
}
function sliderHeightInput(){//input update slider Height
    sliderHeight.value = (inputHeight.value);
}

//calling function on change of inputs to update in slider
inputWeight.addEventListener("change",sliderWeightInput);
inputHeight.addEventListener("change",sliderHeightInput);

///value updation from slider to input
//functions to update from slider to inputs
function inputWeightSliderWeight(){//slider update inputs
    inputWeight.value = sliderWeight.value;
}
function inputHeightSliderHeight(){//slider update inputs
    inputHeight.value = sliderHeight.value;
}
sliderWeight.addEventListener("change",inputWeightSliderWeight);
sliderHeight.addEventListener("change",inputHeightSliderHeight);
</script>
</div>

</body>
</html>

```

error.jsp

```

<%@ page language="java" contentType="text/html; charset=windows-1255"
    pageEncoding="windows-1255"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://fonts.googleapis.com/css?family=Montserrat:200,400,700"
rel="stylesheet">
    <link type="text/css" rel="stylesheet" href="css/style.css" />
    <script src="js/error/html5shiv.min.js"></script>
    <script src="js/error/respond.min.js"></script>
    <title>404 Error</title>
</head>

<!-- Body error --%>
<body>
<div id="notfound">
    <div class="notfound">
        <div class="notfound-404">
            <h1>Oops!</h1>
            <h2>404 - The Page can't be found</h2>
        </div>
    </div>
</div>
</body>

</html>

```

logout.jsp

```

<%@ page language="java" contentType="text/html; charset=windows-1255" pageEncoding="windows-
1255"%>
<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="utf-8">
<link rel="stylesheet" href="css/bootstrap.min.css">
<script src="js/view/jquery/jquery-3.3.1.slim.min.js"></script>
<script src="js/view/popper/popper.min.js"></script>
<script src="js/view/bootstrap/bootstrap.min.js"></script>
<title>Add Active</title>
</head>

<body>
<!-- A body of logout --%>
<div align="center">
    
    <h1 style="color:green; font-weight:bold;"><%if(request.getAttribute("message") != null)
        out.write((String)request.getAttribute("message")); %></h1>
    <h3>We hope you've had a good workout, and look forward to seeing you here in the next
workout.</h3>
    <br>
    <a class="btn btn-outline-success" href="Login.jsp" role="button">Login again</a>
</div>

</body>
</html>

```