

Music Store

Summary/Proposal:

- Design a website for artist to add albums and tracks and subscribers, create playlist and tracks.
- The website would be able to register users. The users could either be subscriber or artist.
- Artist can add an album to the Music Library that would be in the pending state, till the admin approves or rejects it.
- The Subscribers can add or delete a playlist. Playlist is a list of tracks.
- Subscribers would be able to search the tracks based on the album or artist name.

Functionalities:

- The users when registered, would get an email for activating their account.
- Artists would be able to add their albums or tracks (implemented using SimpleFormController).
- Session Management implemented
- Subscribers can search for any track and add it to his playlist

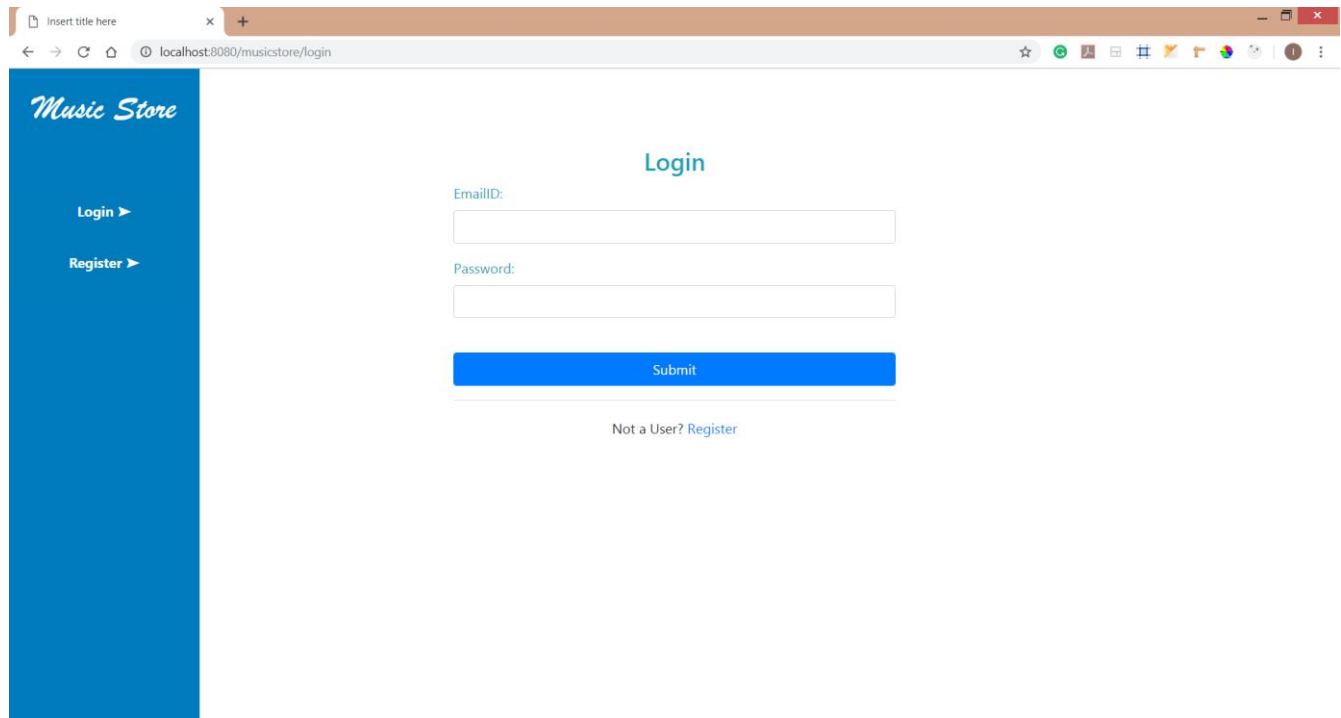
Technologies Used:

- JAVA Spring MVC
- Hibernate
- MS-SQL Database
- J-Query
- AJAX
- HTML5
- CSS3
- BOOTSTRAP
- JAVASCRIPT

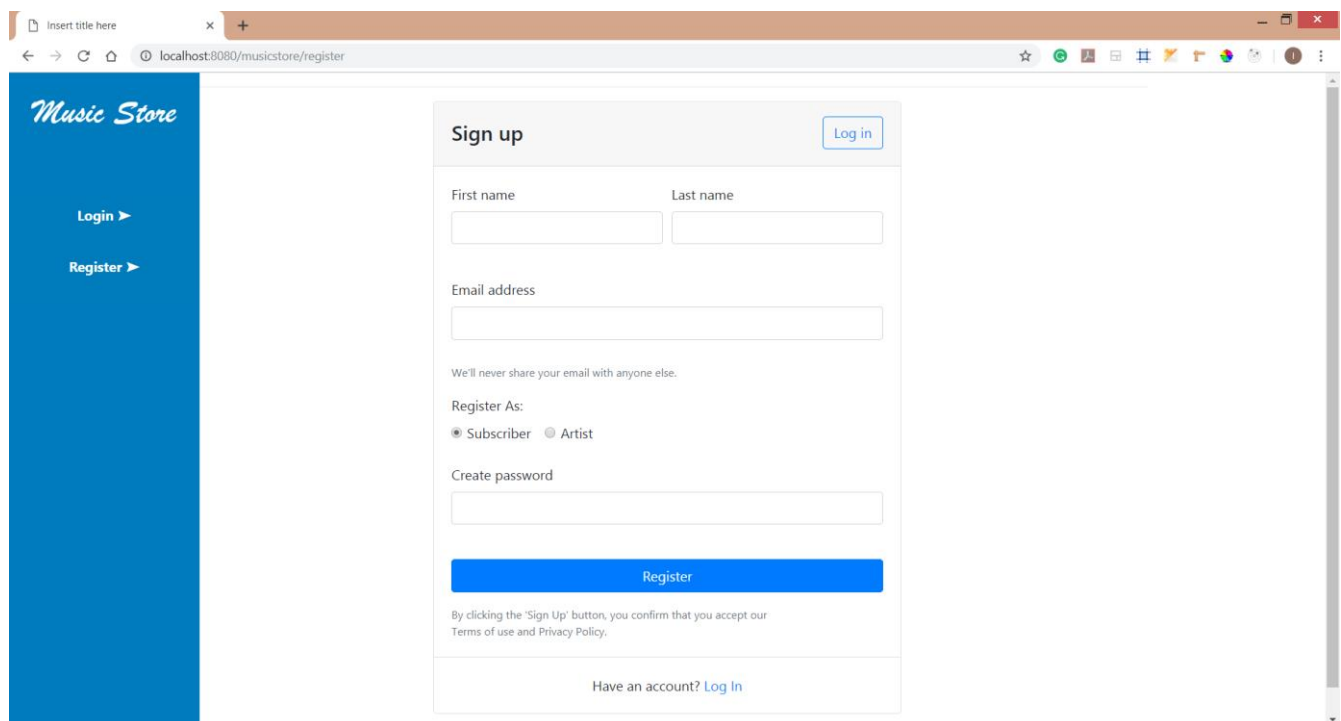
Roles and Respective Tasks

- Admin
 - Will approve/reject albums
 - View a generic report of Users
- Artist
 - Create and view the status of albums created by him
 - Can add track to the albums
- Subscriber
 - Can create a playlist
 - Can add tracks to the playlist created by him

Screenshots



A screenshot of a web browser displaying the 'Music Store' login page. The browser's address bar shows 'localhost:8080/musicstore/login'. On the left, a blue sidebar contains the 'Music Store' logo and two links: 'Login >' and 'Register >'. The main content area has a 'Login' heading. Below it are two input fields labeled 'EmailID:' and 'Password:'. A blue 'Submit' button is positioned below the password field. At the bottom, a link reads 'Not a User? Register'.



A screenshot of a web browser displaying the 'Music Store' registration page. The browser's address bar shows 'localhost:8080/musicstore/register'. The left sidebar is identical to the login page. The main content area features a 'Sign up' heading with a 'Log in' button in the top right corner. The registration form includes fields for 'First name', 'Last name', and 'Email address'. Below the email field is a note: 'We'll never share your email with anyone else.' The 'Register As:' section has two radio buttons: 'Subscriber' (selected) and 'Artist'. A 'Create password' field is also present. A blue 'Register' button is at the bottom of the form. Below the button, a disclaimer states: 'By clicking the "Sign Up" button, you confirm that you accept our Terms of use and Privacy Policy.' At the very bottom, a link reads 'Have an account? Log In'.

Insert title here x +

localhost:8080/musicstore/addTrack

Music Store

Home

Albums

New Track/Album

My Requests

Logout

Add New Album

Add New Track

Select Album

qwerty

Title

Audio File

Choose File No file chosen

Submit

Insert title here x +

localhost:8080/musicstore/viewalbums

Music Store

Home

Albums

New Track/Album

My Requests

Logout

My Albums

ALBUM NAME	STATUS	ACTION
mango	PENDING	VIEW TRACKS
qwerty	ACCEPTED	VIEW TRACKS

Insert title here

localhost:8080/musicstore/requeststatus

Music Store

Home

Albums

New Track/Album

My Requests

Logout

Request Status

mango	PENDING
qwerty	ACCEPTED

Insert title here

localhost:8080/musicstore/viewalbumtracks?albumid=3

Music Store

Home

Albums

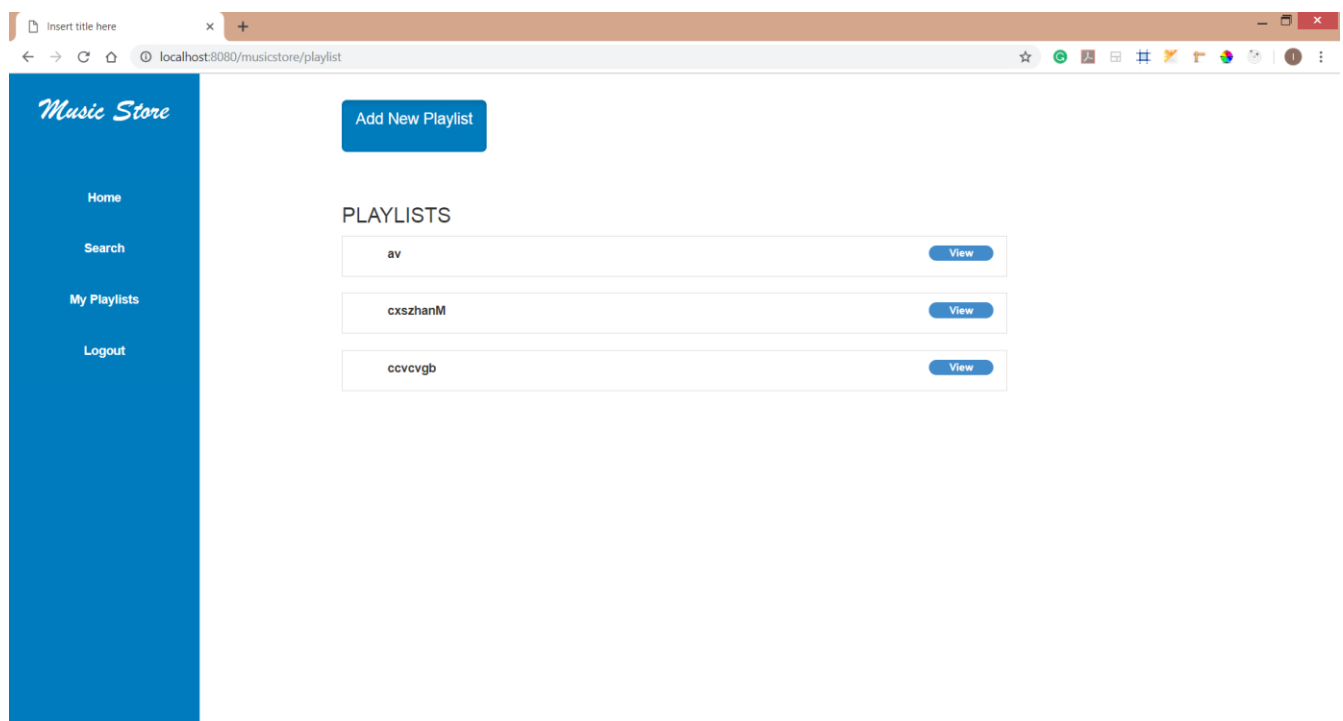
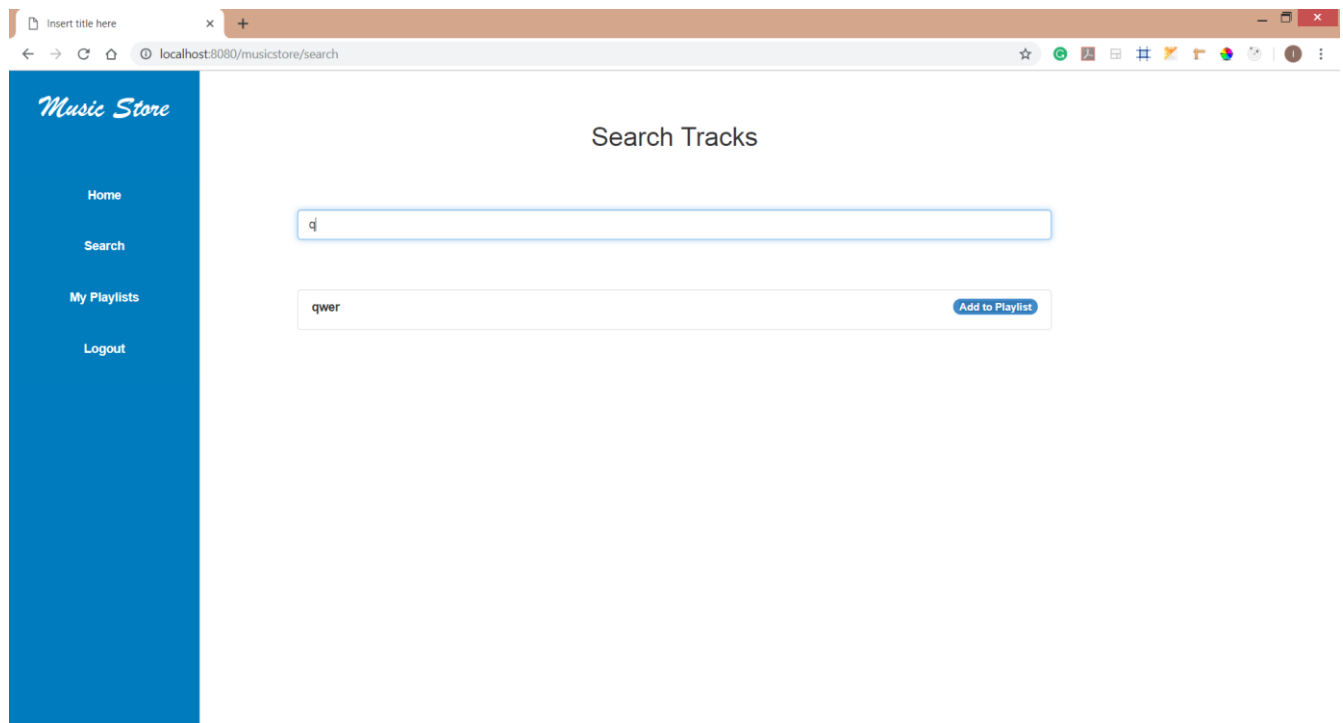
New Track/Album

My Requests

Logout

ALBUM-Tracks

Track Id	Title	Filename
4	qwer	web1.wav



Music Store

Home

Search

My Playlists

Logout

PLAYLIST-Tracks

Track Name	Album Name	Artist Name
qwer	qwerty	ishu awqs

Appendix:

Controllers:

1. AdminController

```
package com.music.musicstore.controllers;
```

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

```
import java.util.HashMap;
```

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

```
import java.util.Map;
```

```
import javax.servlet.http.HttpServletRequest;
```

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

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.servlet.ModelAndView;
```

```
import com.music.musicstore.dao.AlbumDAO;
```

```
import com.music.musicstore.dao.ArtistDao;
```

```
import com.music.musicstore.dao.SubscriberDAO;
```

```
import com.music.musicstore.pojo.Album;
```

```
import com.music.musicstore.pojo.Artist;
```

```
import com.music.musicstore.pojo.Subscriber;
```

```
import com.music.musicstore.pojo.User.UserRole;
```

```
@Controller
```

```
public class AdminController {
```

```
    @RequestMapping(method = RequestMethod.GET, value = "/viewartists")
```

```
    public String viewArtists(ArtistDao artistdao, ModelMap map) {
```

```
        ArrayList<Artist> artistList = artistdao.getAllArtist();
```

```
        System.out.println(artistList.size());
```

```
        map.put("display", UserRole.ARTIST.getRole());
```

```
        map.addAttribute("userList", artistList);
```

```
        return "viewusers";
```

```
    }
```

```
    @RequestMapping(method = RequestMethod.GET, value = "/viewsubscribers")
```

```
    public String viewSubscriber(SubscriberDAO subscriberdao, ModelMap map) {
```

```
        ArrayList<Subscriber> subscriberList = subscriberdao.getAllSubscriber();
```

```
        System.out.println(subscriberList.size());
```

```
        map.put("display", UserRole.SUBSCRIBER.getRole());
```

```
        map.addAttribute("userList", subscriberList);
        return "viewusers";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/viewalbumrequests")
    public String viewAlbumRequests(AlbumDAO albumdao, ModelMap map) {
        List<Album> albumlist = albumdao.getViewAlbumRequests();
        System.out.println(albumlist.size());
        map.addAttribute("albumlist", albumlist);
        return "viewalbumrequests";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/deletesubscriber")
    public String deleteSubscribers(SubscriberDAO subscriberdao, ModelMap map,
    HttpServletRequest request) {
        int id = Integer.parseInt(request.getParameter("user"));
        int count = subscriberdao.deleteSubscribers(id);
        map.put("display", UserRole.SUBSCRIBER.getRole());
        return "viewusers";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/deleteartist")
    public String deleteArtist(ArtistDao artistdao, ModelMap map, HttpServletRequest request) {
        int id = Integer.parseInt(request.getParameter("user"));
        int count = artistdao.deleteArtist(id);
        return "redirect:./viewartists";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/requestalbum")
    public String requestActions(AlbumDAO albumdao, HttpServletRequest request) {
        System.out.println("qwer");
        String status = request.getParameter("status");
        String albumid = request.getParameter("albumid");
        int count = albumdao.updateAlbumStatus(status, albumid);

        return "redirect:./viewalbumrequests";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/report")
    protected ModelAndView handleRequestInternal(HttpServletRequest request,
    HttpServletResponse response,
        ArtistDao artistdao, SubscriberDAO subscriberDAO) throws Exception {

        Map<String, Object> report = new HashMap<String, Object>();

        long numArtists = (Long) artistdao.getNumberOfArtists().get(0);
        report.put("numArtists", numArtists);

        List<Artist> artists = artistdao.getAllArtist();
```



```
        report.put("artists", artists);

        long numSubscribers = (Long) subscriberDAO.getNumberOfSubscribers().get(0);
        report.put("numSubscribers", numSubscribers);

        List<Subscriber> subscribers = subscriberDAO.getAllSubscriber();
        report.put("subscribers", subscribers);

        // return normal view
        return new ModelAndView(new PdfReportView(), "report", report);
    }
}
```

2. ArtistController

```
package com.music.musicstore.controllers;

import java.io.File;
import java.io.IOException;
import java.util.Calendar;
import java.util.List;
import java.util.Set;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.musicstore.dao.AlbumDAO;
import com.music.musicstore.dao.TrackDAO;
import com.music.musicstore.pojo.Album;
import com.music.musicstore.pojo.Artist;
import com.music.musicstore.pojo.Track;
import com.music.musicstore.pojo.User;

@Controller
public class ArtistController {

    @RequestMapping(value = "/viewalbums", method = RequestMethod.GET)
    public String viewalbums(HttpServletRequest request, AlbumDAO albumdao) {
        HttpSession session = request.getSession();
        User user = (Artist) session.getAttribute("user");
        Set<Album> albumsList = albumdao.getAlbumList(user.getEmail());
    }
}
```

```
        System.out.println(albumsList.size());
        if (albumsList != null)
            request.setAttribute("albumsList", albumsList);

        return "viewalbums";
    }

    @RequestMapping(value = "/viewtracks", method = RequestMethod.GET)
    public String viewtracks(HttpServletRequest request, AlbumDAO albumdao) {
        return "viewtracks";
    }

    @RequestMapping(value = "/addNew", method = RequestMethod.GET)
    public String addPage(HttpServletRequest request) {
        String type = request.getParameter("type");
        request.setAttribute("addType", "ALBUM");
        return "add";
    }

    @RequestMapping(value = "/add", method = RequestMethod.POST)
    public String add(HttpServletRequest request, AlbumDAO albumdao) {
        if (request.getParameter("addType").equals("ALBUM")) {
            Album album = new Album();
            Artist artist = (Artist) request.getSession().getAttribute("user");
            album.setAlbumName(request.getParameter("albumName"));
            album.setArtist((Artist) request.getSession().getAttribute("user"));
            artist.getAlbums().add(album);
            albumdao.addAlbum(album, artist);
        }

        return "redirect:./homepage";
    }

    @RequestMapping(value = "/addTrack", method = RequestMethod.GET)
    public String addTrackForm(HttpServletRequest request, AlbumDAO albumdao, Track track)
    {
        HttpSession session = request.getSession();
        User user = (Artist) session.getAttribute("user");
        List<Album> albums = albumdao.getAlbums(user.getEmail());
        if (albums != null)
            request.setAttribute("albums", albums);
        request.setAttribute("addType", "TRACK");
        return "add";
    }

    @RequestMapping(value = "/addTrack", method = RequestMethod.POST)
    public String addTrack(HttpServletRequest request, AlbumDAO albumdao,
    @ModelAttribute("track") Track track,
        TrackDAO trackDAO) {
```

```
        if (request.getParameter("addType").equals("TRACK")) {
            System.out.println("Here");
            String localPath = "D:\\Web Tools\\ProjectFiles";
            String albumId = null;
            if ((albumId = request.getParameter("albumId")) != null) {
                track.setFileName(Calendar.getInstance().getTimeInMillis() +
track.getMusic().getOriginalFilename());
                File file = new File(localPath, track.getFileName());
                try {
                    track.getMusic().transferTo(file);
                } catch (IllegalStateException e) {
                    System.out.println(e.getMessage());
                } catch (IOException e) {
                    System.out.println(e.getMessage());
                }
                Album album = albumdao.getAlbum(albumId);
                album.getTracks().add(track);
                track.setAlbum(album);
                trackDAO.addTrack(track, album);
            } else {
                System.out.println("Track not uploaded");
            }
        }
        return "redirect:./homepage";
    }
}
```

```
@RequestMapping(value = "/requeststatus", method = RequestMethod.GET)
public String requeststatus(HttpServletRequest request, AlbumDAO albumdao) {
    HttpSession session = request.getSession();
    User user = (Artist) session.getAttribute("user");
    Set<Album> albumsList = albumdao.getPendingAlbumList(user.getEmail());
    System.out.println(albumsList.size());
    if (albumsList != null)
        request.setAttribute("albumsList", albumsList);
    return "requeststatus";
}
```

```
@RequestMapping(value = "/add", method = RequestMethod.GET)
public String addNew(HttpServletRequest request) {
    return "add";
}
```

```
@RequestMapping(value = "/viewalbumtracks", method = RequestMethod.GET)
public String viewalbumtracks(HttpServletRequest request, AlbumDAO albumdao) {
    int album_id = Integer.parseInt(request.getParameter("albumid"));
    List<Track> tracks = albumdao.getViewalbumtracks(album_id);
    request.setAttribute("tracks", tracks);
    return "/viewalbumtracks";
}
```

```
}  
  
}
```

3. PdfReportView

```
package com.music.musicstore.controllers;  
  
import java.util.List;  
import java.util.Map;  
  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
import org.springframework.web.servlet.view.document.AbstractPdfView;  
  
import com.lowagie.text.Chunk;  
import com.lowagie.text.Document;  
import com.lowagie.text.Paragraph;  
import com.lowagie.text.Table;  
import com.lowagie.text.pdf.PdfWriter;  
import com.music.musicstore.pojo.Artist;  
import com.music.musicstore.pojo.Subscriber;  
  
public class PdfReportView extends AbstractPdfView {  
  
    @SuppressWarnings("unchecked")  
    @Override  
    protected void buildPdfDocument(Map model, Document document, PdfWriter writer,  
HttpServletRequest request,  
HttpServletResponse response) throws Exception {  
  
        Map<String, Object> report = (Map<String, Object>) model.get("report");  
  
        Table tableArtist = new Table(2);  
        tableArtist.addCell("Artist");  
        tableArtist.addCell("Number of Albums");  
  
        List<Artist> artists = (List<Artist>) report.get("artists");  
  
        for (Artist artist : artists) {  
            tableArtist.addCell(artist.getFname() + " " + artist.getLname());  
            tableArtist.addCell(String.valueOf(artist.getAlbums().size()));  
        }  
  
        Table tableSubscriber = new Table(2);  
        tableSubscriber.addCell("Subscriber");  
        tableSubscriber.addCell("Number of Playlist");  
    }  
}
```

```
List<Subscriber> subscribers = (List<Subscriber>) report.get("subscribers");

for (Subscriber subscriber : subscribers) {
    tableSubscriber.addCell(subscriber.getFname() + " " + subscriber.getLname());
    tableSubscriber.addCell(String.valueOf(subscriber.getPlayList().size()));
}

Paragraph p = new Paragraph(new Chunk("Number of Artists : " +
report.get("numArtists")));

document.add(p);
document.add(tableArtist);
p = new Paragraph(new Chunk("\n "));
p = new Paragraph(new Chunk("Number of Subscribers : " +
report.get("numSubscribers")));
document.add(p);
document.add(tableSubscriber);
}
}
```

4. RegisterUserController

```
package com.music.musicstore.controllers;

import java.util.Random;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.SimpleEmail;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.musicstore.dao.UserDAO;
import com.music.musicstore.pojo.User;
import com.music.musicstore.validator.UserValidator;
```

```
@Controller
@RequestMapping(value = "register")
public class RegisterUserController {
    @Autowired
    UserValidator uservalidator;

    @Autowired
    UserDAO userdao;

    // Set a form validator
    @InitBinder
    protected void initBinder(WebDataBinder binder) {
        binder.setValidator(uservalidator);
    }

    @RequestMapping(method = RequestMethod.GET)
    public String formView(ModelMap model, User user) {
        model.addAttribute("user", user);
        return "register";
    }

    @RequestMapping(method = RequestMethod.POST)
    public String successView(@Validated @ModelAttribute("user") User user, BindingResult
bindingResult,
        HttpServletRequest request) {
        if (bindingResult.hasErrors()) {
            return "register"; // the are validation errors, go to the form view
        }
        // no errors, so go to the success view
        if (userdao.getUser(user.getEmail()) != null) {
            request.setAttribute("error", "Email address already present!!");
            return "register";
        }
        HttpSession session = request.getSession();
        user.setPassword(String.valueOf(user.getPassword().hashCode()));
        user = userdao.create(user);
        Random rand = new Random();
        int randomNum1 = rand.nextInt(5000000);
        int randomNum2 = rand.nextInt(5000000);
        try {
            String str = "http://localhost:8080/musicstore/validateemail.htm?email=" +
user.getEmail() + "&key1="
                + randomNum1 + "&key2=" + randomNum2;
            session.setAttribute("key1", randomNum1);
            session.setAttribute("key2", randomNum2);
            sendEmail(user.getEmail(), "Click on this link to activate your account : " + str);
        } catch (Exception e) {
            System.out.println("Email cannot be sent");
        }
    }
}
```

```
    }

    request.setAttribute("registration_success",
        "Registration Successful!! Please check your email for activation link!!");
    return "home";
}

public void sendEmail(String useremail, String message) {
    try {
        Email email = new SimpleEmail();
        email.setHostName("smtp.googlemail.com");
        email.setSmtPort(465);
        email.setAuthenticator(new DefaultAuthenticator("wt.musicstore@gmail.com",
            "wt-mstr@123"));
        email.setSSLonConnect(true);
        email.setFrom("no-reply@msis.neu.edu"); // This user email does not
        exist

        email.setSubject("Password Reminder");
        email.setMsg(message); // Retrieve email from the DAO and send this
        email.addTo(useremail);
        email.send();
    } catch (EmailException e) {
        System.out.println(e);
    }
}

@RequestMapping(value = "resendemail.htm", method = RequestMethod.POST)
public String resendEmail(HttpServletRequest request) {
    HttpSession session = request.getSession();
    String useremail = (String) request.getAttribute("username");
    Random rand = new Random();
    int randomNum1 = rand.nextInt(5000000);
    int randomNum2 = rand.nextInt(5000000);
    try {
        String str = "http://localhost:8080/musicstore/validateemail.htm?email=" +
            useremail + "&key1=" + randomNum1
                + "&key2=" + randomNum2;
        session.setAttribute("key1", randomNum1);
        session.setAttribute("key2", randomNum2);
        sendEmail(useremail, "Click on this link to activate your account : " + str);
    } catch (Exception e) {
        System.out.println("Email cannot be sent");
    }

    return "user-created";
}
}
```

5. SearchController

```
package com.music.musicstore.controllers;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.music.musicstore.dao.AlbumDAO;
import com.music.musicstore.dao.TrackDAO;
import com.music.musicstore.pojo.Track;

@Controller
public class SearchController {

    @RequestMapping(value = "/searchTrack", method = RequestMethod.GET, produces =
"application/json")
    @ResponseBody
    public String searchArtist(HttpServletRequest request, TrackDAO trackdao, AlbumDAO
albumdao)
        throws JsonProcessingException {
        String param = request.getParameter("search");
        List<Track> tracks = trackdao.getTracks(param);
        ObjectMapper obj = new ObjectMapper();
        return obj.writeValueAsString(tracks);
    }
}
```

6. SubscriberController

```
package com.music.musicstore.controllers;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;
```



```
import javax.servlet.http.HttpServletRequest;

import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.musicstore.dao.PlayListDAO;
import com.music.musicstore.dao.SubscriberDAO;
import com.music.musicstore.dao.TrackDAO;
import com.music.musicstore.pojo.PlayList;
import com.music.musicstore.pojo.Subscriber;
import com.music.musicstore.pojo.Track;
import com.music.musicstore.pojo.User;

@Controller
public class SubscriberController {

    @RequestMapping(value = "/search", method = RequestMethod.GET)

    public String searchTrack() {
        return "search";
    }

    @RequestMapping(value = "playlist", method = RequestMethod.GET)
    public String playlist(HttpServletRequest request, PlayListDAO playListDAO) {
        request.setAttribute("addType", "VIEW_PLAYLIST");
        List<PlayList> playlists = playListDAO
            .getPlaylists(((User)
request.getSession().getAttribute("user")).getEmail());
        if (playlists == null) {
            playlists = new ArrayList<PlayList>();
        }
        request.setAttribute("playlists", playlists);
        return "playlist";
    }

    @RequestMapping(value = "/addNewPlaylist", method = RequestMethod.GET)
    public String addPlaylist(HttpServletRequest request) {
        request.setAttribute("addType", "PLAYLIST");
        return "playlist";
    }

    @RequestMapping(value = "/addNewPlaylist", method = RequestMethod.POST)
    public String addNewPlaylist(HttpServletRequest request, PlayListDAO playlistdao) {
        // User user = (User) request.getSession().getAttribute("user");
        PlayList playlist = new PlayList();
        Subscriber subscriber = (Subscriber) request.getSession().getAttribute("user");
```

```
        boolean present = false;
        for (Playlist p : subscriber.getPlaylist()) {
            if
(p.getPlaylistName().equalsIgnoreCase(request.getParameter("playlistName"))) {
                present = true;
                break;
            }
        }
        if (!present) {
            playlist.setPlaylistName(request.getParameter("playlistName"));
            playlist.setSubscriber(subscriber);
            subscriber.getPlaylist().add(playlist);
            playlistdao.addPlaylist(playlist, subscriber);
            return "redirect:./playlist";
        } else {
            request.setAttribute("errorMessage", "Playlist with same name already
present!!");
            request.setAttribute("addType", "PLAYLIST");
            return "playlist";
        }
    }

    @RequestMapping(value = "/deletePlaylist", method = RequestMethod.GET)
    public String deletePlaylist(HttpServletRequest request, PlaylistDAO playlistdao) {
        request.setAttribute("addType", "VIEW_PLAYLIST");
        int id = Integer.parseInt(request.getParameter("playlistid"));
        int count = playlistdao.deletePlaylist(id);
        return "playlist";
    }

    @RequestMapping(value = "/accountsettings", method = RequestMethod.GET)
    public String accountSettings() {
        return "accountsettings";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/playlisttracks")
    public String requestActions(PlaylistDAO playlistdao, HttpServletRequest request) {
        System.out.println("qwergbhjknsm");
        String playlistid = request.getParameter("playlistid");
        Set<Track> tracks = playlistdao.playlisttracks(playlistid);
        request.setAttribute("tracks", tracks);
        System.out.println(playlistid);
        request.setAttribute("playlistid", playlistid);
        return "viewtracks";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/addtrackstoplaylist")
    public String addtrackstoplaylist(TrackDAO trackdao, HttpServletRequest request, ModelMap
map,
```

```
SubscriberDAO subscriberdao) {
    System.out.println(request.getParameter("trackid"));

    Track track = trackdao.getTrack(Integer.parseInt(request.getParameter("trackid")));
    System.out.println(track.getTitle());
    map.put("track", track);

    Subscriber subscriber = subscriberdao
        .getSubscriber(((User)
request.getSession().getAttribute("user")).getEmail());
    map.put("playlists", subscriber.getPlayList());

    return "addtrackstoplaylist";
}

@RequestMapping(method = RequestMethod.GET, value = "/addToPlylist")
public String addtoplaylist(TrackDAO trackdao, HttpServletRequest request, ModelMap map,
PlayListDAO playListDAO) {
    System.out.println("qwergbhjknsm");

    Track track = trackdao.getTrack(Integer.parseInt(request.getParameter("trackid")));
    PlayList playList =
playListDAO.getplaylist(Integer.parseInt(request.getParameter("playlist")));

    if (track != null && playList != null) {
        playList.getTracks().add(track);
    }
    playListDAO.updatePlayList(playList);
    return "redirect:./playlist";
}

@RequestMapping(method = RequestMethod.GET, value = "/deleteTrackFromPlaylist")
public String deleteTrackFromPlaylist(TrackDAO trackdao, HttpServletRequest request,
ModelMap map,
    PlayListDAO playListDAO) {
    System.out.println(request.getParameter("playlistid"));

    Track track = trackdao.getTrack(Integer.parseInt(request.getParameter("trackid")));
    PlayList playList =
playListDAO.getplaylist(Integer.parseInt(request.getParameter("playlistid")));

    if (track != null && playList != null) {
        playList.getTracks().remove(track);
    }
    playListDAO.updatePlayList(playList);
    return "redirect:./playlist";
}
}
```

7. UserController

```
package com.music.musicstore.controllers;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.music.musicstore.dao.UserDAO;
import com.music.musicstore.pojo.User;

@Controller
public class UserController {

    @Autowired
    UserDAO userdao;

    @RequestMapping(method = RequestMethod.GET, value = "/login")
    public String login() {
        return "home";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/home")
    public String home() {
        return "home";
    }

    @RequestMapping(method = RequestMethod.GET, value = "/homepage")
    public String homepage(HttpServletRequest request) {
        User user = (User) request.getSession().getAttribute("user");
        System.out.println("here");
        if (user == null) {
            return "redirect:./home";
        } else {
            if (user.getRole().equals(User.UserRole.ADMIN.getRole())) {
                return "admin-view";
            } else if (user.getRole().equals(User.UserRole.ARTIST.getRole())) {
```

```

        return "artist-view";
    } else {
        return "subscriber-view";
    }
}

}

@RequestMapping(method = RequestMethod.POST, value = "/home")
public String loginandRegister(HttpServletRequest request, HttpServletResponse response)
throws IOException {
    HttpSession session = request.getSession();
    System.out.println("inside user login");
    String email = (String) request.getAttribute("email");
    String password = (String) request.getAttribute("password");
    System.out.println(password);
    password = String.valueOf(password.hashCode());
    User user = userdao.login(email, password);
    if (user == null) {
        request.setAttribute("login_error", "Invalid Credentials");
        return "home";
    } else {
        if (user.getStatus() == 0) {
            request.setAttribute("login_error",
                "Please activate the account using the link sent to your
email first!!");
            return "home";
        }
        session.setAttribute("user", user);
        if (user.getRole().equals(User.UserRole.ADMIN.getRole())) {
            return "admin-view";
        } else if (user.getRole().equals(User.UserRole.ARTIST.getRole())) {
            return "artist-view";
        } else {
            return "subscriber-view";
        }
    }
}
}

```

```

@RequestMapping(value = "validateemail.htm", method = RequestMethod.GET)
public String validateEmail(HttpServletRequest request, ModelMap map) {

    System.out.println("Herererererere");
    // The user will be sent the following link when the use registers
    // This is the format of the email
    //
    http://hostname:8080/musicstore/validateemail.htm?email=useremail&key1=<random_number>&key2
=<body
    // of the email that when user registers>
    HttpSession session = request.getSession();

```

```
String email = (String) request.getParameter("email");
int key1 = Integer.parseInt((String) request.getParameter("key1"));
int key2 = Integer.parseInt((String) request.getParameter("key2"));
System.out.println(session.getAttribute("key1"));
System.out.println(session.getAttribute("key2"));

if ((Integer) (session.getAttribute("key1")) == key1 && ((Integer)
session.getAttribute("key2")) == key2) {
    try {
        System.out.println("HI_____");
        boolean updateStatus = userDao.updateUser(email);
        if (updateStatus) {
            return "home";
        } else {
            return "error";
        }
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
} else {
    map.addAttribute("errorMessage", "Link expired , generate new link");
    map.addAttribute("resendLink", true);
    return "error";
}

return "home";

}

@RequestMapping(value = "/logout", method = RequestMethod.GET)
public String logoutUser(HttpServletRequest request) {
    request.getSession().removeAttribute("user");
    return "home";
}

}
```

DAO:

1. AlbumDAO

```
package com.music.musicstore.dao;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
```

```
import java.util.Set;

import org.hibernate.Criteria;
import org.hibernate.Query;
import org.hibernate.criterion.Restrictions;

import com.music.musicstore.pojo.Album;
import com.music.musicstore.pojo.Artist;
import com.music.musicstore.pojo.Track;

public class AlbumDAO extends DAO {

    public Album addAlbum(Album album, Artist artist) {

        try {
            begin();
            getSession().save(album);
            getSession().save(artist);
            commit();
        } catch (Exception e) {
            System.out.println(e);
        }
        close();
        return album;
    }

    @SuppressWarnings("deprecation")
    public List<Album> getAlbums(String email) {

        try {
            begin();

            Criteria criteriaArtist = getSession().createCriteria("artist");
            Criteria criteriaAlbum = getSession().createCriteria(Album.class);
            criteriaArtist.add(Restrictions.eq("email", email));
            criteriaAlbum.add(Restrictions.eq("status", "ACCEPTED"));

            List<Album> list = criteriaAlbum.list();
            close();
            if (list.size() == 0) {
                return new ArrayList<Album>();
            }
            return list;
        } catch (Exception e) {
            System.out.println(e);
            close();
        }

        return null;
    }
}
```

```
}

public Album getAlbum(String albumId) {
    begin();
    Query q = getSession().createQuery("from Album where album_id = :albumId ");
    q.setParameter("albumId", Integer.parseInt(albumId));
    Album album = (Album) q.uniqueResult();
    close();
    return album;
}

public Set<Album> getAlbumList(String email) {
    try {
        begin();
        Query query = getSession().createQuery(" from Artist where email
=:email").setParameter("email", email);
        query.setMaxResults(1);
        List<Artist> list = query.list();
        close();
        if (list.size() == 0) {
            return new HashSet<Album>();
        }
        return list.get(0).getAlbums();
    } catch (Exception e) {
        System.out.println(e);
        close();
    }

    return null;
}

public Set<Album> getPendingAlbumList(String email) {
    try {
        begin();
        Criteria crit = getSession().createCriteria(Artist.class);
        Criteria albumcriteria = crit.createCriteria("albums");
        crit.add(Restrictions.eq("email", email));
        albumcriteria.add(Restrictions.eq("status", "PENDING"));
        crit.setMaxResults(1);
        List<Artist> list = crit.list();
        close();
        if (list.size() == 0) {
            return new HashSet<Album>();
        }
        return list.get(0).getAlbums();
    } catch (Exception e) {
        System.out.println(e);
        close();
    }
}
```



```
        return null;
    }

    public List<Album> getviewalbumrequests() {
        try {
            begin();
            Query q = getSession().createQuery("from Album where
status=:status").setParameter("status", "PENDING");
            List<Album> album = q.list();
            close();
            if (album.size() == 0) {
                return new ArrayList();
            }
            return album;
        } catch (Exception e) {
            System.out.println(e);
            close();
        }
        return null;
    }

    public int updateAlbumStatus(String status, String albumid) {
        // TODO Auto-generated method stub
        int count = 0;
        try {
            begin();
            Query q = getSession().createQuery("update Album set status=:status where
album_id=:albumid")
                .setParameter("status", status);
            q.setParameter("albumid", Integer.parseInt(albumid));
            count = q.executeUpdate();
            System.out.println("count" + count);
            commit();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            close();
        }
        return count;
    }

    public List<Track> getviewalbumtracks(int album_id) {
        try {
            begin();
            Query q = getSession().createQuery("from Track");
            List<Track> tracks = q.list();
```

```

        List<Track> albumTracks = new ArrayList();

        for (Track t : tracks) {
            if (t.getAlbum().getAlbum_id() == album_id) {
                albumTracks.add(t);
            }
        }
        close();
        return albumTracks;
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }

    return null;
    // return tracks;
}
}

```

2. ArtistDAO

```
package com.music.musicstore.dao;
```

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

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

```
import org.hibernate.Criteria;
```

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

```
import org.hibernate.criterion.Projections;
```

```
import com.music.musicstore.pojo.Artist;
```

```
public class ArtistDao extends DAO {
```

```
    @SuppressWarnings({ "deprecation", "unchecked" })
```

```
    public ArrayList<Artist> getAllArtist() {
```

```
        ArrayList<Artist> artistList = new ArrayList<Artist>();
```

```
        try {
```

```
            begin();
```

```
            Query q = getSession().createQuery("from Artist");
```

```
            artistList = (ArrayList<Artist>) q.list();
```

```
            System.out.println(artistList.size() + "fgbhnmxmncfb");
```

```
            close();
```

```
        } catch (Exception e) {
```

```
            System.out.println(e);
```

```
            close();
```

```
        }
```

```

        return artistList;
    }

    public int deleteartist(int id) {
        int count = 0;
        try {
            begin();
            Query q = getSession().createQuery("delete from Artist where
id=:id").setParameter("id", id);
            count = q.executeUpdate();
        } catch (Exception e) {
        } finally {
            close();
        }
        return count;
    }

    public List getNumberOfArtists() {
        List result = null;
        try {
            begin();
            Criteria crit = getSession().createCriteria(Artist.class);
            crit.setProjection(Projections.rowCount());
            result = crit.list();
            System.out.println("Result : " + result);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            close();
        }
        return result;
    }
}

```

3. DAO

```

package com.music.musicstore.dao;

import java.util.logging.Level;
import java.util.logging.Logger;

import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class DAO {

```

```
private static final Logger log = Logger.getAnonymousLogger();

private static final ThreadLocal sessionThread = new ThreadLocal();
private static final SessionFactory sessionFactory = new
Configuration().configure("hibernate.cfg.xml")
                .buildSessionFactory();

protected DAO() {
}

public static Session getSession() {

    Session session = (Session) DAO.sessionThread.get();

    if (session == null) {
        session = sessionFactory.openSession();
        DAO.sessionThread.set(session);
    }
    return session;
}

public static void begin() {
    getSession().beginTransaction();
}

public static void commit() {
    getSession().getTransaction().commit();
}

protected void rollback() {
    try {
        getSession().getTransaction().rollback();
    } catch (HibernateException e) {
        log.log(Level.WARNING, "Cannot rollback", e);
    }
    try {
        getSession().close();
    } catch (HibernateException e) {
        log.log(Level.WARNING, "Cannot close", e);
    }
    DAO.sessionThread.set(null);
}

public static void close() {
    getSession().close();
    DAO.sessionThread.set(null);
}
}
```

4. PlayListDAO

```
package com.music.musicstore.dao;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;

import org.hibernate.Criteria;
import org.hibernate.criterion.Restrictions;
import org.hibernate.query.Query;

import com.music.musicstore.pojo.PlayList;
import com.music.musicstore.pojo.Subscriber;
import com.music.musicstore.pojo.Track;

public class PlayListDAO extends DAO {

    public PlayList addPlaylist(PlayList playlist, Subscriber subscriber) {

        try {
            begin();
            getSession().save(playlist);
            getSession().save(subscriber);
            commit();
        } catch (Exception e) {
            System.out.println(e);
        }
        close();
        return playlist;
    }

    @SuppressWarnings("deprecation")
    public List<PlayList> getPlaylists(String email) {

        try {
            begin();
            Criteria criteria = getSession().createCriteria(PlayList.class);
            Criteria criteriaSubscriber = getSession().createCriteria("subscriber");

            criteriaSubscriber.add(Restrictions.eq("email", email));

            List<PlayList> list = criteria.list();
            close();
            if (list.size() == 0) {
                return new ArrayList<PlayList>();
            }
            return list;
        }
    }
}
```

```
        } catch (Exception e) {
            System.out.println(e);
            close();
        }

        return null;
    }

    public int deletePlaylist(int id) {
        int count = 0;
        try {
            begin();
            Query q = getSession().createQuery("delete from PlayList where
playlist_Id=:id").setParameter("id", id);
            count = q.executeUpdate();
            commit();
        } catch (Exception e) {
        } finally {
            close();
        }
        return count;
    }

    public Set<Track> playlisttracks(String playlistid) {
        Set<Track> tracks = null;
        try {
            Query q = getSession().createQuery("from PlayList where
playlist_Id=:playlistid").setParameter("playlistid",
Integer.parseInt(playlistid));
            tracks = ((PlayList) q.uniqueResult()).getTracks();
        } catch (Exception e) {
            System.out.println(e);
        } finally {
            close();
        }
        return tracks;
    }

    public PlayList getplaylist(int playlistid) {
        PlayList playList = null;
        try {
            Query q = getSession().createQuery("from PlayList where
playlist_Id=:playlistid").setParameter("playlistid",
playlistid);
            playList = (PlayList) q.uniqueResult();
        } catch (Exception e) {
            System.out.println(e);
        } finally {
            close();
        }
    }
```

```
    }  
    return playList;  
}  
  
public void updatePlayList(PlayList playlist) {  
    try {  
        begin();  
        getSession().saveOrUpdate(playlist);  
        commit();  
    } catch (Exception e) {  
        System.out.println(e);  
    } finally {  
        close();  
    }  
}  
  
}
```

5. SubscriberDAO

```
package com.music.musicstore.dao;
```

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

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

```
import org.hibernate.Criteria;
```

```
import org.hibernate.criterion.Projections;
```

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

```
import com.music.musicstore.pojo.Subscriber;
```

```
public class SubscriberDAO extends DAO {
```

```
    @SuppressWarnings({ "deprecation", "unchecked" })
```

```
    public ArrayList<Subscriber> getAllSubscriber() {
```

```
        ArrayList<Subscriber> subscriberList = new ArrayList<Subscriber>();
```

```
        try {
```

```
            begin();
```

```
            Query<Subscriber> q = getSession().createQuery("from Subscriber");
```

```
            subscriberList = (ArrayList<Subscriber>) q.list();
```

```
            close();
```

```
        } catch (Exception e) {
```

```
            System.out.println(e);
```

```
            close();
```

```
        }
```

```
        return subscriberList;
```

```
    }
```

```
public int deleteSubscribers(int id) {
    int count = 0;
    try {
        begin();
        Query q = getSession().createQuery("delete from PlayList where
id=:id").setParameter("id", id);
        count = q.executeUpdate();
        commit();
    } catch (Exception e) {
    } finally {
        close();
    }
    return count;
}

public Subscriber getSubscriber(String email) {
    // TODO Auto-generated method stub
    Subscriber subscriber = null;
    try {
        begin();
        Query query = getSession().createQuery("from Subscriber where
email=:email");
        query.setParameter("email", email);
        subscriber = (Subscriber) query.uniqueResult();
    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        close();
    }
    return subscriber;
}

public List getNumberOfSubscribers() {
    List result = null;
    try {
        begin();
        Criteria crit = getSession().createCriteria(Subscriber.class);
        crit.setProjection(Projections.rowCount());
        result = crit.list();
        System.out.println("Result : " + result);
    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        close();
    }
    return result;
}
```


6. TrackDAO

```
package com.music.musicstore.dao;

import java.util.List;

import org.hibernate.Query;

import com.music.musicstore.pojo.Album;
import com.music.musicstore.pojo.Track;

public class TrackDAO extends DAO {

    public Track addTrack(Track track, Album album) {

        try {
            begin();
            getSession().saveOrUpdate(album);
            getSession().save(track);
            commit();
        } catch (Exception e) {
            System.out.println(e);
        }
        close();
        return track;
    }

    public List<Track> getTracks(String param) {
        List<Track> tracks = null;
        try {
            begin();
            Query q = getSession().createQuery("from Track where title like '%" + param +
"%");

            // q.setParameter("title", "%" + param + "%");
            System.out.println(q.toString());
            tracks = q.list();

        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            close();
        }
        return tracks;
    }

    public Track getTrack(int parameter) {
        // TODO Auto-generated method stub
        Track track = null;
        try {
```

```
        begin();
        Query q = getSession().createQuery("from Track where track_Id=:trackid");
        q.setParameter("trackid", parameter);
        track = (Track) q.uniqueResult();
    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        close();
    }
    return track;
}
}
```

7. UserDAO

```
package com.music.musicstore.dao;
```

```
import org.hibernate.Criteria;
import org.hibernate.HibernateException;
import org.hibernate.Query;
import org.hibernate.criterion.Restrictions;
```

```
import com.music.musicstore.pojo.Artist;
import com.music.musicstore.pojo.Subscriber;
import com.music.musicstore.pojo.User;
import com.music.musicstore.pojo.User.UserRole;
```

```
@SuppressWarnings("deprecation")
```

```
public class UserDAO extends DAO {
```

```
    public User getUser(String email) {
        User user1 = null;
        try {
```

```
            begin();
```

```
            Query q = getSession().createQuery("from User where email = '" + email + "'");
```

```
            user1 = (User) q.uniqueResult();
```

```
            if (user1 == null) {
```

```
                Criteria crit = getSession().createCriteria(Artist.class);
```

```
                crit.add(Restrictions.eq("email", email));
```

```
                crit.setMaxResults(1);
```

```
                user1 = (User) crit.uniqueResult();
```

```
            }
```

```
            if (user1 == null) {
```

```
                Criteria crit = getSession().createCriteria(Subscriber.class);
```

```
                crit.add(Restrictions.eq("email", email));
```

```
                crit.setMaxResults(1);
```

```
                user1 = (User) crit.uniqueResult();
```

```
            }
```

```
        } catch (Exception e) {
            rollback();
            System.out.println(e.getMessage());
        } finally {
            close();
        }
        return user1;
    }

    // to save user when registered
    public User create(User user) {
        try {
            begin();
            if (user.getRole().equals(UserRole.ARTIST.getRole())) {
                Artist artist = new Artist(user);
                getSession().save(artist);
            } else {
                if (user.getRole().equals(UserRole.SUBSCRIBER.getRole())) {
                    Subscriber subscriber = new Subscriber(user);
                    getSession().save(subscriber);
                }
            }
            commit();
        } catch (Exception e) {
            rollback();
        } finally {
            close();
        }

        return user;
    }

    public User login(String email, String password) {
        User user = null;
        try {
            user = this.getUser(email);
            if (!user.getPassword().equals(password)) {
                user = null;
            }
        } catch (Exception e) {
            rollback();
            System.out.println(e);
        } finally {
            close();
        }
        return user;
    }
}
```

```
@SuppressWarnings({ "rawtypes", "unused" })
public boolean updateUser(String email) throws Exception {
    try {
        begin();
        System.out.println(email);
        Query q = getSession().createQuery("from User where email = :email");
        q.setString("email", email);
        User user = (User) q.uniqueResult();
        System.out.println("ID: " + user.getId());
        if (user != null) {
            System.out.println("ABC-----");
            user.setStatus(1);
            getSession().update(user);
            commit();
            close();
            return true;
        }
    } catch (HibernateException e) {
        rollback();
        close();
        throw new Exception("Exception while creating user: " + e.getMessage());
    }
    return false;
}
}
```

Interceptor:

1. LoginInterceptor

```
package com.music.musicstore.interceptor;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.handler.HandlerInterceptorAdapter;

import com.music.musicstore.pojo.User;

public class LoginInterceptor extends HandlerInterceptorAdapter {

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
        Object handler)
        throws Exception {
```

```
String uri = request.getRequestURI();
User user = (User) request.getSession().getAttribute("user");

System.out.println(uri);

if (uri.endsWith("login") || uri.endsWith("register") || uri.endsWith(".htm")) {
    if (user == null) {
        return true;
    } else {
        response.sendRedirect("./homepage");
        System.out.println("here1");
        return false;
    }
} else {
    if (uri.endsWith("home")) {
        return true;
    } else {
        if (user == null) {
            response.sendRedirect("./login");
            return false;
        } else {
            System.out.println("here3");
            // response.sendRedirect("./homepage");
            if (uri.endsWith("musicstore/")) {
                response.sendRedirect("./login");
                return false;
            }
            return true;
        }
    }
}

}

}

@Override
public void postHandle(HttpServletRequest request, HttpServletResponse response,
Object handler,
        ModelAndView modelAndView) throws Exception {
    // TODO Auto-generated method stub

}

@Override
public void afterCompletion(HttpServletRequest request, HttpServletResponse
response, Object handler, Exception ex)
        throws Exception {
    // TODO Auto-generated method stub
```

```
}
```

```
}
```

Filter:

1. SanitizationFilter

```
package com.music.musicstore.filter;
```

```
import java.io.IOException;
```

```
import javax.servlet.Filter;
```

```
import javax.servlet.FilterChain;
```

```
import javax.servlet.FilterConfig;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.ServletRequest;
```

```
import javax.servlet.ServletResponse;
```

```
/**
```

```
 * Servlet Filter implementation class SanitizeFilter
```

```
 */
```

```
public class SanitizeFilter implements Filter {
```

```
    /**
```

```
     * Default constructor.
```

```
    */
```

```
    public SanitizeFilter() {
```

```
        // TODO Auto-generated constructor stub
```

```
    }
```

```
    /**
```

```
     * @see Filter#destroy()
```

```
    */
```

```
    public void destroy() {
```

```
        // TODO Auto-generated method stub
```

```
    }
```

```
    /**
```

```
     * @see Filter#doFilter(ServletRequest, ServletResponse, FilterChain)
```

```
    */
```

```
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
```

```
        throws IOException, ServletException {
```

```
        // TODO Auto-generated method stub
```

```
        // place your code here
```

```
        // pass the request along the filter chain
```

```
        try {
```

```
            System.out.println("here in filter");
```

```

String email = request.getParameter("email");
if (email != null) {
    email = email.replaceAll("[^\\dA-Za-z0-9@.-_]", "");
    System.out.println(email);
    request.setAttribute("email", email);
}

String password = request.getParameter("password");
System.out.println(password);
if (password != null) {
    password = password.replaceAll("[^\\dA-Za-z0-9@.-_]", "");
    System.out.println(password);
    request.setAttribute("password", password);
}

} catch (Exception e) {
    System.out.println(e);
    request.setAttribute("errorMessage", e);
    // request.getRequestDispatcher("error.jsp").forward(request,
    // response);
}
chain.doFilter(request, response);
}

/**
 * @see Filter#init(FilterConfig)
 */
public void init(FilterConfig fConfig) throws ServletException {
    // TODO Auto-generated method stub
}

}

```

Validator:

1. UserValidator

```

package com.music.musicstore.validator;

import org.springframework.validation.Errors;
import org.springframework.validation.ValidationUtils;
import org.springframework.validation.Validator;

import com.music.musicstore.pojo.User;

public class UserValidator implements Validator {
    @Override

```

```
    public boolean supports(Class<?> clazz) {
        // This Validator validates *just* User instances
        return User.class.equals(clazz);
    }

    @Override
    public void validate(Object target, Errors errors) {
        // basic validations
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "fname", "empty.fname", "First
Name is Required");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "lname", "empty.lname", "Last
Name is Required");
        ValidationUtils.rejectIfEmpty(errors, "email", "empty.email", "Email is Required");
        ValidationUtils.rejectIfEmpty(errors, "password", "empty.password", "Password is
Required");

        // add additional custom validations below
    }
}
```

Pojo:

1. Album

```
package com.music.musicstore.pojo;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.Table;

import com.fasterxml.jackson.annotation.JsonManagedReference;

@Entity
@Table(name = "Album")

public class Album {
```



```
public enum Status{
    ACCEPTED("ACCEPTED") ,REJECTED("REJECTED"), PENDING("PENDING");
    private String role;

    public String getRole() {
        return role;
    }
    Status(String role){
        this.role=role;
    }
}

@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private int album_id;

@ManyToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
@JoinColumn(name="artist_id")
private Artist artist;

@Column
private String status;

@Column
private String albumName;

@OneToMany(fetch =FetchType.EAGER, cascade = CascadeType.ALL , mappedBy =
"track_Id")
@JsonManagedReference
private Set<Track> tracks;

public Album() {
    this.status =Status.PENDING.getRole();
    this.tracks = new HashSet<Track>();
}

public int getAlbum_id() {
    return album_id;
}

public void setAlbum_id(int album_id) {
    this.album_id = album_id;
}

public User getArtist() {
    return artist;
}
```

```
    public void setArtist(Artist artist) {
        this.artist = artist;
    }

    public String getAlbumName() {
        return albumName;
    }

    public void setAlbumName(String albumName) {
        this.albumName = albumName;
    }

    public Set<Track> getTracks() {
        return tracks;
    }

    public void setTracks(Set<Track> tracks) {
        this.tracks = tracks;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

}
```

2. Artist

```
package com.music.musicstore.pojo;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.AttributeOverride;
import javax.persistence.AttributeOverrides;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="Artist")
@AttributeOverrides({
    @AttributeOverride(name="fname", column=@Column(name="FIRSTNAME")),
    @AttributeOverride(name="lname", column=@Column(name="LASTNAME"))
})
public class Artist extends User {

    @OneToMany(mappedBy="artist",fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Set<Album> albums;

    public Artist(){
        this.albums = new HashSet<Album>();
    }

    public Artist(User user) {
        this.setId(user.getId());
        this.setFname(user.getFname());
        this.setLname(user.getLname());
        this.setEmail(user.getEmail());
        this.setPassword(user.getPassword());
        this.setStatus(0);
        this.setRole(UserRole.ARTIST.getRole());
        this.albums = new HashSet<Album>();
    }

    public Set<Album> getAlbums() {
        return albums;
    }

    public void setAlbums(Set<Album> albums) {
        this.albums = albums;
    }
}
```

```
}
```

3. PlayList

```
package com.music.musicstore.pojo;
```

```
import java.util.Set;
```

```
import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.ManyToOne;
```

```
@Entity
```

```
public class PlayList {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    private int playlist_Id;
```

```
    @ManyToMany(cascade = CascadeType.REFRESH , fetch=FetchType.EAGER)
```

```
    @JoinTable(name = "playlist_track_table",joinColumns = {
```

```
        @JoinColumn(name = "playlist_Id", nullable = false, updatable = false) },
```

```
    inverseJoinColumns = {
```

```
        @JoinColumn(name = "track_Id") })
```

```
    private Set<Track> tracks;
```

```
    @ManyToOne(fetch = FetchType.LAZY, cascade = CascadeType.ALL)
```

```
    @JoinColumn(name = "subscriber_id" )
```

```
    private Subscriber subscriber;
```

```
    private String playListName;
```

```
    public Subscriber getSubscriber() {
```

```
        return subscriber;
```

```
    }
```

```
    public void setSubscriber(Subscriber subscriber) {
```

```
        this.subscriber = subscriber;
```

```
    }

    public String getPlayListName() {
        return playListName;
    }

    public void setPlayListName(String playListName) {
        this.playListName = playListName;
    }

    public int getPlaylist_Id() {
        return playlist_Id;
    }

    public void setPlaylist_Id(int playlist_id) {
        this.playlist_Id = playlist_id;
    }

    public Set<Track> getTracks() {
        return tracks;
    }

    public void setTracks(Set<Track> tracks) {
        this.tracks = tracks;
    }
}
```

4. Subscriber

```
package com.music.musicstore.pojo;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.OneToOne;

@Entity
public class Subscriber extends User {

    @OneToOne(fetch=FetchType.EAGER,cascade= CascadeType.ALL
,mappedBy="subscriber")
    private Set<PlayList> playList;

    public Subscriber(){}
```

```
    public Subscriber(User user) {
        this.setId(user.getId());
        this.setFname(user.getFname());
        this.setLname(user.getLname());
        this.setEmail(user.getEmail());
        this.setPassword(user.getPassword());
        this.setStatus(0);
        this.setRole(UserRole.SUBSCRIBER.getRole());
        this.playList = new HashSet<PlayList>();
    }

    public Set<PlayList> getPlayList() {
        return playList;
    }

    public void setPlayList(Set<PlayList> playList) {
        this.playList = playList;
    }
}
```

5. Track

package com.music.musicstore.pojo;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Transient;

import org.springframework.web.multipart.MultipartFile;

import com.fasterxml.jackson.annotation.JsonBackReference;

@Entity

public class Track {

@Id

@GeneratedValue(strategy = GenerationType.*AUTO*)

private int track_Id;

@ManyToOne(fetch = FetchType.*EAGER*, cascade = CascadeType.*ALL*)

@JoinColumn(name = "album_id")

```
@JsonBackReference
private Album album;
```

```
@Column
private String title;
```

```
@Column
private String fileName;
```

```
@Transient
private String albumId;
```

```
@Transient
MultipartFile file;
```

```
@Transient
private MultipartFile music;
```

```
public Track() {

}
```

```
public int getTrack_Id() {
    return track_Id;
}
```

```
public void setTrack_Id(int track_Id) {
    this.track_Id = track_Id;
}
```

```
public Album getAlbum() {
    return album;
}
```

```
public void setAlbum(Album album) {
    this.album = album;
}
```

```
public String getTitle() {
    return title;
}
```

```
public void setTitle(String title) {  
    this.title = title;  
}
```

```
public String getFileName() {  
    return fileName;  
}
```

```
public void setFileName(String fileName) {  
    this.fileName = fileName;  
}
```

```
public String getAlbumId() {  
    return albumId;  
}
```

```
public void setAlbumId(String albumId) {  
    this.albumId = albumId;  
}
```

```
public MultipartFile getFile() {  
    return file;  
}
```

```
public void setFile(MultipartFile file) {  
    this.file = file;  
}
```

```
public MultipartFile getMusic() {  
    return music;  
}
```

```
public void setMusic(MultipartFile music) {  
    this.music = music;  
}
```

```
}
```


6. User

```
package com.music.musicstore.pojo;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Inheritance;
import javax.persistence.InheritanceType;

@Entity
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
public class User {

    public enum UserRole{
        ADMIN("ADMIN") ,ARTIST("ARTIST"), SUBSCRIBER("SUBSCRIBER");
        private String role;

        public String getRole() {
            return role;
        }
        UserRole(String role){
            this.role=role;
        }
    }

    @Id
    @GeneratedValue
    private int id;

    @Column(name="firstName")
    private String fname;

    @Column(name="lastName")
    private String lname;

    @Column(name="email_id")
    private String email;

    @Column(name="password")
    private String password;

    @Column(name="role")
    private String role;

    @Column(name="status")
    private int status;
```

```
//default Constructor
public User() {

}
//Parameterised constructor
public User(String fname, String lname, String email, String password, String role) {
    super();
    this.fname = fname;
    this.lname = lname;
    this.email = email;
    this.password = password;
    this.role = role;
    this.status = 0;
}
public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public String getFname() {
    return fname;
}
public void setFname(String fname) {
    this.fname = fname;
}
public String getLname() {
    return lname;
}
public void setLname(String lname) {
    this.lname = lname;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
public String getRole() {
    return role;
}
public void setRole(String role) {
```

```
        this.role = role;
    }
    public int getStatus() {
        return status;
    }
    public void setStatus(int status) {
        this.status = status;
    }
}
```

JSP:**1. add.jsp**

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<% @taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag -->

<link
    href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
    rel="stylesheet">
</head>
<body>
<jsp:include page="dashboard-sidenav.jsp"></jsp:include>
<div style="margin-left: 30%; margin-top: 5%">
    <a href="/addNew" class="btn btn-primary btn-lg active"
        role="button" aria-pressed="true">Add New Album</a> &nbsp;&nbsp;&nbsp;<a
        style="margin-left: 10%;" href="/addTrack"
        class="btn btn-primary btn-lg active" role="button"
        aria-pressed="true">Add New Track</a>
    <c:if test="${requestScope.addType == 'ALBUM'}">
        <div style="margin-top: 5%">
            <form action="/add" method="post">
                <label>ALBUM name </label> <input type="text" class="form-
control"
                    style="width: 41%" name="albumName" placeholder="">
                <input
                    type="hidden" name="addType" value="ALBUM" /> <br
/>
                <br /> <input type="submit" class="form-control btn-primary"
                    style="width: 20%; margin-left: 10%" name="submit"
placeholder="">
            </form>
        </div>
    </c:if>

```

```

    </c:if>

    <c:if test="${requestScope.addType == 'TRACK'}">

        <form:form commandName="track" enctype="multipart/form-data"
            class="form-horizontal" method="post">
            <br/><br/><br/><br/>
            <c:if test="${requestScope.albums.size()<=0 || requestScope.albums == null }">
                <h1> NO ALBUMS TO ADD TRACKS!!!! </h1>
            </c:if>
            <c:if test="${requestScope.albums.size()>0 && requestScope.albums !=
null }">

                <h4>Select Album</h4>
                <form:select id="albumId" name="album" path="albumId"
style="width:40%">
                    <c:forEach items="${requestScope.albums}" var="album">
                        <option
value="${album.album_id}">${album.albumName}</option>
                    </c:forEach>
                </form:select><br/><br/>

                <h4> Title</h4>
                <form:input type="text" class="form-control"
                    style="width:40%" name="trackName" path="title"
placeholder=""></form:input><br/><br/>

                <h4> Audio File </h4>
                <form:input type="file" id="track" name="track" required="required"
path="music" accept=".mp3,audio/*"></form:input>

                <br/>
                <input type="hidden" name="addType" value="TRACK" />

            <br />
            <input type="submit" class="form-control btn-primary" style="width: 30%;
margin-left: 5%" name="submit" >
        </c:if>
    </form:form>
    </c:if>

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

```

2. addtracktoplaylist.jsp

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag ----->

<link
    href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
    rel="stylesheet">
</head>
<body>
    <jsp:include page="dashboard-sidenav.jsp"></jsp:include>
    <div style="margin: 5% 20%;">
        <c:if test="${requestScope.playlists != null}">
            <form action="/addToPlaylist" method="GET">
                <label>Track:</label><input type="text" name="track" value="${track.title}"
disabled="disabled"/><br/><br/>
                <input type="hidden" name="trackid" value="${track.track_Id}"/>
                <br/><br/><label>PLAYLISTS</label>
                <select name="playlist">

                    <c:forEach var="p" items="${requestScope.playlists}">
                        <option value="${p.playlist_Id}">${p.playlistName}</option>
                    </c:forEach>
                </select><br/><br/>
                <input type="submit" name="submit"
                    class="btn btn-primary btn-block"
value="Submit">
            </form>
        </c:if>
        <c:if test="${requestScope.playlists == null}">
            <h1>No playlists available to add to. Please create one!!</h1>
        </c:if>
    </div>
</body>
</html>

```

3. admin-view

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<link rel="stylesheet" href="/resources/css/adminView.css">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag -->

<link
    href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
    rel="stylesheet">

</head>
<body>
<jsp:include page="dashboard-sidenav.jsp"></jsp:include>

<div style="margin: 2% 20% 40% 40%;">
    <h1>Welcome to Music Store</h1>
</div>

<table>
<thead><th></th></thead>
<tr></tr>
</table>
</body>
</html>
```

4. album

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
```

```

<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#"><h3>Music Library</h3></a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><h3><a href="#">Home</a></h3></li>&nbsp;
      <li><h3><a href="/searchtrack">Search Track</a></h3></li>&nbsp;
      <li><h3><a href="/searchartist">Search Artist</a></h3></li>&nbsp;
      <li><h3><a href="/playlist">Playlist</a></h3></li>&nbsp;
      <li><h3><a href="/album">Album</a></h3></li>&nbsp;
      <li><h3><a href="/accountsettings">Account Settings</a></h3></li>&nbsp;
    </ul>
    <form class="navbar-form navbar-left" action="/action_page.php">
      <div class="form-group">
        <input type="text" class="form-control" placeholder="Search">
      </div>
      <button type="submit" class="btn btn-default">Submit</button>
    </form>
  </div>
</nav>
</body>
</html>

```

5. artist-view

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag ----->

<link
    href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
    rel="stylesheet">
</head>

```



```
<body>

<jsp:include page="dashboard-sidenav.jsp"></jsp:include>

<div style="margin: 2% 20% 40% 40%;">
  <h1>Welcome to Music Store</h1>
</div>

</body>
</html>
```

6. dashboard-sidenav

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/malihu-custom-scrollbar-
plugin/3.1.5/jquery.mCustomScrollbar.min.css">
<style>
#sidebar {
    width: 220px;
    position: fixed;
    top: 0;
    left: 0;
    height: 100vh;
    z-index: 999;
    background: #007BBD;
    color: #fff;
}

h1{
    margin: 10%;
    font-family: 'Brush Script MT',cursive;
}

.active{
    height: 60px;
    background-color: #007BBB !important;
    border: none;
}
</style>
</head>
```

```
<body>
  <nav id="sidebar">
    <div class="sidebar-header">
      <h1 class="text-center">Music Store</h1>
    </div>
    <br/><br/>
    <div class="list-group text-center">
      <b>
        <c:if test="\${sessionScope.user == null}">
          <a href="/" class="list-group-item list-group-item-action active">Login
            &#10148;</a>
          <a href="/register" class="list-group-item list-group-item-action
            active">Register &#10148;</a>
        </c:if>
        <c:if test="\${sessionScope.user.role == 'ADMIN'}">
          <a href="/homepage" class="list-group-item list-group-item-action
            active">Home</a>
          <a href="/viewartists" class="list-group-item list-group-item-action
            active">View Artists</a>
          <a href="/viewsubscribers" class="list-group-item list-group-item-action
            active">View Subscribers</a>
          <a href="/viewalbumrequests" class="list-group-item list-group-item-
            action active">Requests</a>
          <a href="/report" class="list-group-item list-group-item-action
            active">View Report</a>
          <a href="/logout" class="list-group-item list-group-item-action
            active">Logout</a>
        </c:if>
        <c:if test="\${sessionScope.user.role == 'ARTIST'}">
          <a href="/homepage" class="list-group-item list-group-item-action
            active">Home</a>
          <a href="/viewalbums" class="list-group-item list-group-item-action
            active">Albums</a>
          <a href="/add" class="list-group-item list-group-item-action
            active">New Track/Album</a>
          <a href="/requeststatus" class="list-group-item list-group-item-action
            active">My Requests</a>
          <a href="/logout" class="list-group-item list-group-item-action
            active">Logout</a>
        </c:if>
        <c:if test="\${sessionScope.user.role == 'SUBSCRIBER'}">
          <a href="/homepage" class="list-group-item list-group-item-action
            active">Home</a>
          <a href="/search" class="list-group-item list-group-item-action
            active">Search</a>
          <a href="/playlist" class="list-group-item list-group-item-action
            active">My Playlists</a>
          <a href="/logout" class="list-group-item list-group-item-action
            active">Logout</a>
```

```

        </c:if>
    </b>
</div>
</nav>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
        integrity="sha384-
q8i/X+965DzO00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
        crossorigin="anonymous"></script>

<script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"
        integrity="sha384-
cs/chFZiN24E4KMATLdqdvsézGxaGsi4hLGOzIXwp5UZB1LY//20VyM2taTB4QvJ"
        crossorigin="anonymous"></script>

<script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"
        integrity="sha384-
uefMccjFJAIv6A+rW+L4AHf99KvxDjWsu1z9VI8SKNVmz4sk7buKt/6v9KI65qnm"
        crossorigin="anonymous"></script>

<script
        src="https://cdnjs.cloudflare.com/ajax/libs/malihu-custom-scrollbar-
plugin/3.1.5/jquery.mCustomScrollbar.concat.min.js"></script>

</body>
</html>

```

7. error

```

<% @taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<% @taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<% @taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html><html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<h1>Error Page</h1>
<p>${errorMessage}</p>

<c:if test = "${resendLink== true}" >

    <c:set var="contextPath" value="${pageContext.request.contextPath}" />
    <form action="${contextPath}/user/resendemail.htm" method="POST">

```

```

        User Email:<input type="text" name="username" size="30" required="required" />
        <br/>
        <input type="submit" value="Resend Email" />
    </form>
</c:if>

</body>
</html>

```

8. home

```

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<% @ page session="false"%>
<html>
<head>
<link
    href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script
    src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<!-- Include the above in your HEAD tag ----->
</head>
<body>
<jsp:include page="dashboard-sidenav.jsp"></jsp:include>
    <div id="login">
        <h3 class="text-center text-white pt-5">Login form</h3>
        <div class="container">
            <div id="login-row"
                class="row justify-content-center align-items-center">
                <div id="login-column" class="col-md-6">
                    <div id="login-box" class="col-md-12">
                        <c:if test="${requestScope.registration_success != null}">
                            <div class="alert alert-
success">${requestScope.registration_success}</div>
                        </c:if>
                        <form id="login-form" class="form" action="home"
method="post">
                            <h3 class="text-center text-info">Login</h3>
                            <div class="form-group">
                                <c:if test="${requestScope.login_error!=
null}">
                                    <div class="alert alert-
danger">${requestScope.login_error}</div>
                                </c:if>
                                <label for="username" class="text-
info">EmailID:</label><br>

```

```

id="username"
    <input type="email" name="email"
        class="form-control" required>
    </div>
    <div class="form-group">
        <label for="password" class="text-
info">Password:</label><br>
id="password"
        <input type="password" name="password"
            class="form-control" required>
        </div>
        <br />
    <div class="form-group">
        <input type="submit" name="submit"
            class="btn btn-primary btn-block"
value="Submit">
    </div>
</form>
<div class="border-top card-body text-center">
    Not a User? <a href="/register">Register</a>
</div>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

9. playlist

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">

```

```

<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
</head>
<body>

    <jsp:include page="dashboard-sidenav.jsp"></jsp:include>

    <div style="margin: 2% 25%">

        <a href="/addNewPlaylist" class="btn btn-primary btn-lg active" role="button">Add
New Playlist</a>

        <c:if test="${requestScope.addType == 'PLAYLIST'}">
            <div style="margin-top: 5%">

                <form action="/addNewPlaylist" method="post">
                    <c:if test="${requestScope.errorMessage != null}">
                        <label class="alert-danger">${errorMessage}</label>
                    </c:if>
                    <br/>
                    <label>ALBUM name </label> <input type="text" class="form-
control"
                        style="width: 41%" name="playlistName"
                        placeholder=""> <input
                        type="hidden" name="addType" value="ALBUM" /> <br
/> <br /> <input
                        type="submit" class="form-control btn-primary"
                        style="width: 20%; margin-left: 10%" name="submit"
                        placeholder="">
                    </form>
                </div>
            </c:if>

            <c:if test="${requestScope.addType == 'VIEW_PLAYLIST'}">
                <ul class="list-group list">
                    <c:set var="list" value="${requestScope.playlists}" />

                    <c:if test="${list.size()>0 && list != null}">
                        <br/><br/>
                        <h3>PLAYLISTS</h3>
                        <c:forEach var="a" items="${list}">
                            <li class="list-group-item d-flex justify-content-between
align-items-center">
                                <label style="margin-
left: 5%">${a.playlistName}</label> &nbsp;
                                <span class="badge progress-bar-primary"
style="background-color: #428BCA;">

```

10. register

```
<% @page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<% @taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<link rel="stylesheet" href="/resources/css/register.css">
<link
    href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<script
    src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<!-- Include the above in your HEAD tag -->
```

```
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.0.8/css/all.css">
<title>Insert title here</title>
</head>
<body>

<jsp:include page="dashboard-sidenav.jsp"></jsp:include>

<div class="container">
  <hr>
  <div class="row justify-content-center">
    <div class="col-md-6">
      <div class="card">
        <header class="card-header"> <a href="/login.htm"
class="float-right btn btn-outline-primary mt-1">Log
in</a>

        <h4 class="card-title mt-2">Sign up</h4>
        </header>
        <article class="card-body"> <form:form
commandName="user">

          <div class="form-row">
            <div class="col form-group">
              <label>First name </label>
              <form:input type="text" class="form-
control" path="fname"
placeholder=""></form:input>
              <form:errors path="fname" />
              <br />
            </div>

            <!-- form-group end.-->
            <div class="col form-group">
              <label>Last name</label>
              <form:input type="text" class="form-
control" path="lname"
placeholder=""></form:input>
              <form:errors path="lname" />
              <br />
            </div>
            <!-- form-group end.-->
          </div>

          <!-- form-row end.-->
          <div class="form-group">
            <label>Email address</label>
            <form:input type="email" class="form-control"
path="email"
placeholder=""></form:input>
            <form:errors path="email" />
          </div>
        </article>
      </div>
    </div>
  </div>
</div>
```


danger">\${requestScope.error}</div>

never

<form:radiobutton

input" value="SUBSCRIBER"

checked="checked"></form:radiobutton> <span

inline"> <form:radiobutton

input" value="ARTIST"></form:radiobutton>

Artist

type="password" path="password"

btn-block">

Terms of use

<c:if test="\${requestScope.error != null}">

<div class="alert alert-

</c:if>

 <small class="form-text text-muted">We'll

share your email with anyone else.</small>

</div>

<!-- form-group end.// -->

<div class="form-group">

<label>Register As:</label>
 <label

class="form-check form-check-inline">

path="role" class="form-check-

class="form-check-label"> Subscriber

</label> <label class="form-check form-check-

path="role" class="form-check-

**

</label>

</div>

<!-- form-row.// -->

<div class="form-group">

<label>Create password</label>

<form:input class="form-control"

placeholder=""></form:input>

<form:errors path="password" />

</div>

<!-- form-group end.// -->

<div class="form-group">

<button type="submit" class="btn btn-primary

Register</button>

</div>

<!-- form-group.// -->

<small class="text-muted">By clicking the 'Sign Up' button, you confirm that you accept our

and Privacy Policy.

</small>

</form:form> </article>

<!-- card-body end.// -->

```

        <div class="border-top card-body text-center">
            Have an account? <a href="/login">Log In</a>
        </div>
    </div>
    <!-- card.// -->
</div>
<!-- col.// -->

</div>
<!-- row.// -->

</div>
<br>
<br>
</article>
</body>
</html>

```

11. requeststatus

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <% @taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag -->

<link href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
rel="stylesheet">
</head>
<body>

    <jsp:include page="dashboard-sidenav.jsp"></jsp:include>

    <div style="margin-left: 30%; margin-top: 5%">

        <h1 style="margin-left: 20%">Request Status</h1>
        <ul class="list-group list">

```

```

<c:set var="list" value="\${requestScope.albumsList}" />

<c:if test="\${list.size()}>0 && list != null}">
    <c:forEach var="a" items="\${list}">
        <li class="list-group-item d-flex justify-content-between align-items-center">
            <label>\${a.albumName}</label> &nbsp; <label
style="margin-left:30%;">\${a.status}</label>
        </li>
    <br />
    </c:forEach>
</c:if>
<c:if test="\${list.size()}<=0 || list == null}">
    <h1 style="margin-left:15%">No Requests present !!!!</h1>
</c:if>
</ul>
</div>

</body>
</html>

```

12. search

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
/>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<style>

    #result {
    position: absolute;
    width: 100%;
    max-width:870px;
    cursor: pointer;
    overflow-y: auto;
    max-height: 400px;
    box-sizing: border-box;
    z-index: 1001;
    }
    .link-class:hover{

```

```

    background-color:#1f1f1f;
}
</style>

<script type="text/javascript">
function ajaxFunction(param)
{
    var xmlHttp;

    try // Firefox, Opera 8.0+, Safari
    {
        xmlHttp=new XMLHttpRequest();
    }
    catch (e)
    {
        try // Internet Explorer
        {
            xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (e)
        {
            try
            {
                xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e)
            {
                alert("Your browser does not support AJAX!");
                return false;
            }
        }
    }

    xmlHttp.onreadystatechange=function()
    {
        if(xmlHttp.readyState==4)
        {
            var result = JSON.parse(xmlHttp.responseText);

            result.forEach(function(track){
                console.log(track.track_id);
                var html = "<li class='list-group-item d-flex justify-content-between
align-items-center'>";

                html += "<label>" + track.title + "</label>";
                html+="

```

```

        html += "</li>"
    };

    document.getElementById("result").innerHTML += html;
});

    console.log(result);
}

}

document.getElementById("result").innerHTML = "";
var url = "http://localhost:8080/musicstore/searchTrack?search="+param;
xmlHttp.open("GET",url,true);
xmlHttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlHttp.send();
}

function searchTracks(){
    var str = document.getElementById("search").value;
    ajaxFunction(str);
}
</script>

</head>
<body>
<jsp:include page="dashboard-sidenav.jsp"></jsp:include>
<br /><br />
<div class="container" style="width:900px;">
    <h2 align="center">Search Tracks</h2>
    <br /><br />
    <div align="center" style="margin-top:2%;">
        <input type="text" name="search" id="search" placeholder="Search Tracks" class="form-control"
onchange="searchTracks()"/>
    </div><br/><br/>
    <div style="margin-top:2%;">
    <ul class="list-group" id="result"></ul>
    </div>
    <br />
    </div>
</body>
</html>

```

13. subscriber-view

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
</head>
<body>

<jsp:include page="dashboard-sidenav.jsp"></jsp:include>

<div style="margin: 2% 20% 40% 40%;">
    <h1>Welcome to Music Store</h1>
</div>

</body>
</html>
```

14. viewalbumrequest

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag -->

<link
    href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
    rel="stylesheet">
</head>
<body>
    <jsp:include page="dashboard-sidenav.jsp"></jsp:include>

    <div style="margin-left: 30%; margin-top: 5%">
```

```

<c:set var="list" value="${requestScope.albumlist}" />

<c:if test="${list.size()}>0 && list != null}">
    <h1 style="margin-left: 20%">Request Status</h1>

    <ul class="list-group list">
        <c:forEach var="a" items="${list}">
            <li class="list-group-item d-flex justify-content-between align-items-center">

                <label>${a.albumName}</label>
                <span class="badge progress-bar-success">
                    <a
href="/requestalbum?status=ACCEPTED&albumid=${a.album_id}"
class="btn-success" role="button" aria-
pressed="true">Accept</a>

                </span>
                <span class="badge progress-bar-danger">
                    <a
href="/requestalbum?status=REJECTED&albumid=${a.album_id}"
class="btn-danger" role="button" aria-
pressed="true">Reject</a>

            </span>
        </li>
        <br />
        </c:forEach>
    </ul>

</c:if>
<c:if test="${list.size()}<=0 || list == null}">
    <h1 style="margin-left: 20%">No Requests Found!!</h1>
</c:if>

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

```

15. viewalbums

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script

```

```

        src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag ----->

<link
    href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
    rel="stylesheet">
</head>
<style type="text/css">
.list {
    margin: 5% 20% 0% 30%;
}
</style>
<body>

    <jsp:include page="dashboard-sidenav.jsp"></jsp:include>

    <ul class="list-group list">
        <c:set var="list" value="${requestScope.albumsList}" />
        <h1 style="margin-left: 20%">My Albums</h1>
        <c:if test="${list.size()>0 && list != null}">
            <label>ALBUM NAME</label> &nbsp; <label style="margin-left:25%">
STATUS</label><label style="margin-left:40%">ACTION</label>
            <c:forEach var="a" items="${list}">
                <li class="list-group-item d-flex justify-content-between align-items-center">
                    <label>${a.albumName}</label> &nbsp; <label
style="margin-left:30%"> ${a.status}</label>
                    <c:if test="${a.status=='ACCEPTED'}">
                        <span class="badge progress-bar-primary"
style="background-color: #428BCA;">
                            <a
href="/viewalbumtracks?albumid=${a.album_id}" class="btn-primary " role="button"
aria-pressed="true">VIEW
TRACKS</a>
                        </span>
                    </c:if>
                    <c:if test="${a.status!='ACCEPTED'}">
                        <span class="badge progress-bar-secondary">
                            <a href="#" class="btn-secondary"
role="button"
aria-pressed="true"
disabled>VIEW TRACKS</a>
                        </span>
                    </c:if>
                </li>
            </c:forEach>
        </c:if>
    </ul>

```



```

        </c:forEach>
    </c:if>
    <c:if test="${list.size()<=0 || list == null}">
        No Albums present !!!!
    </c:if>
</ul>
</div>
</body>
</html>

```

16. viewalbumtracks

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<!-- Include the above in your HEAD tag -->

<link
    href="//maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css"
    rel="stylesheet">
</head>
<style type="text/css">
.list {
    margin: 5% 20% 0% 30%;
}
</style>
<body>

<jsp:include page="dashboard-sidenav.jsp"></jsp:include>

<ul class="list-group list">
    <c:set var="list" value="${requestScope.albumsList}" />
    <h1 style="margin-left: 20%">My Albums</h1>
    <c:if test="${list.size()>0 && list != null}">
        <label>ALBUM NAME</label> &nbsp;  <label style="margin-left:25%">
STATUS</label><label style="margin-left:40%">ACTION</label>

```

```

        <c:forEach var="a" items="${list}">
            <li class="list-group-item d-flex justify-content-between align-items-center">
                <label>${a.albumName}</label> &nbsp; <label
style="margin-left:30%"> ${a.status}</label>
                <c:if test="${a.status=='ACCEPTED'}">
                    <span class="badge progress-bar-primary"
style="background-color: #428BCA;">
                        <a
href="/viewalbumtracks?albumid=${a.album_id}" class="btn-primary " role="button"
aria-pressed="true">VIEW
TRACKS</a>
                    </span>
                </c:if>
                <c:if test="${a.status!='ACCEPTED'}">
                    <span class="badge progress-bar-secondary">
                        <a href="#" class="btn-secondary"
role="button"
aria-pressed="true"
disabled>VIEW TRACKS</a>
                    </span>
                </c:if>
            </li>
        </c:forEach>
    </c:if>
    <br />
    </div>
</body>
</html>

```

17. viewsubscriber

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<% @ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>View Artist</title>

```

```

<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<style type="text/css">
.list{
    margin: 5% 20% 0% 30%;
}
</style>
</head>
<body>

    <jsp:include page="dashboard-sidenav.jsp"></jsp:include>
    <ul class="list-group">
        <c:set var="list" value="${requestScope.subscriberList }" />
        <c:if test="${list.size()>0 && list != null}">
            <c:forEach var="a" items="${list}">
                <li
                    class="list-group-item d-flex justify-content-between align-items-
center">
                    <label>${a.fname}</label> &nbsp; <label>${a.lname}</label>
                    <span class="badge badge-primary badge-pill">
                        <a href="/deletesubscriber?user=${a.id}" class="btn btn-
primary btn active"
                            role="button" aria-pressed="true">Delete Subscriber</a>
                    </span>
                </li>
            </c:forEach>
        </c:if>
        <c:if test="${list.size()<=0 || list == null}">
            No artist present!!!!
        </c:if>
    </ul>
</body>
</html>

```

18. viewtracks

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link

```

```

    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
</head>
<body>

    <jsp:include page="dashboard-sidenav.jsp"></jsp:include>
    <div style="margin:5% 30%;">
        <h2>PLAYLIST-Tracks</h2>
        <ul class="list-group list" style="margin-top: 5%">
            <c:set var="list" value="${requestScope.tracks}" />

            <c:if test="${list.size()>0 && list != null}">
                <br/><br/>
                <b>
                    <li class="list-group-item d-flex justify-content-between align-
items-center">
                        <label>Track Name</label><label style="margin-left:
20%">Album Name</label> &nbsp;
                        <label style="margin-left: 26%">Artist Name</label>
                    </li>
                </b>

                <c:forEach var="a" items="${list}">
                    <li class="list-group-item d-flex justify-content-between
align-items-center">
                        <label>${a.title}</label><label style="margin-left:
30%">${a.album.albumName}</label> &nbsp;
                        <label style="margin-left:
30%">${a.album.artist.fname}</label>&nbsp;&nbsp;<label>${a.album.artist.lname}</label>
                    </li>
                <br />
                </c:forEach>
            </c:if>
            <c:if test="${list.size()<=0 || list == null}">
                <h1>No Tracks present !!!!</h1>
            </c:if>

        </ul>
    </div>

</body>
</html>

```

19. viewusers

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```

    pageEncoding="ISO-8859-1"%>
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>View Artist</title>
<link
    href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
    rel="stylesheet" id="bootstrap-css">
<script
    src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<style type="text/css">
.list {
    margin: -8% 20% 0% 30%;
}
</style>
</head>
<body>

<jsp:include page="dashboard-sidenav.jsp"></jsp:include>

<div>

    <c:set var="list" value="${requestScope.userList }" />
    <c:if test="${requestScope.display=='ARTIST'}">
        <h1 style="margin-left:40%">Artists</h1>
        <ul class="list-group list">
            <c:if test="${list.size()>0 && list != null}">
                <c:forEach var="a" items="${list}">
                    <li class="list-group-item d-flex justify-content-between
align-items-center">
                        <label>${a.fname}</label> &nbsp;
                        <span class="badge badge-primary"
style="background-color: #428BCA;">
                            <a href="/deleteartist?user=${a.id}" class="btn-
primary" role="button">Delete Artist</a>
                        </span>
                    </li>
                </c:forEach>
            </c:if>
        </ul>
        <c:if test="${list.size()<=0 || list == null}">
            <h1 style="margin-left:40%">No Artists Found!!</h1>
        </c:if>
    </c:if>

```

```
</c:if>

</div>

<div>
  <c:if test="${requestScope.display=='SUBSCRIBER'}">
    <h1 style="margin-left:40%">Subscribers</h1>
    <c:if test="${list.size()>0 && list != null}">
      <ul class="list-group list">
        <c:forEach var="a" items="${list}">
          <li class="list-group-item d-flex justify-content-between
align-items-center">
            <label>${a.fname}</label> &nbsp;
            <span class="badge badge-primary"
style="background-color: #428BCA;">
              <a href="/deletesubscriber?user=${a.id}"
class="btn-primary " role="button">Delete Subscriber</a>
            </span>
          </li>
        </c:forEach>
      </ul>
    </c:if>
    <c:if test="${list.size()<=0 || list == null}">
      <h1 style="margin-left:40%">No Subscribers Found!!</h1>
    </c:if>
  </c:if>
</div>

</body>
</html>
```