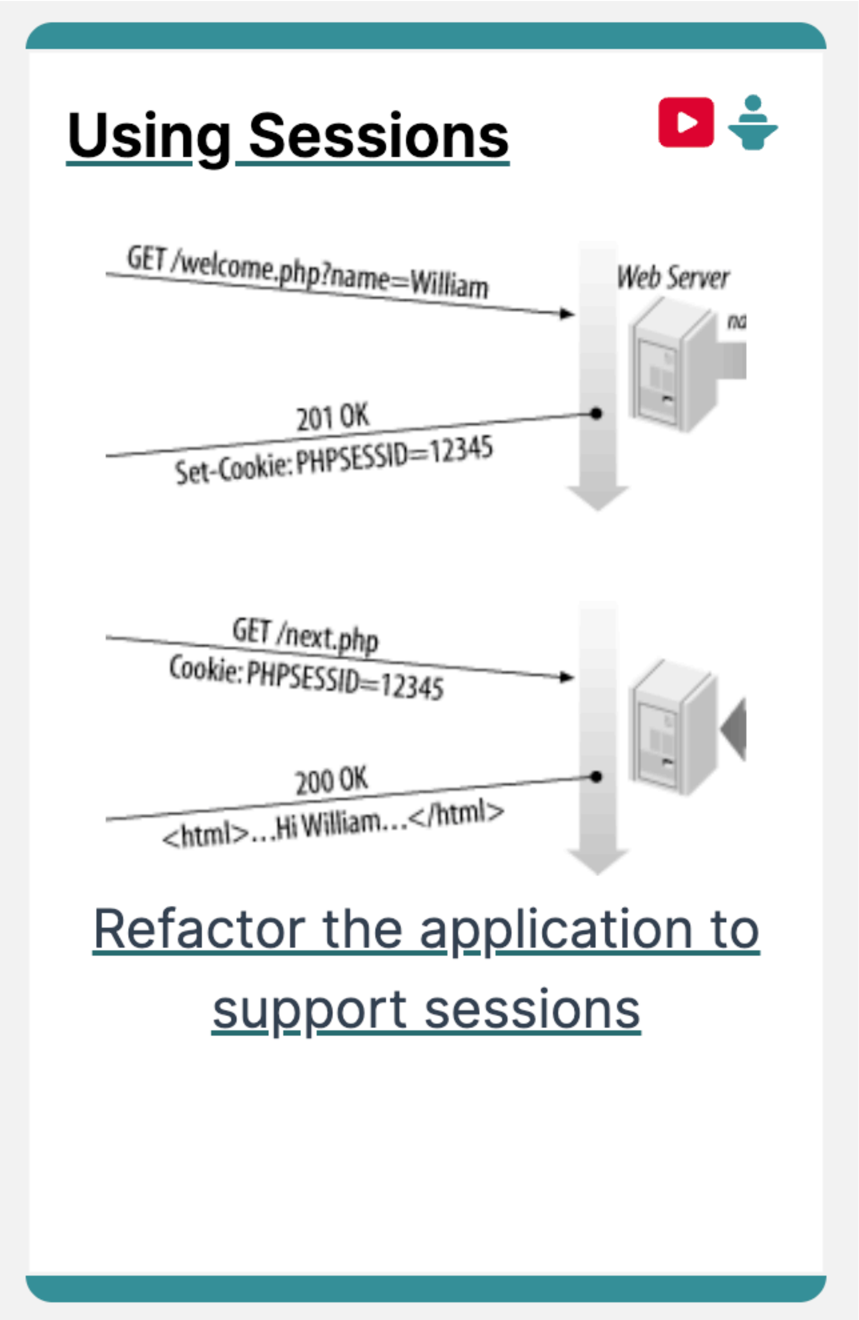
Using Sessions

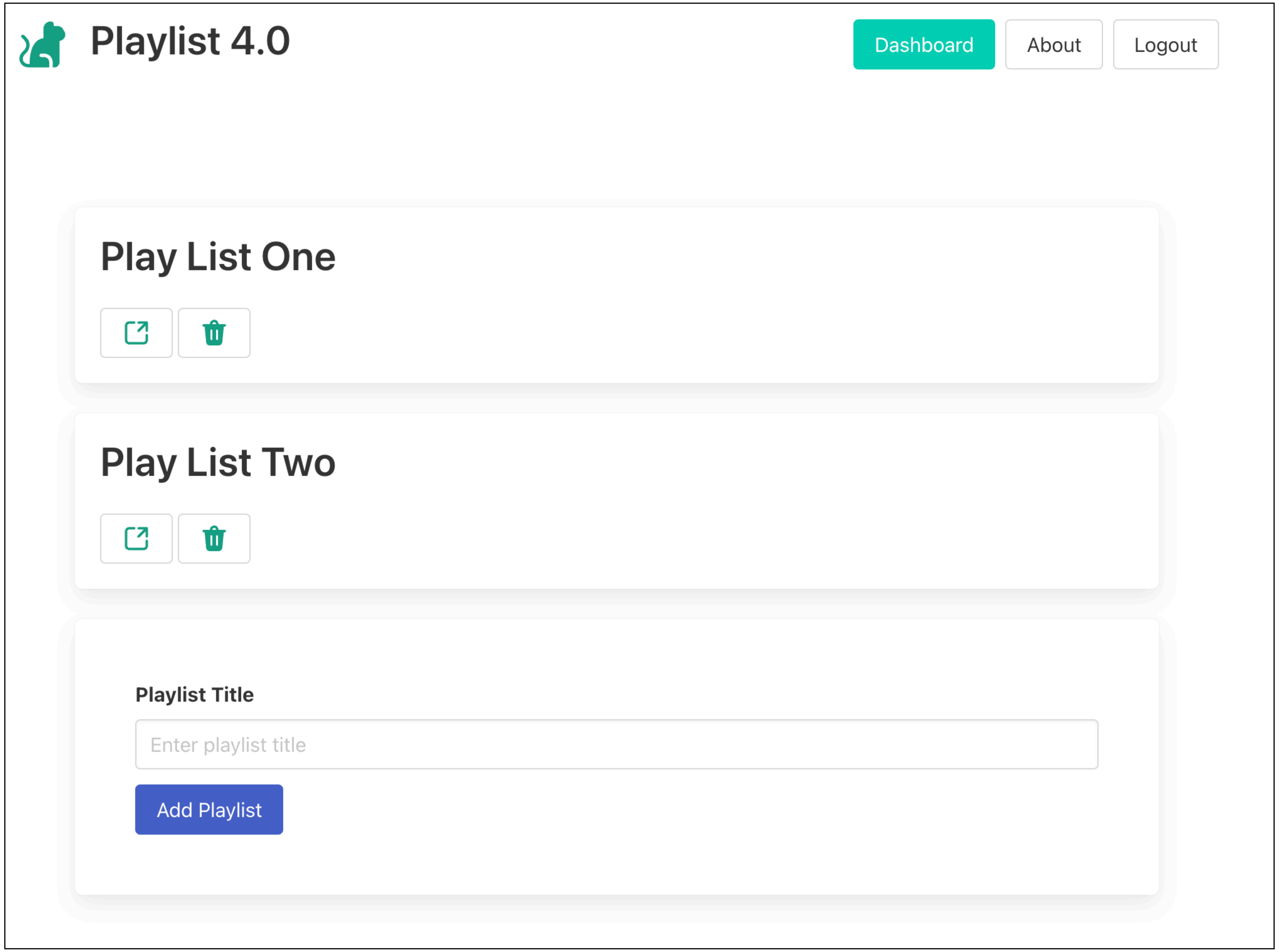


Using Sessions

GET /welcome.php?name=William
Web Server
201 OK
Set-Cookie: PHPSESSID=12345
GET /next.php
Cookie: PHPSESSID=12345
200 OK
<html>...Hi William...</html>

Refactor the application to support sessions

Playlist Model



Revised Playlist Model

```
{
  "file": "./models/playlists.json",
  "playlists": [
    {
      "title": "Play List One",
      "_id": "acf1d2a4-332a-4a45-afa8-cd2704894ef4"
    },
    {
      "title": "Play List Two",
      "_id": "f3b15c0d-33a3-47fd-a3f3-60a4f606fc7f"
    }
  ]
}
```

Global Playlist, no user support

Playlist V3 (No Session/User support)

Revised Playlist Model

additional property identifies user id

```
{
  "file": "./models/playlists.json",
  "playlists": [
    {
      "title": "Play List One",
      "userid": "538a527f-e42d-47db-b3fe-cdb2da9548c0",
      "_id": "acf1d2a4-332a-4a45-afa8-cd2704894ef4"
    },
    {
      "title": "Play List Two",
      "userid": "538a527f-e42d-47db-b3fe-cdb2da9548c0",
      "_id": "f3b15c0d-33a3-47fd-a3f3-60a4f606fc7f"
    }
  ]
}
```

Playlist belongs to specific user

Playlist-V4

playlistStore

New method to retrieve all playlists belonging to a specific user

Id of user passed as parameter

```
  async getPlaylistsByUserId(userid) {
    await db.read();
    return db.data.playlists.filter((playlist) => playlist.userid === userid);
  },
```

dashboard: index

```
export const dashboardController = {

  ...
  async index(request, response) {

    const loggedInUser = await accountsController.getLoggedInUser(request);

    const viewData = {
      title: "Playlist Dashboard",
      playlists: await playlistStore.getPlaylistsByUserId(loggedInUser._id),
    };

    console.log("dashboard rendering”);

    response.render("dashboard-view", viewData);
  },

  ...
}
```

Discover which user is currently logged in

Retrieve only those playlists associated with the logged in user

dashboard: addPlaylist

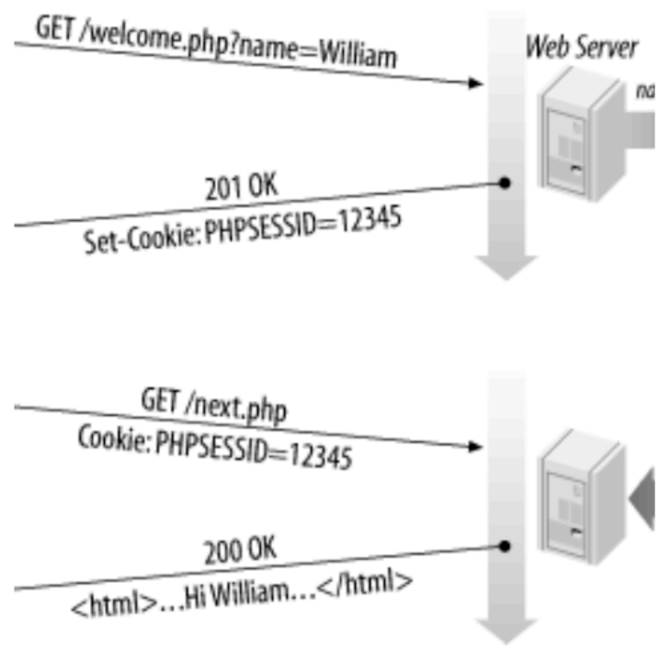
```
export const dashboardController = {

 ...
  async addPlaylist(request, response) {
    const loggedInUser = await accountsController.getLoggedInUser(request);

    const newPlayList = {
      title: request.body.title,
      userid: loggedInUser._id,
    };

    console.log(`adding playlist ${newPlayList.title}`);
    await playlistStore.addPlaylist(newPlayList);
    response.redirect("/dashboard");
  },

  ...
}
```

Discover which user is currently logged in

In the new playlist, include the id of the currently logged in user

Using Sessions

Using Sessions


GET /welcome.php?name=William
Web Server
201 OK
Set-Cookie: PHPSESSID=12345
GET /next.php
Cookie: PHPSESSID=12345
200 OK
<html>...Hi William...</html>

Refactor the application to support sessions