# LS

# Backend:

I use spring boot as the back end for the LS app,

|  |  |
| --- | --- |
| Spring boot | Backend. |
| Connected database | Oracle |
| Spring boot security | Authentication – authorization process using token.  --Angular also used in authorization process by \*ngIf and authGurad. |
| httpBasic | Generating the token. |
| BCryptPasswordEncoder | Encode the password. |

## API calls for method:

* User

|  |  |
| --- | --- |
| getAllUsers() | <http://localhost:8080/userAPI/getAllUsers> |
| getUser (int userId) | <http://localhost:8080/userAPI/getUser/1>053 |
| getUserByIdOrUsername(int id, String username) | <http://localhost:8080/userAPI/getUserByIdOrUsername/77/user_deem> |
| createUser(User user) | <http://localhost:8080/register>  Body:  {      "username": "user\_fromAPI",      "name": "API",      "mail": "API@gmail.com",      "phoneNumber": "0530381254",      "password": "1234Deem$",      "person\_role": "user"  } |
| updateUser(User user) | <http://localhost:8080/userAPI/updateUser>  Body:  {          "id": 1102,          "username": "user\_fromAPI\_UPDATED",          "name": "API",          "mail": "API@gmail.com",          "phoneNumber": "0530381254",          "password": "$2a$10$OZqYqUa4S4f8HdBHrzfPbuwnJNwAS428/.0uRYG5NJ8ei4To2/UBq",          "person\_role": "user"  } |
| deleteUser(int id) | <http://localhost:8080/userAPI/deleteUser/1053> |

* Librarian

|  |  |
| --- | --- |
| getAllLibrarians() | <http://localhost:8080/librarianAPI/getAllLibrarians> |
| getLibrarian(int librarianId) | <http://localhost:8080/librarianAPI/getLibrarian/1052> |
| getLibrarianByIdOrUsername(int id, String username) | <http://localhost:8080/librarianAPI/getLibrarianByIdOrLibrarianname/10521/Lib_deem>  OR  <http://localhost:8080/librarianAPI/getLibrarianByIdOrLibrarianname/1052/Lib_de> |
| createLibrarian(librarian librarian) | <http://localhost:8080/register>  Body:  {      "username": "user\_fromAPI",      "name": "API",      "mail": "API@gmail.com",      "phoneNumber": "0530381254",      "password": "1234Deem$",      "person\_role": "librarian"  } |
| updateLibrarian(librarian librarian) | <http://localhost:8080/librarianAPI/updateLibrarian>  {      "id": 1152,      "username": "librarian\_fromAPI\_UPDATE",      "name": "API",      "mail": "API@gmail.com",      "phoneNumber": "0530381254",      "password": "1234Deem$",      "person\_role": "librarian"  } |
| deleteLibrarian(int id) | <http://localhost:8080/librarianAPI/deleteLibrarian/1154> |

* Book

|  |  |
| --- | --- |
| getAllBooks() | <http://localhost:8080/booksAPI/books> |
| getBook(int bookId) | <http://localhost:8080/booksAPI/books/1> |
| getBookByNameOrIdOrAuthorname(int id, String name, String authorName) |  |
| addBook(book book) | <http://localhost:8080/booksAPI/addBook>  body:  {      "title": "Book from API",      "authorName": "API",      "quantity": 23,      "borrowedQuantity": 0  } |
| addBooks(List<Book> books) | <http://localhost:8080/booksAPI/addBooks>  body:  [      {      "title": "Book 1",      "authorName": "API",      "quantity": 23,      "borrowedQuantity": 0      },      {      "title": "Book 2",      "authorName": "API",      "quantity": 23,      "borrowedQuantity": 0      }  ] |
| updateBook(book book) | <http://localhost:8080/booksAPI/updateBook>  body:      {          "id": 1203,          "title": "Book 1 Updated",          "authorName": "API",          "quantity": 23,          "borrowedQuantity": 0  } |
| deleteBook(int id) | <http://localhost:8080/booksAPI/deleteBook/12> |

* Borrow

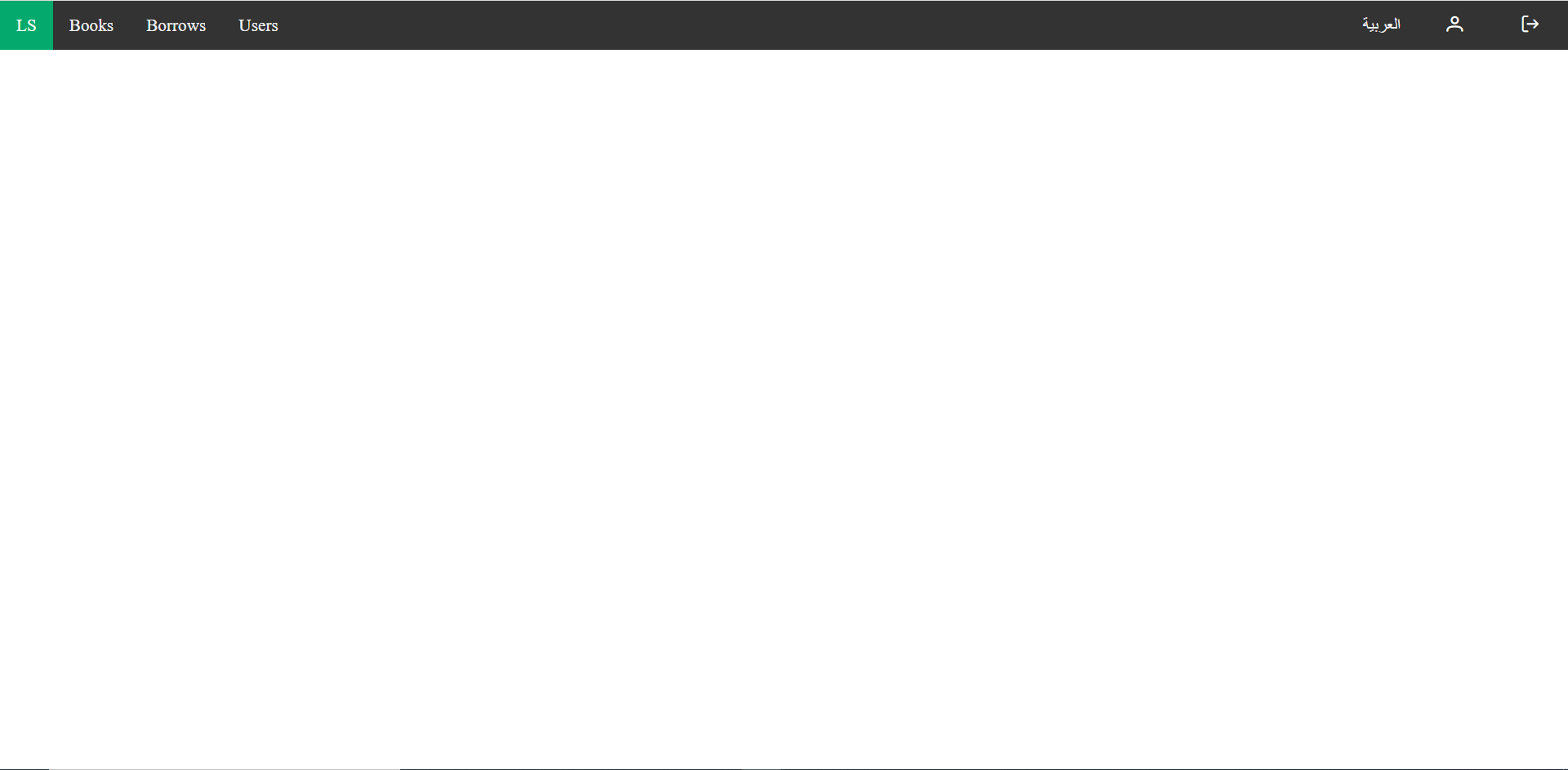
|  |  |
| --- | --- |
| getAllBorrows() | <http://localhost:8080/borrowAPI/borrows> |
| getBorrow(int id) | <http://localhost:8080/borrowAPI/borrows/1002> |
| getBorrowByIdOrUserIdOrBookId(int id, Long userId, Long bookId) | <http://localhost:8080/borrowAPI/getBorrowByIdOrUserIdOrBookId/10023/12/953> |
| addBorrow(borrow borrow) | <http://localhost:8080/borrowAPI/addborrow>  body:  {      "book": {              "id": 953,              "title": "التحدث بالاسبانية",              "authorName": "تهاني",              "quantity": 43,              "borrowedQuantity": 1          }          ,      "person": {              "id": 1,              "username": "deem",              "name": "d",              "mail": "deem\_sam@gmail.com",              "phoneNumber": "0530381254",              "password": "$2a$10$aOhbDIcEvsFe9dECg46uTuN7w.Ev5nJ28ljRgE0qkww1I6eH2.yau",              "person\_role": "user"          }  } |
| updateBorrow(borrow borrow) | <http://localhost:8080/borrowAPI/updateborrow>  body:  {      "id": 0,      "book": {              "id": 953,              "title": "التحدث بالاسبانية",              "authorName": "تهاني",              "quantity": 43,              "borrowedQuantity": 1          }          ,      "person": {              "id": 1,              "username": "deem",              "name": "d",              "mail": "deem\_sam@gmail.com",              "phoneNumber": "0530381254",              "password": "$2a$10$aOhbDIcEvsFe9dECg46uTuN7w.Ev5nJ28ljRgE0qkww1I6eH2.yau",              "person\_role": "user"          }  } |
| deleteBorrow(int id) | <http://localhost:8080/borrowAPI/deleteBorrow/1002> |

# Front end

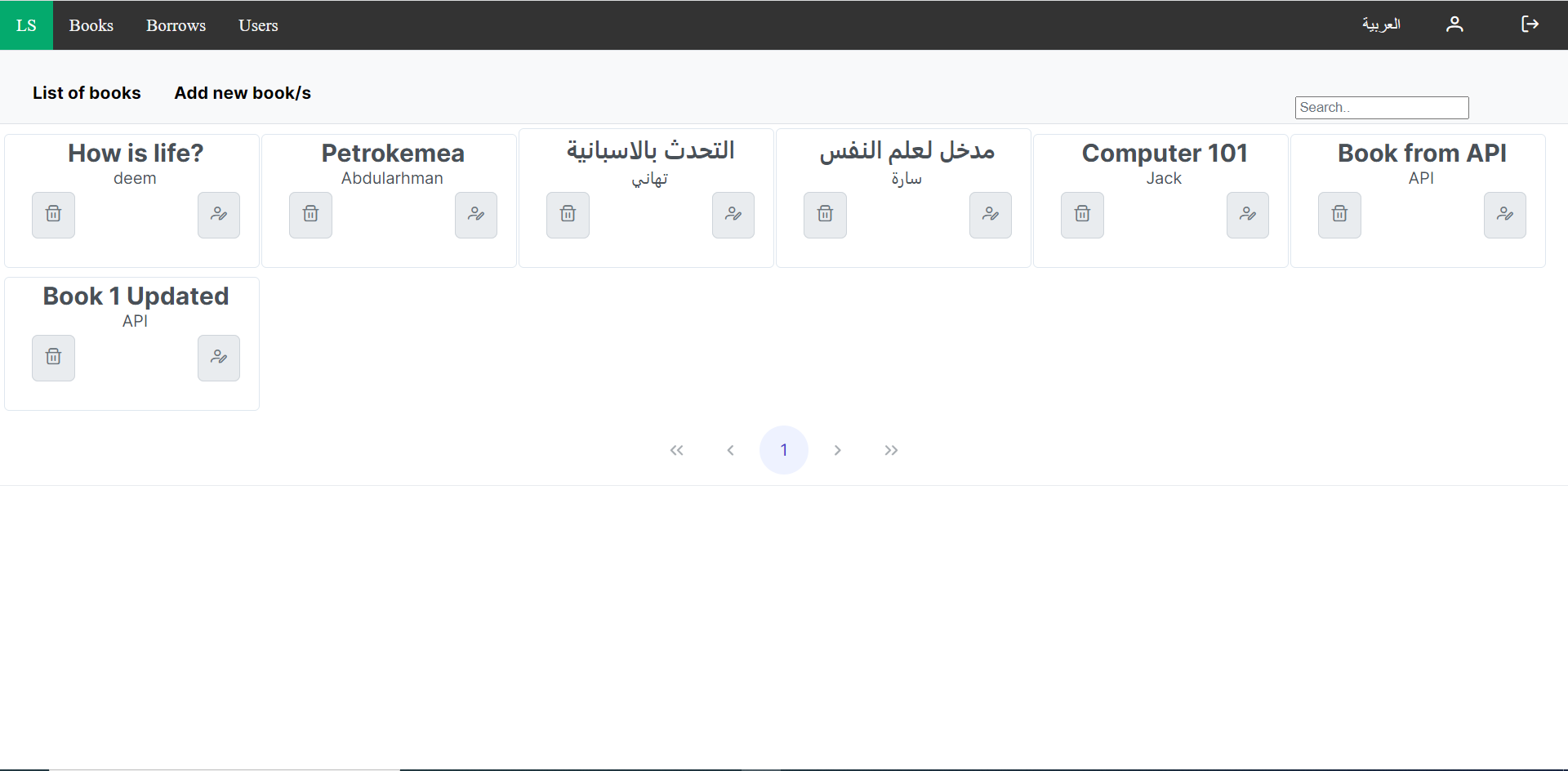
|  |  |
| --- | --- |
| PrimeNG | Components |
| TranslateModule | Use translate pipe |
| Field validation | Using pattern and required attribute |
| Wizard | MatStepperModule , MatNativeDateModule |

## Front end UIs:

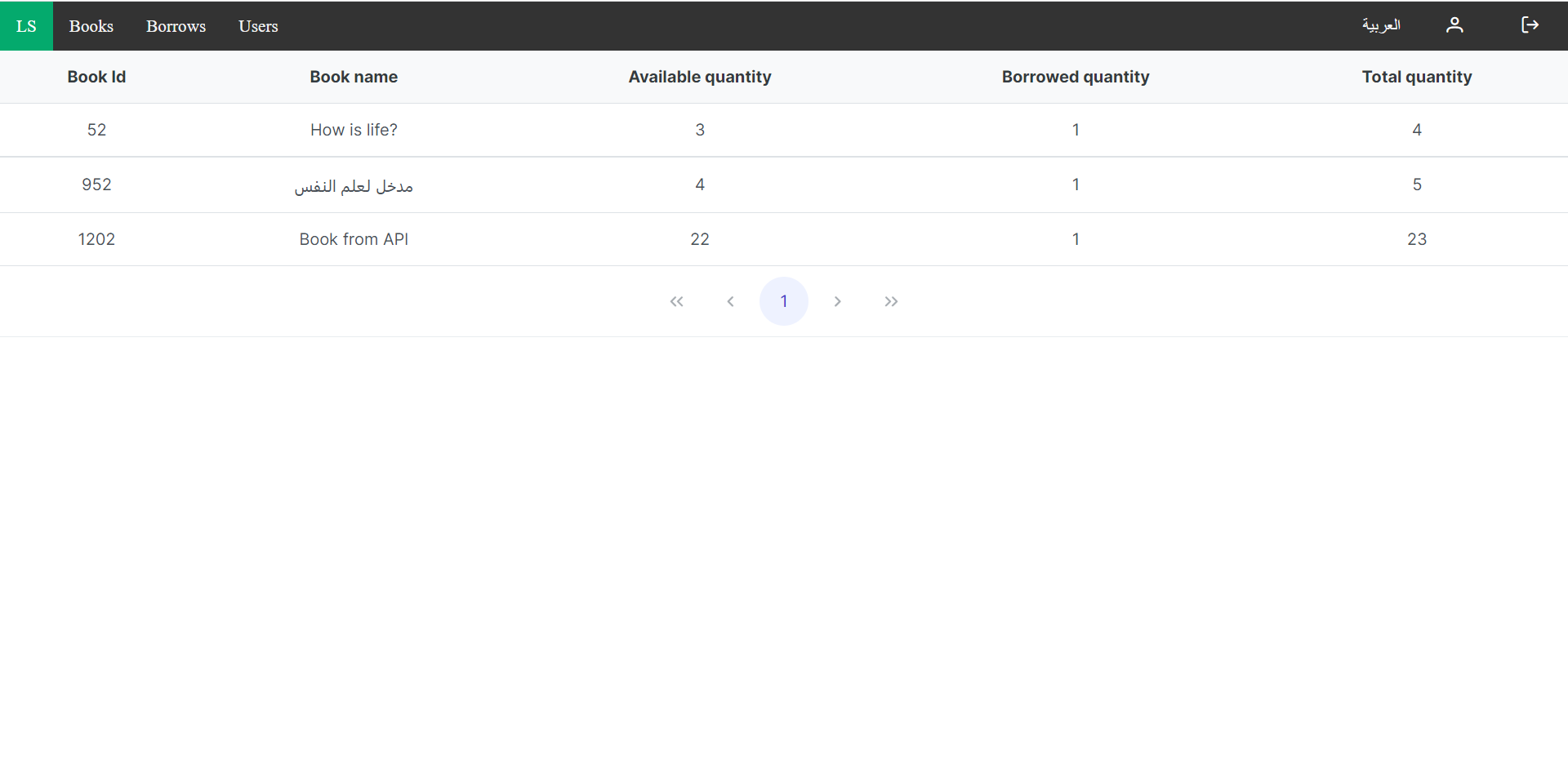
### After login as librarian:



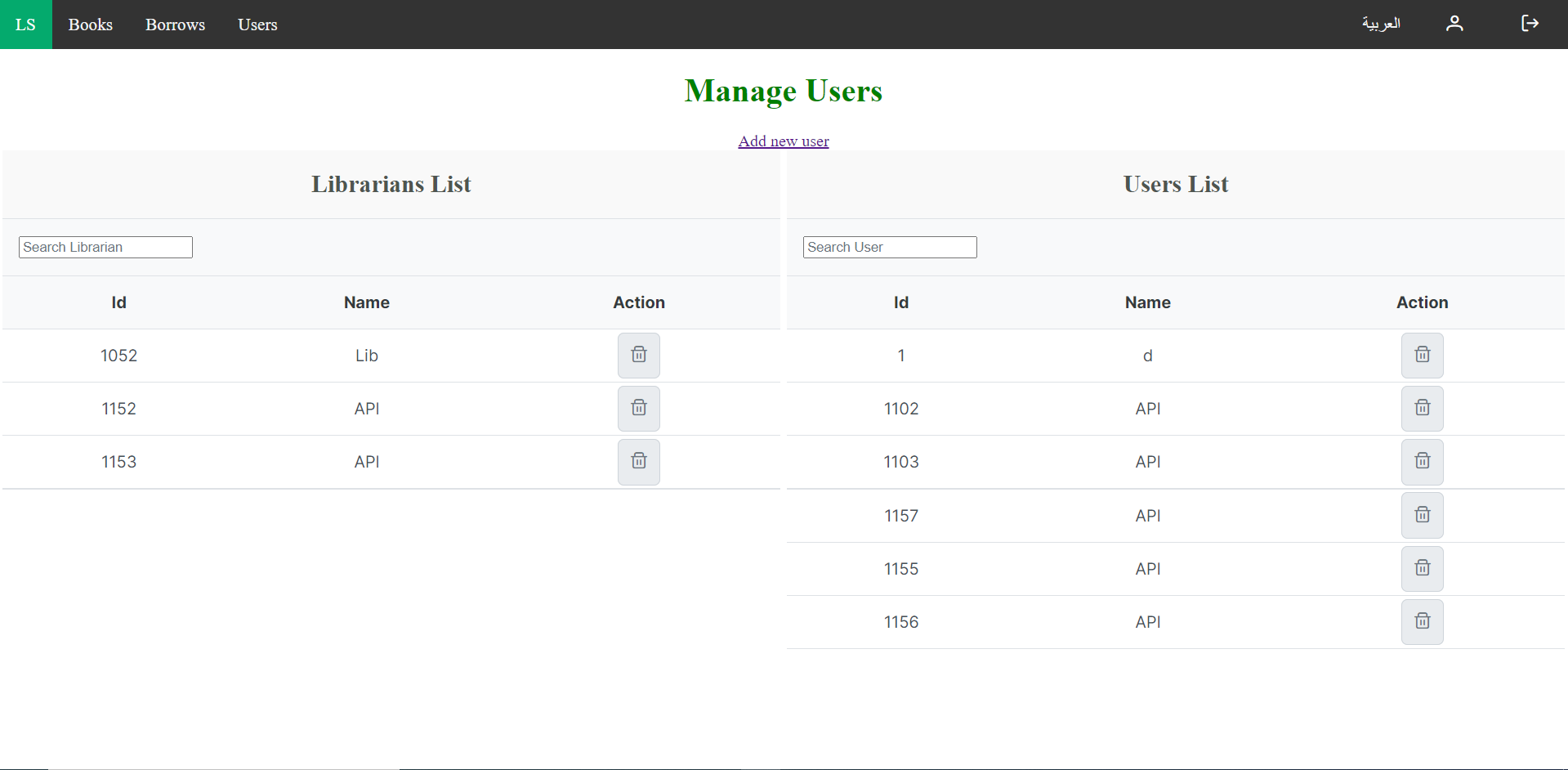
First app page.



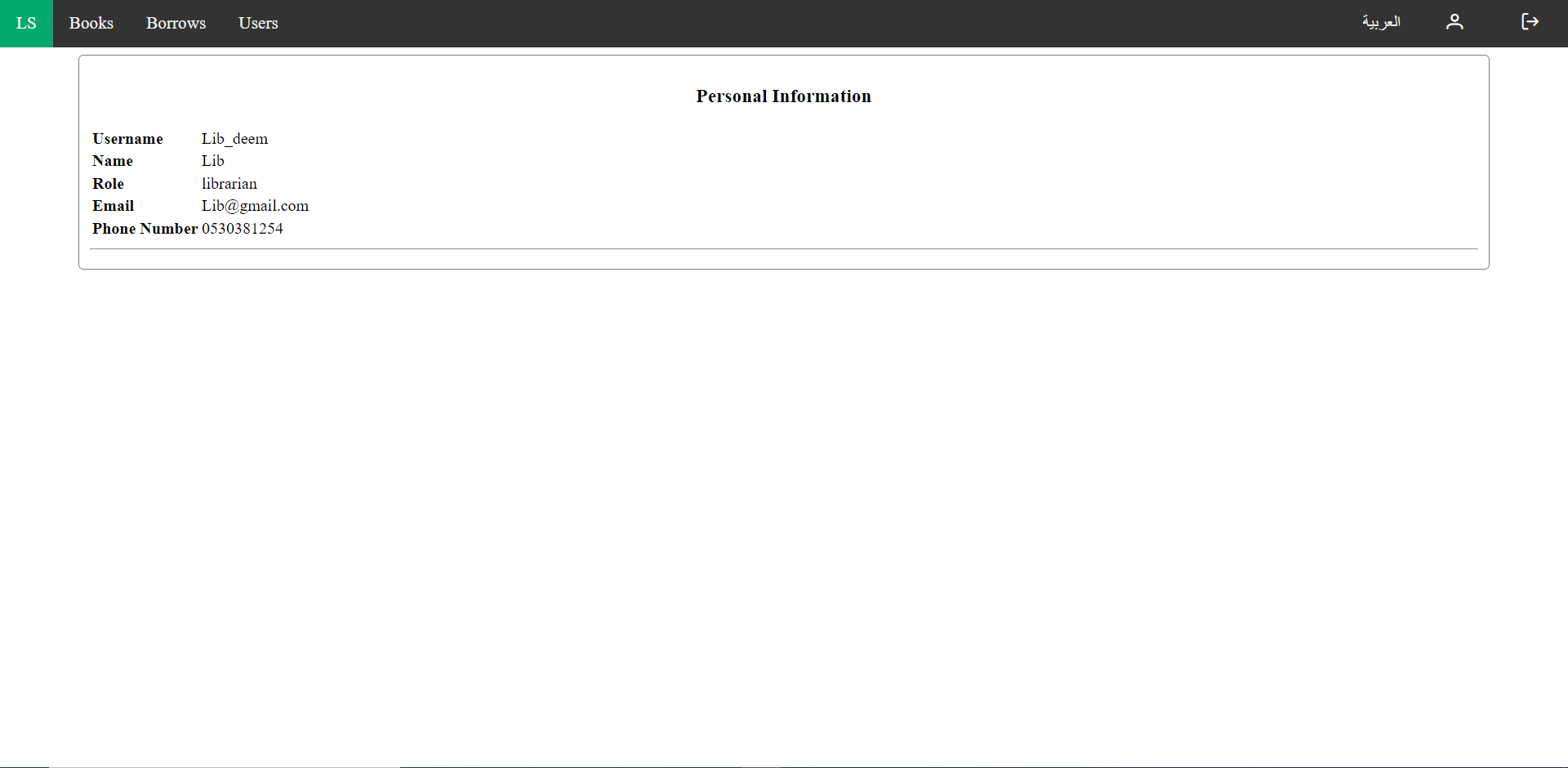
Books page.



Borrow page.



Users page.

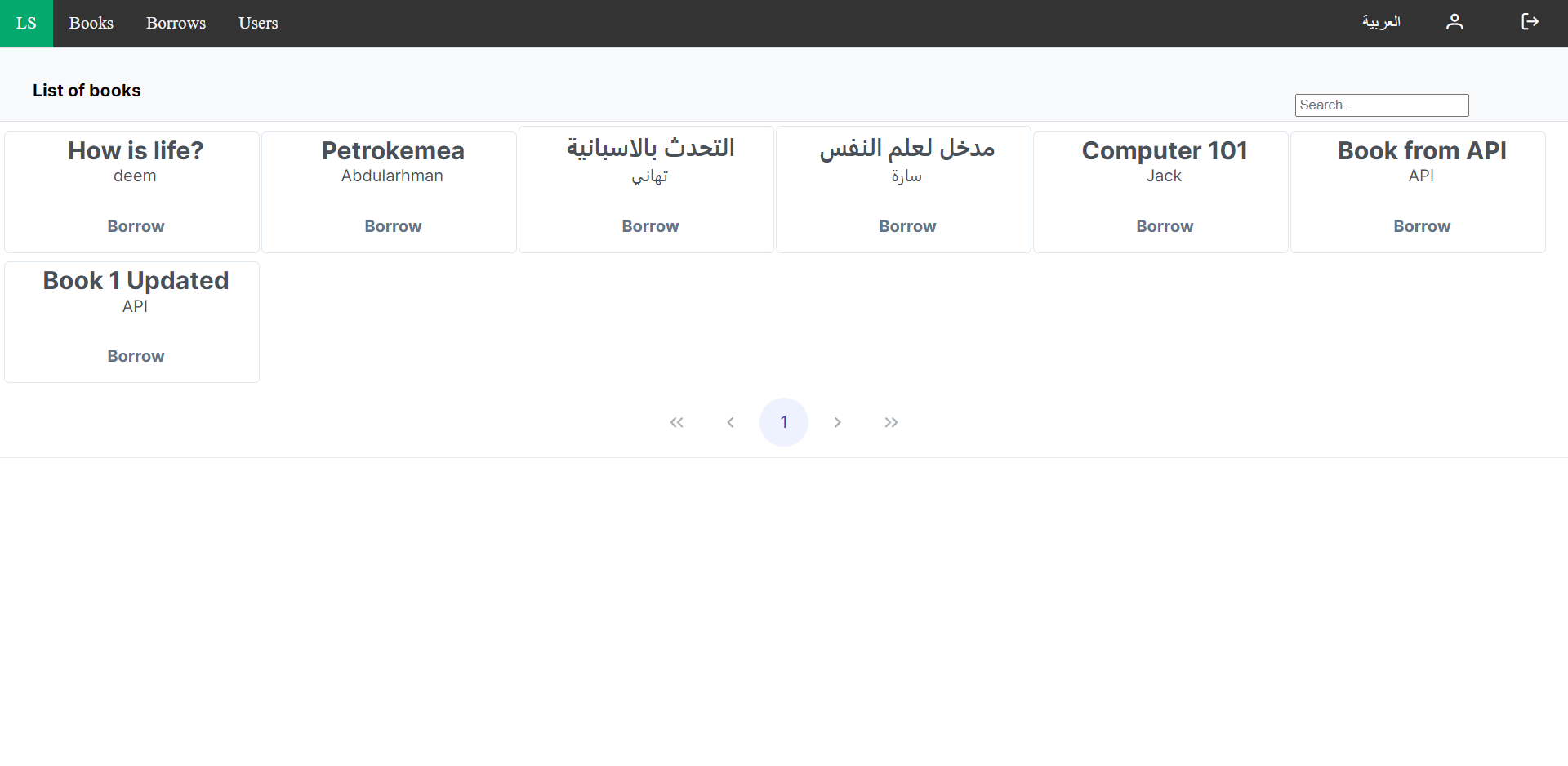


Account page.

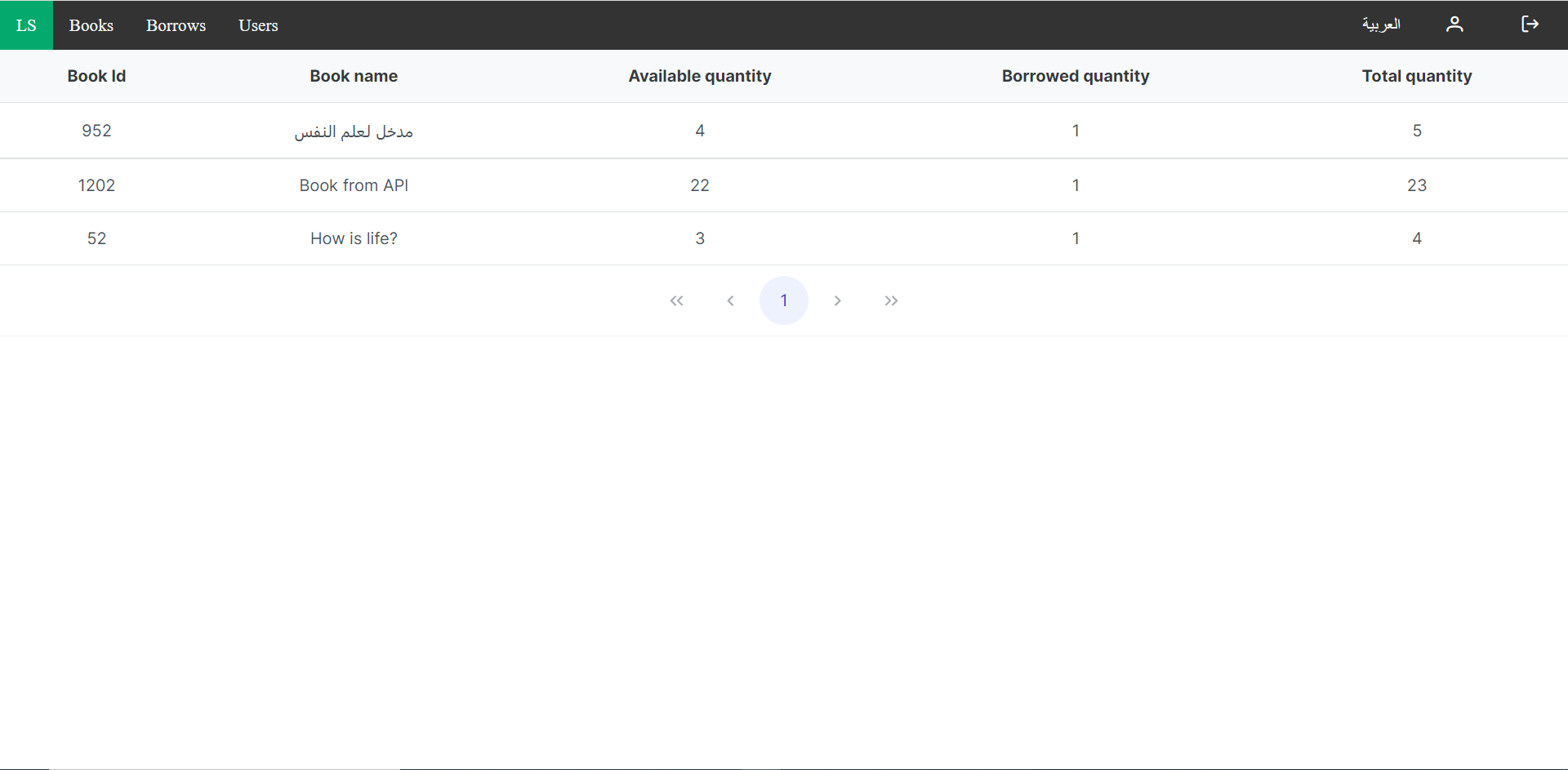
### After login as user:



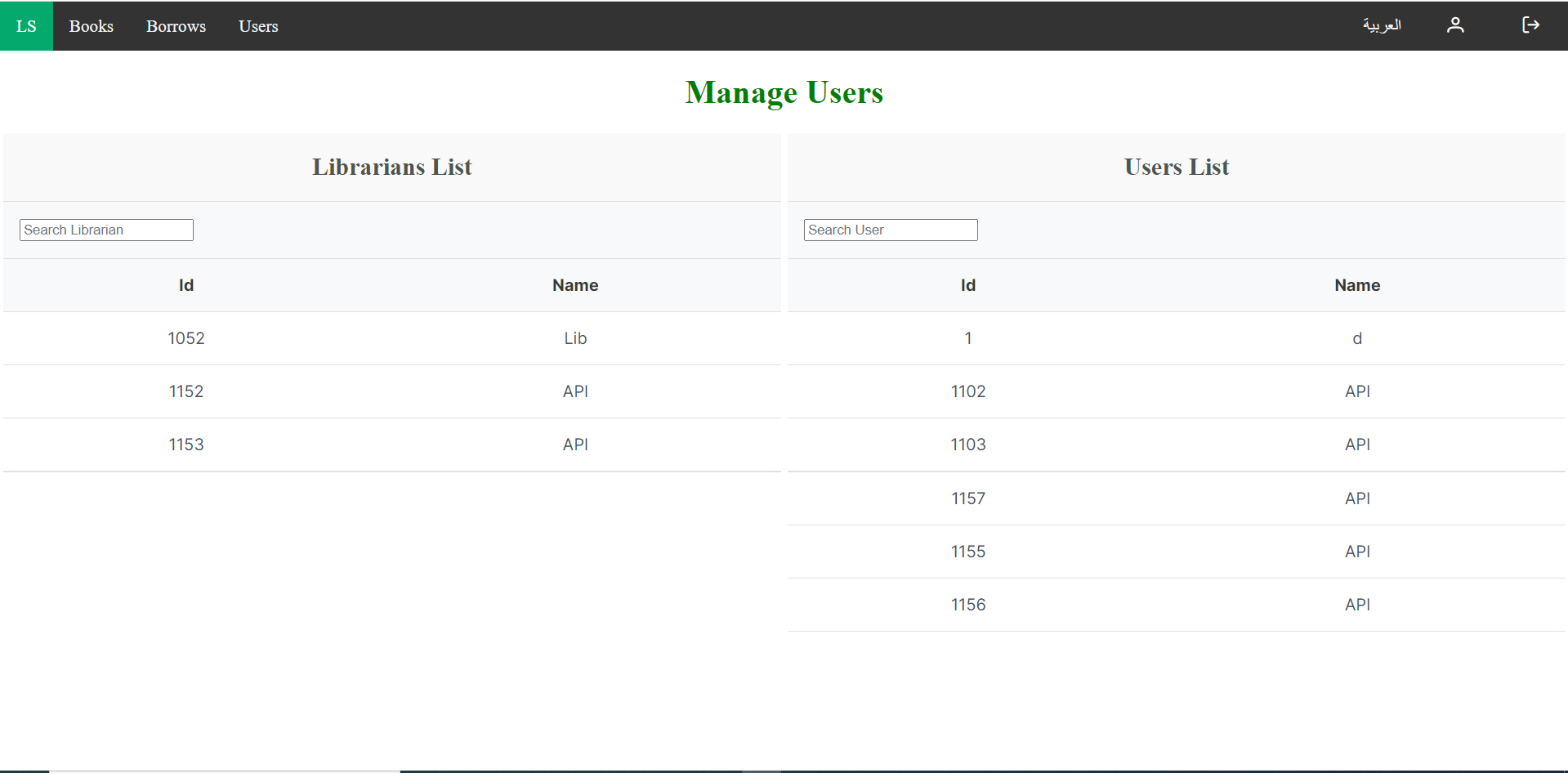
First app page.



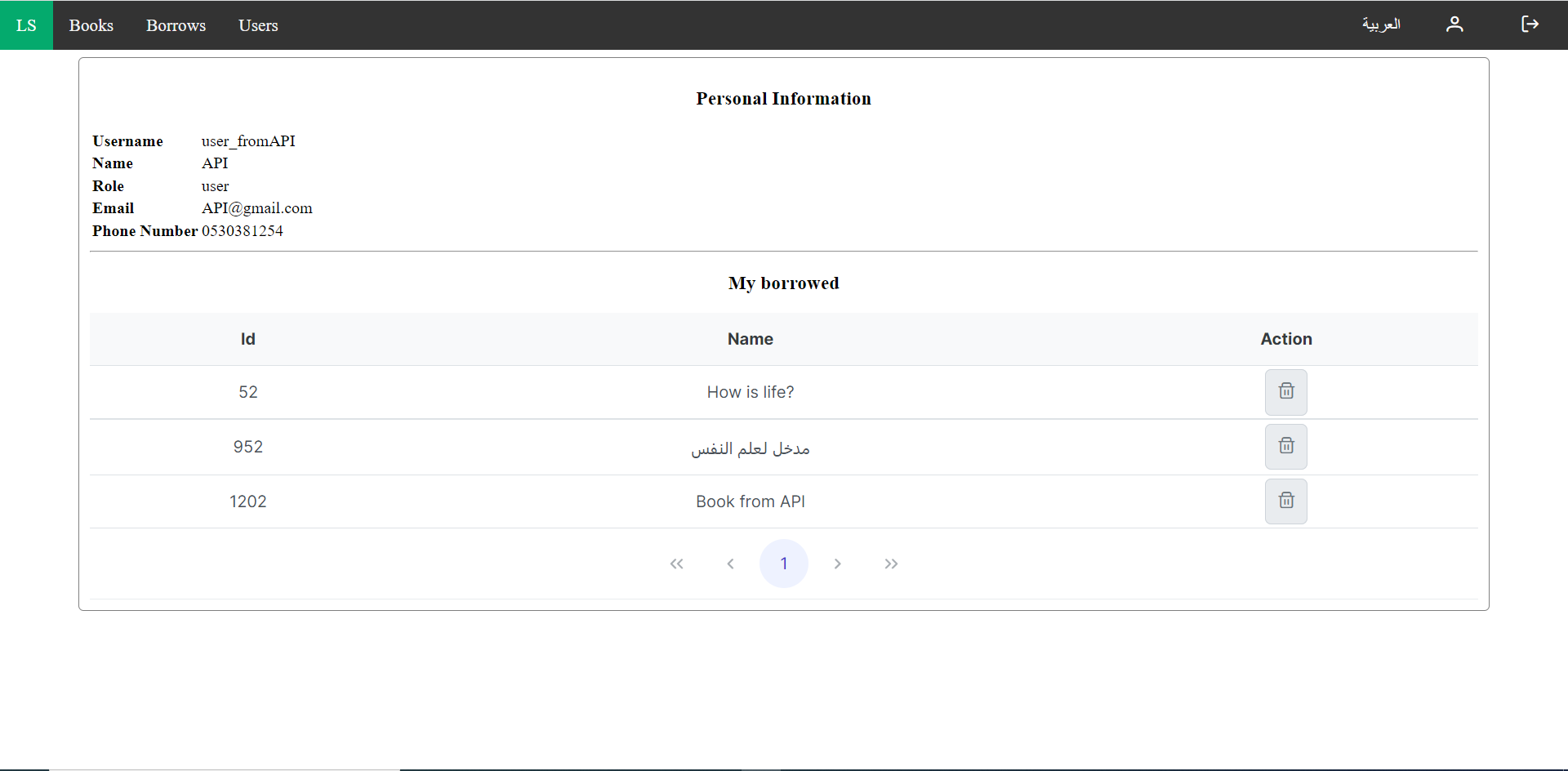
Books page.



Borrow page.



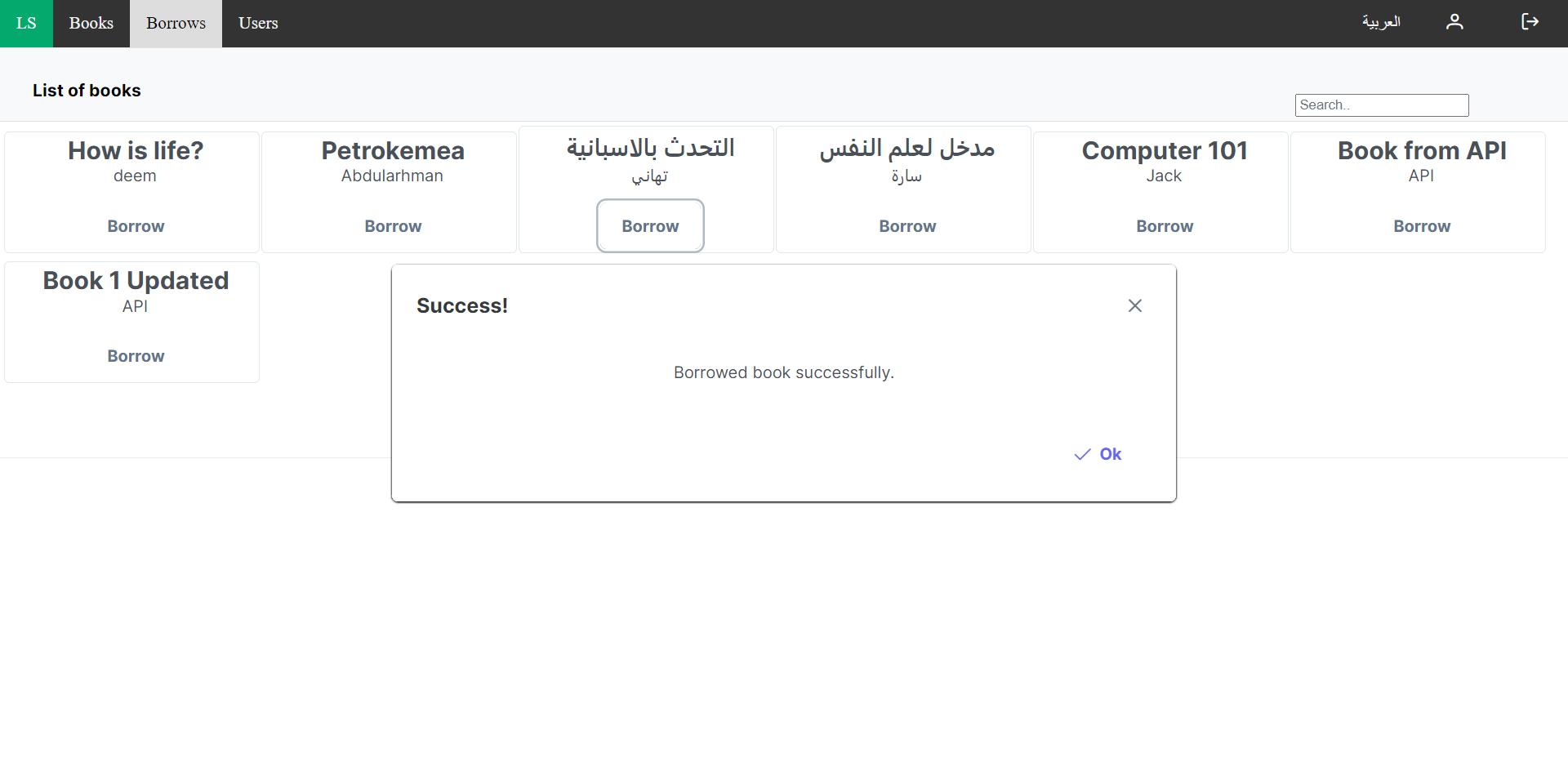
Users page.



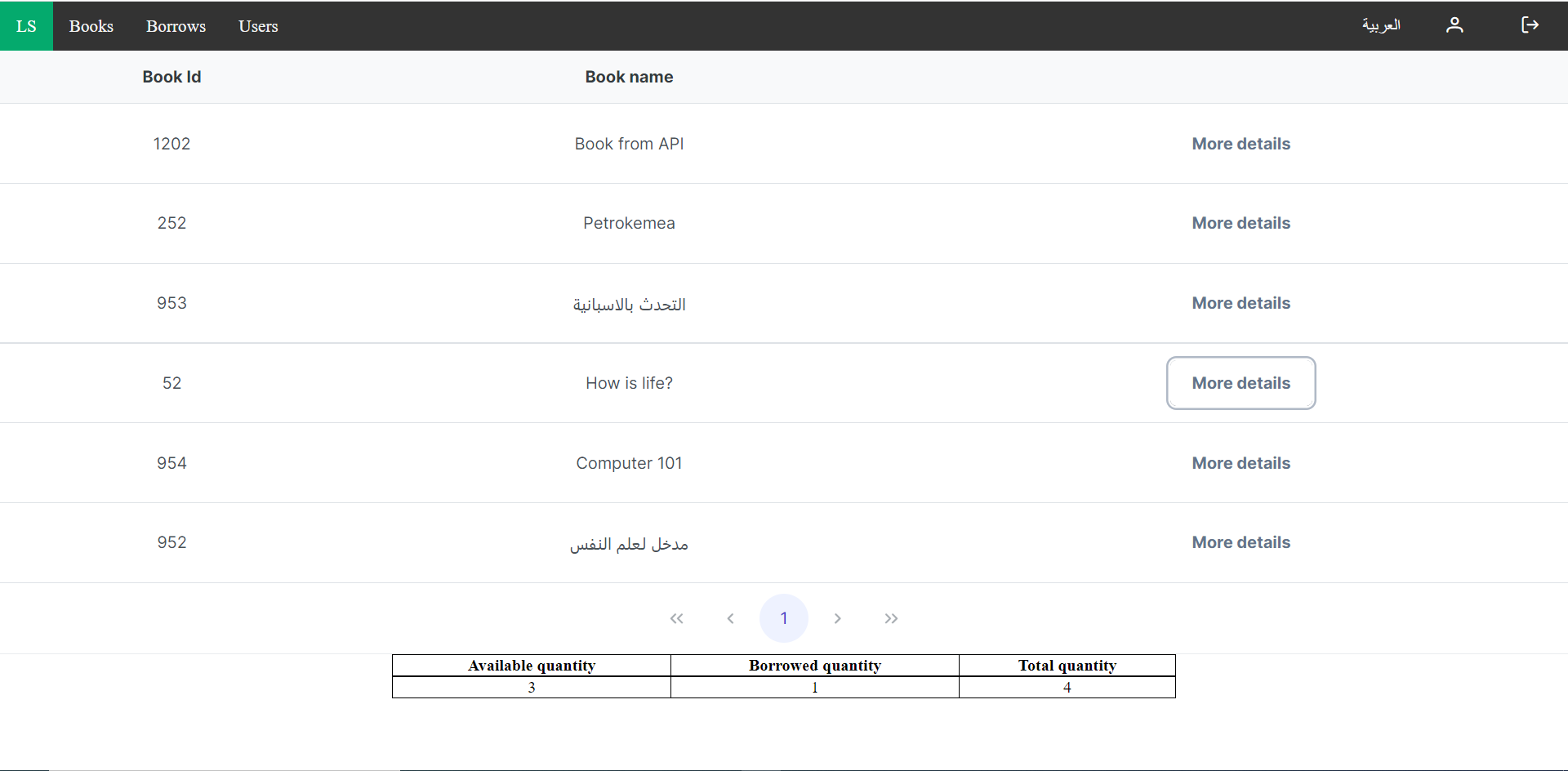
Account page.

# ENG.Donia comments:

### Add primeNG/Angular martial dialog component

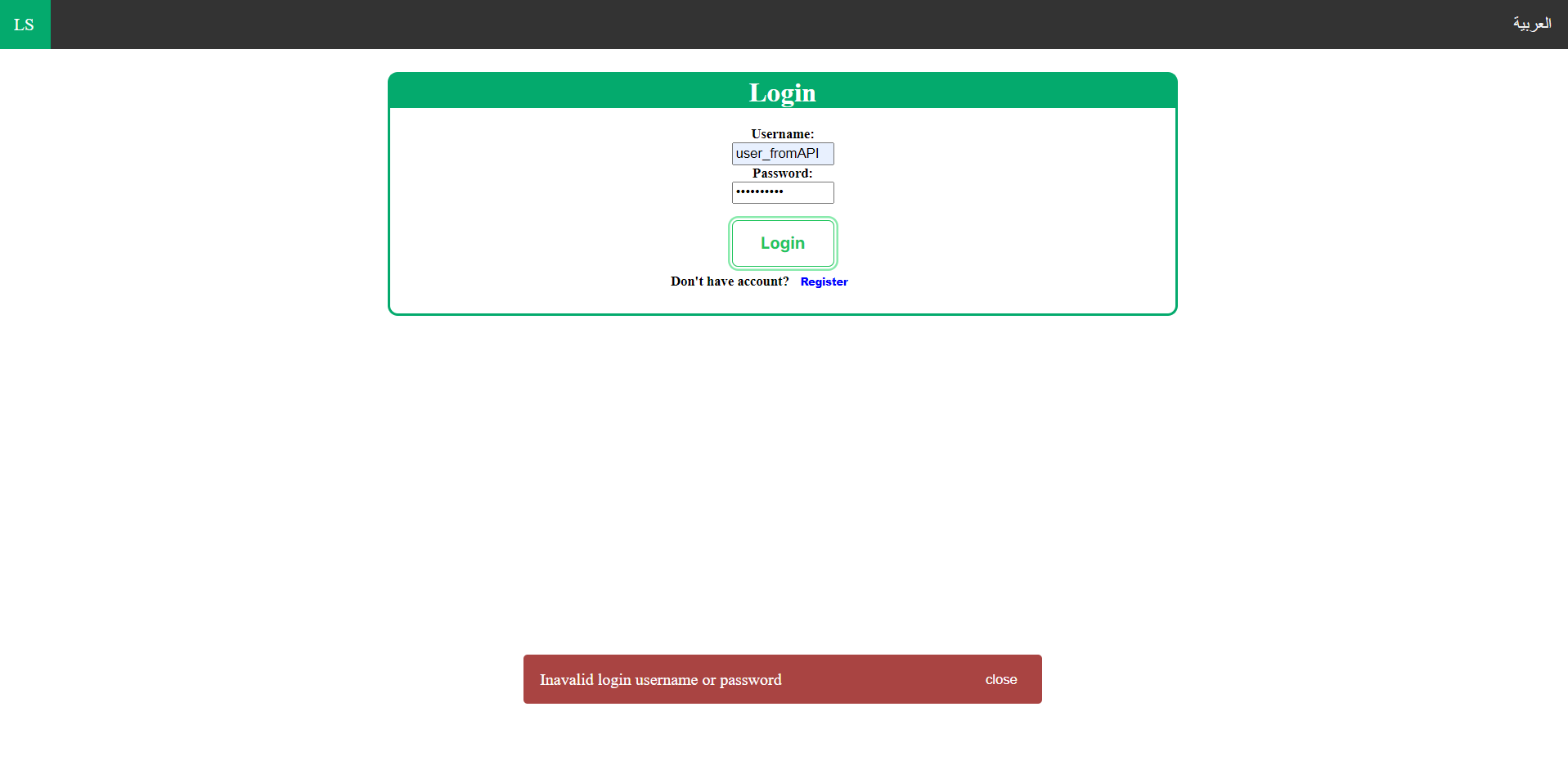
dialog component.

## Master Detail table



Master-detail in Borrow page.

### Snackbar



Toast message.