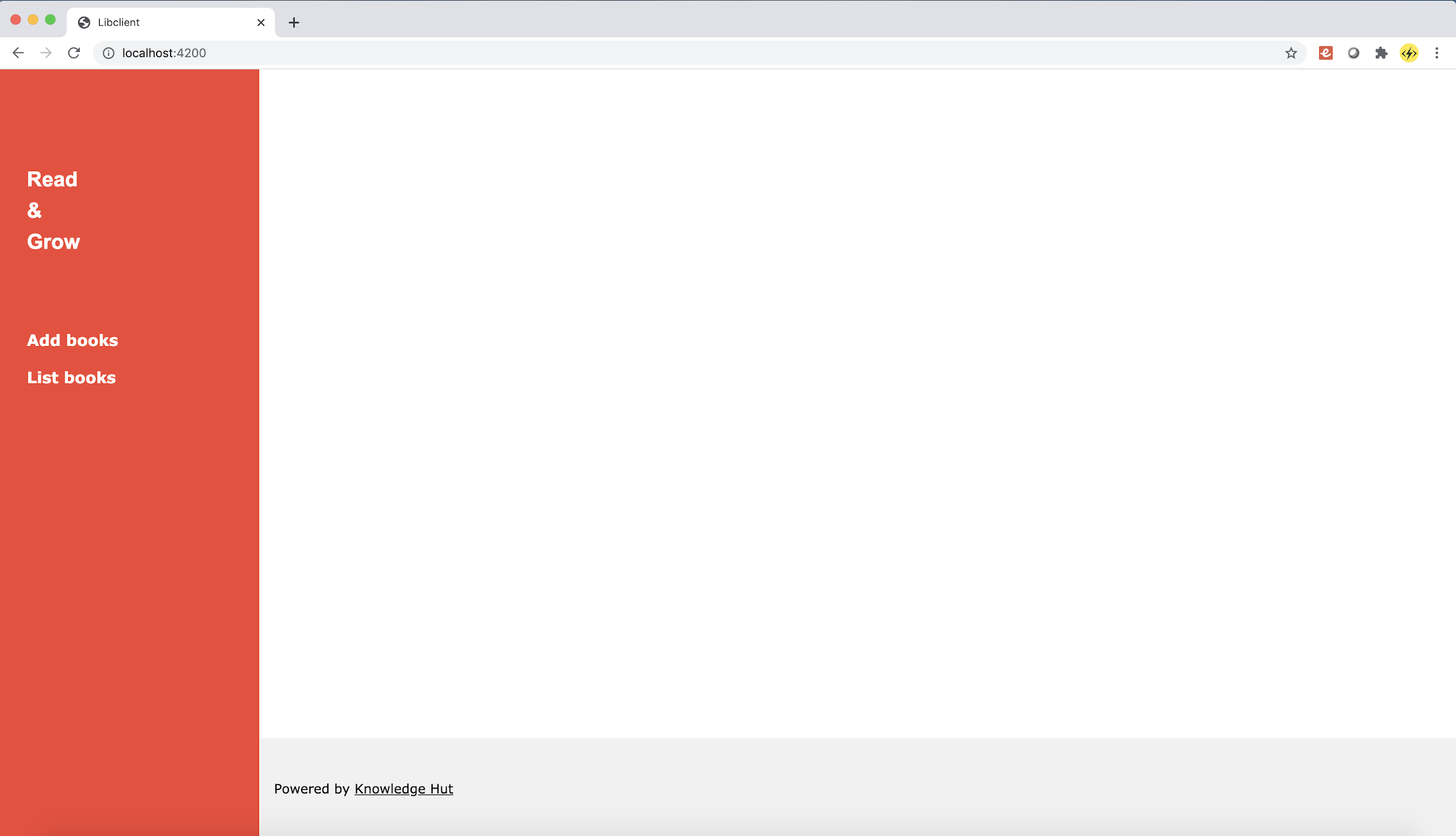
## Title

A Library management application that allows the user to view a list of books and add new books.

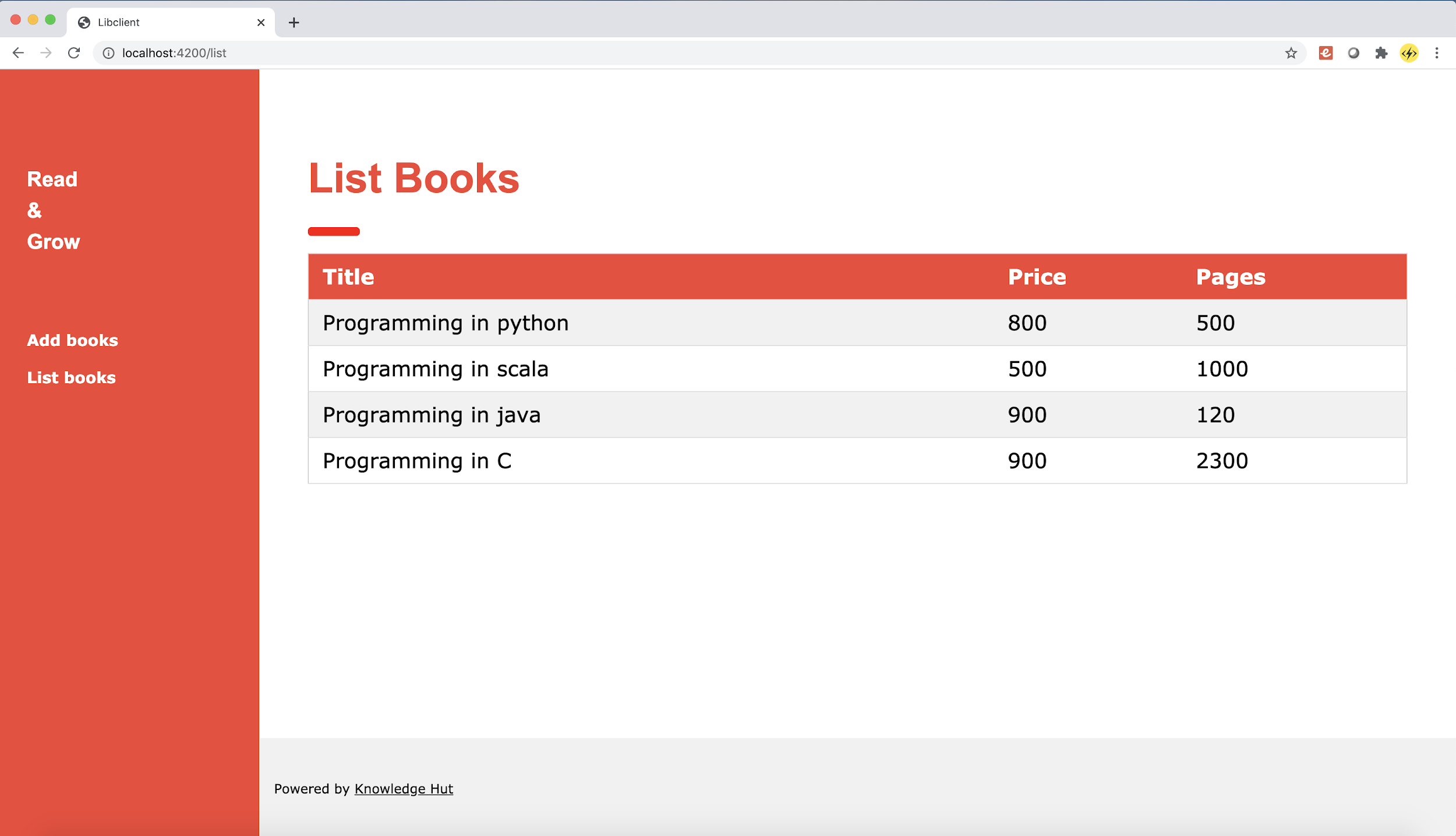
## Landing Page



The above page is shown when the user hits the root url “/” of the application. A very bare bones landing page with Two side menus

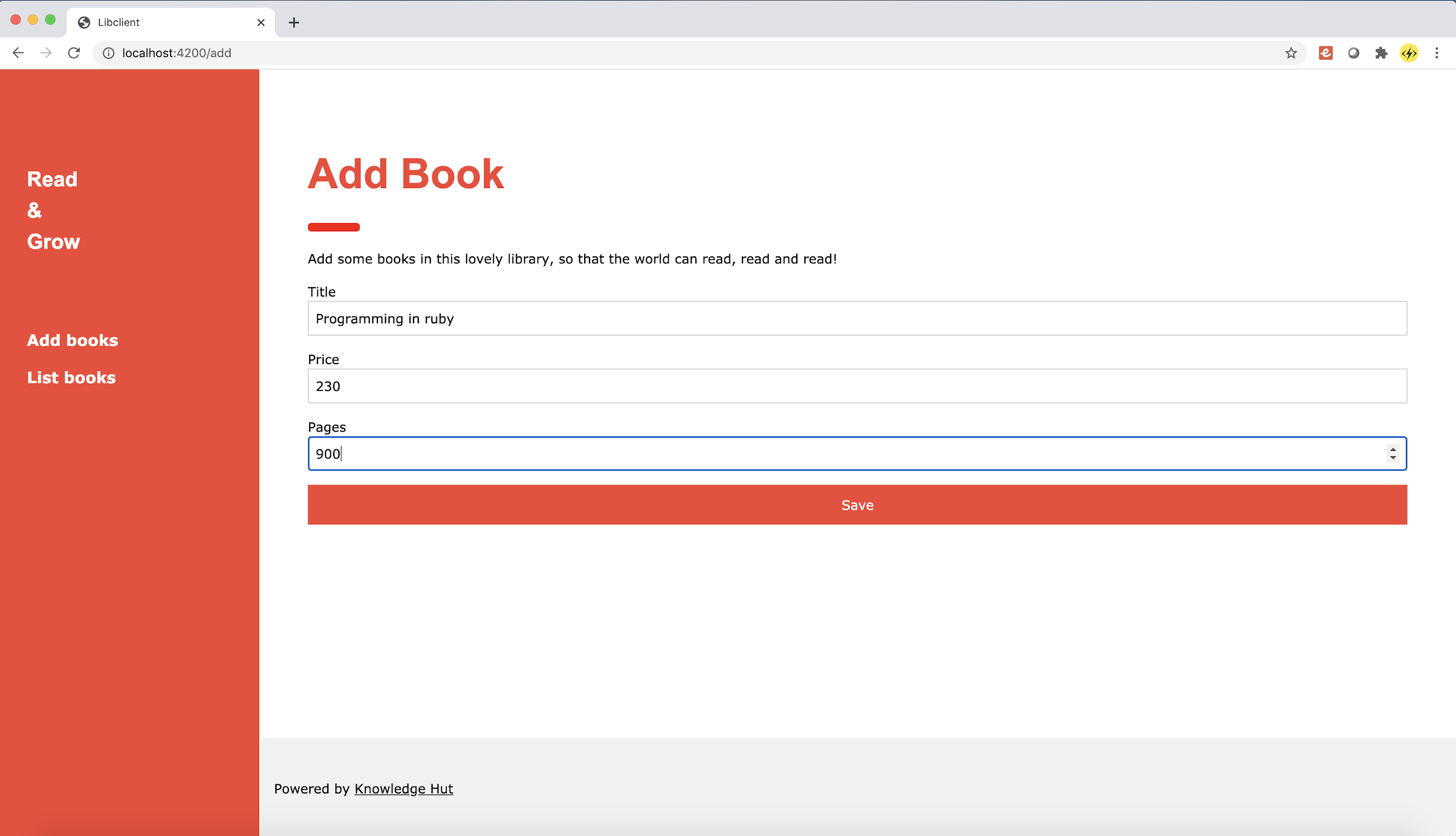
1. Add Books
2. List Books

## List Books



The above page is shown when the user clicks on the List books menu option in the left hand side menu. Can also be directly accessed using the url “/list”

## Create Book



The above page is shown when the user clicks on the Add books menu option in the left hand side menu. Can also be directly accessed using the url “/add”

## REST Web service API endpoints

1. GET list of books
   1. Method: GET
   2. URL : /books
   3. JSON Response expected

[

{

"id": 1,

"title": "Programming in java",

"price": 900,

"pages": 400

},

{

"id": 2,

"title": "Programming in scala",

"price": 340,

"pages": 800

}

]

* 1. Response code: 200

1. POST a book
   1. Method : POST
   2. URL: /books
   3. JSON Request expected

{

"title": "Programming in C++",

"price": 900,

"pages": 450

}

* 1. JSON Response expected

{

“Id”: 5,

"title": "Programming in C++",

"price": 900,

"pages": 450

}

* 1. Response code: 201

## Basic Instructions

1. Use Spring integration with JPA and Hibernate for data persistence
2. Use the MYSQL database.
3. Follow the MVC folders that have been already created in the starter projects
4. The Angular application in the starter project is already pre built.
5. After building the Spring REST web service, please update the service urls in the angular project, at the following locations
   1. list-books.component.ts
   2. add-books.component.ts