

Database search from Node

Welcome to this lab activity

In this lab activity, you will explore how to implement a search page that searches the database and shows matching results.

Task 1: Make a copy of the previous lab code

You will start with your previous Node.js lab code.

1. Make a copy of the folder `17_inserting_data_to_database` from your previous lab and call it `18_database_search`.
2. Open the new folder in Visual Studio Code.
3. Run the code and confirm that the list page still works (refer to the previous lab if you need a reminder of how to do this).

Task 2: Modify the existing 'search' code in your web application

Your task is to modify your search functionality. When the user enters a search term in the search form, your application should perform a search on the database and then show any matching books in the `/list` web page.

4. Modify the handler for the `/search_result` route as follows:

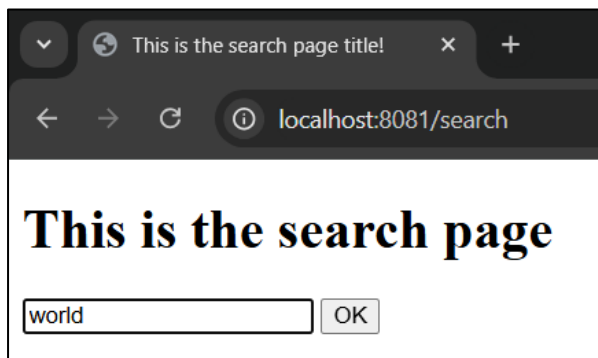
```
router.get("/search_result", function (req, res, next) {
  // Search the database
  let sqlquery = "SELECT * FROM books WHERE name LIKE ?";

  // Execute sql query
  let word = ['%' + req.query.search_text + '%'];
  db.query(sqlquery, word, (err, result) => {
    if (err) {
      next(err);
    }
    else if (result.length==0) {
      res.send("No book matches the search term " + word);
    }
    else {
      res.render ('list.ejs', {availableBooks:result});
    }
  });
});
```

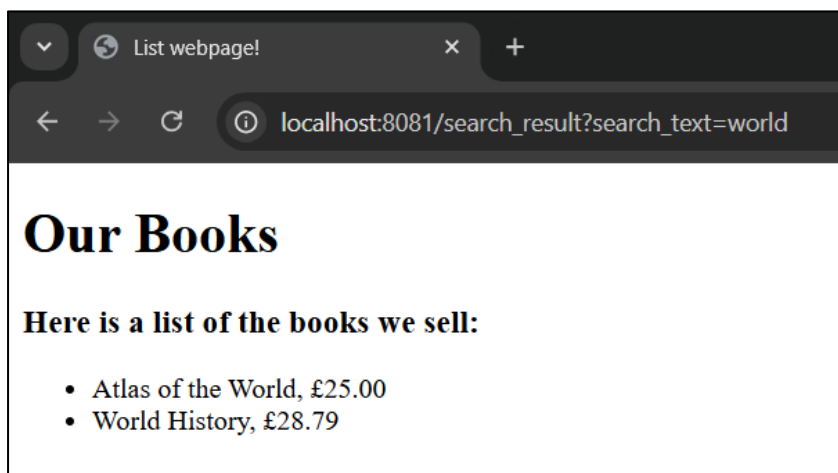
```
    }  
  });  
});
```

Task 3: Test your app

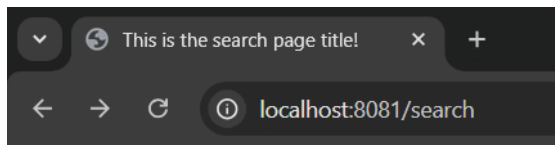
5. Run the app and browse to the `/search` route. Enter a search term that matches one or more books, for example:



You should see a list of books:

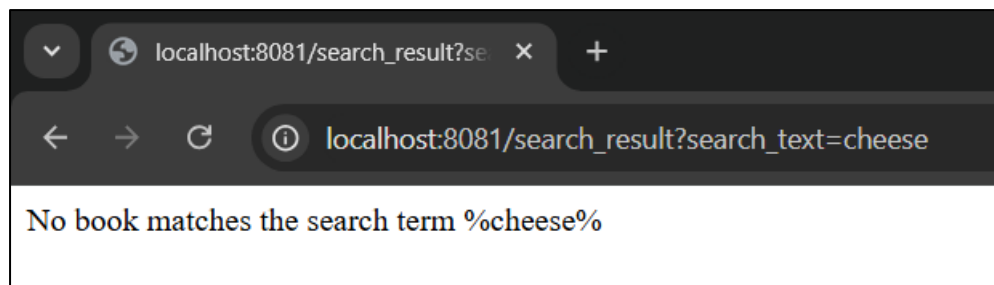


6. Go back to your search page and enter a search term for a book that doesn't exist, for example:



This is the search page

Confirm that you get an appropriate error message:



Task 4: Explore further

When tackling these lab activities, it's always good to stretch yourself by doing some research and attempting some changes on your own.

Change message from 'Here is a list of the books we sell' to 'Here is a list of books matching 'world''. Do this dynamically, so both the `/list` and the `/search_result` routes use the same `list.ejs` page.

End of lab

Congratulations on completing this lab.

In the next lab, you will be designing a data model to fulfil a set of requirements.