Work Breakdown Structure.

**Create Database:** (Estimated time : 4h)

Create tables of Books , Students, Teachers in database and populate some data in it

**Create APIs:** (Estimated time : 1.5 days)

CRUD Operations

1. **For Books:** (Estimated time : 5 hours)

Create api to list all the books available.

Create api to list specific book by id

Create api to delete book by id

Create api to update book data by id.

1. **For Teachers**: (Estimated time : 5 hours)

Create api to list all the teachers available in the db.

Create api to list specific teachers by id

Create api to delete teacher by id

Create api to update teachers data by id

1. **For Student:** (Estimated time :5 hours)

Create api to list all the Students available.

Create api to list specific Student by id

Create api to delete Student by id

Create api to update Student data by id.

**Front-End** (Estimated time : 1day)

Create 3 folders: 1. Books 2. Teachers 3. Students

Each folder will have 3 views .html file

* **Insertbooks.html** -> getting data from user in the fields of from and send it to the database (patch) |Estimated time: 3h
* **Displaybooks.html** -> displaying all the data from the database in table form. |Estimated time: 3h
* **Displayspecificbooks.html** -> getting id from user and display the data which matches the id in from of table (get/:id) |Estimated time: 3h

**Integration** (Estimate 2days)

Connecting backend to the frontend (Estimate 5h)

Making routes for each operation separately for separate table (Estimate 5h)

3 routes per table 1 -insert 2- update 3 -display (Estimate 5h)

Js code for taking data from backend to the front end and display then in a frontend in table from. (Estimate 5h)

**Conclusion:**

This project have took 5 days for successful implementation.

Notes:

Ensure that your API routes and HTML routes are distinct and don't conflict with each other. For example, make sure your API routes start with **/api/** and your HTML routes don't.