**EziTak***A Task Management Website*

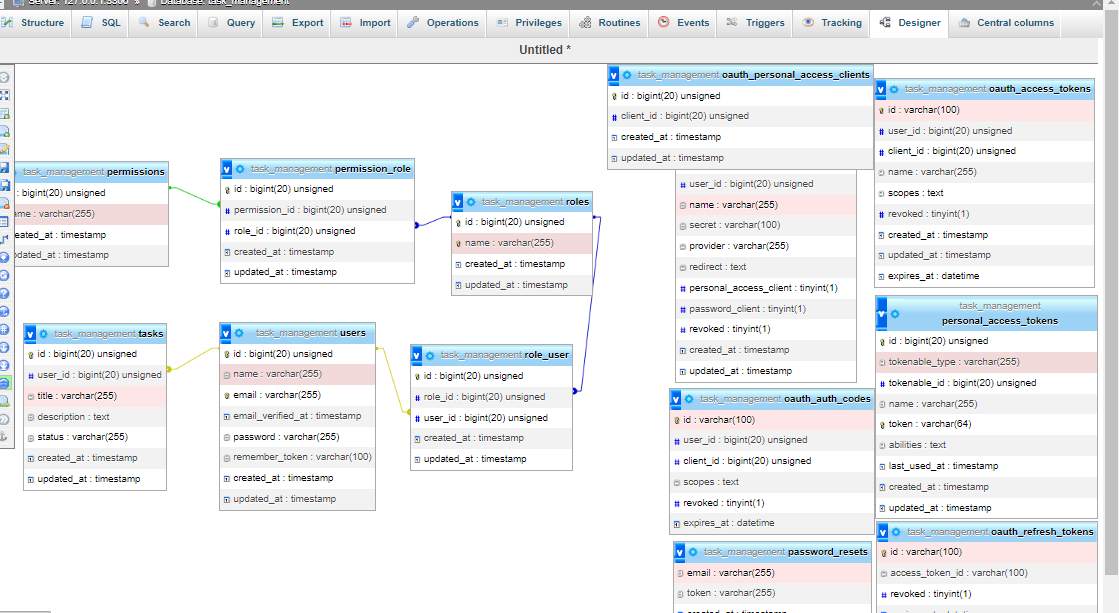
=====================================================================================

**Implementation:**

* Database Design and Migrations
* Models and Relations
* User interfaces using Blade files
* Controllers for back end logics
* CRUD operations for users
* Routes and middleware’s
* Authentication (Login and Registration) for users
* Ajax and models (for seamless deletion and updating of task)
* Web hocks (E-mailing on success events)
* REST API for CRUD operation

**Database Design and Migrations:**

Designed and developed a database according to the needs of project make necessary migrations with appropriate attributes within them

****

**Models and Relations:**

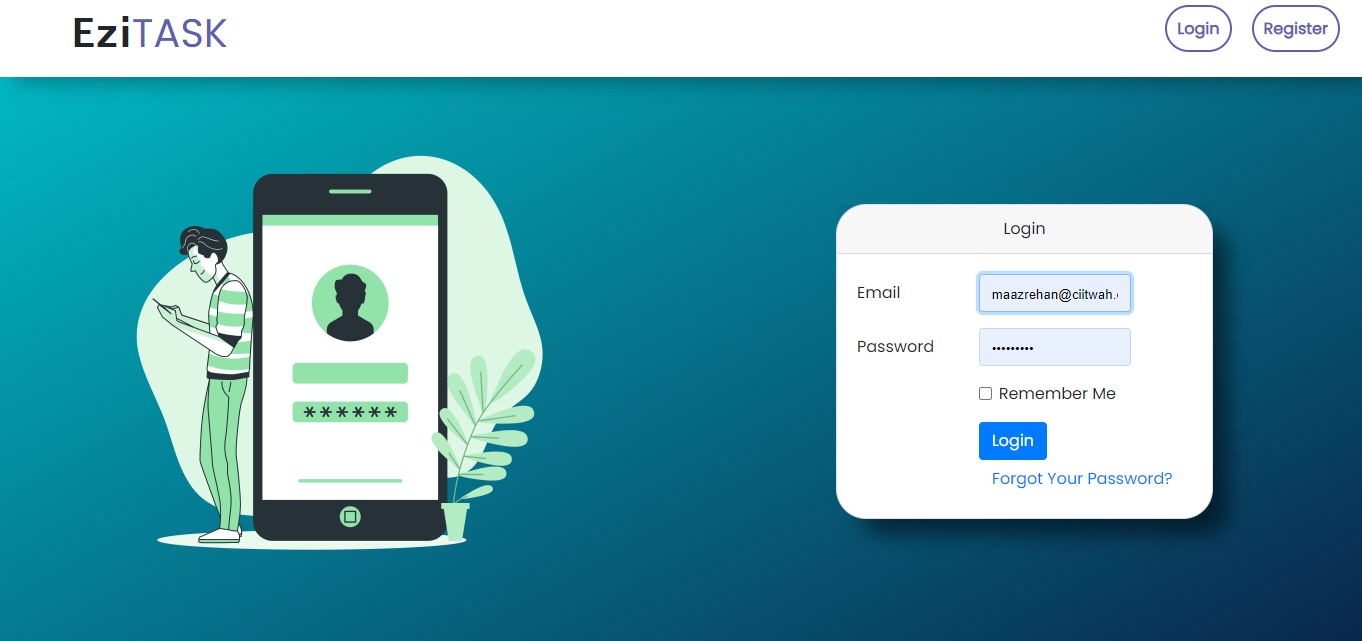
Four main models are made as user, task, roles, permission and the their relations are made accordingly as shown above in the database design

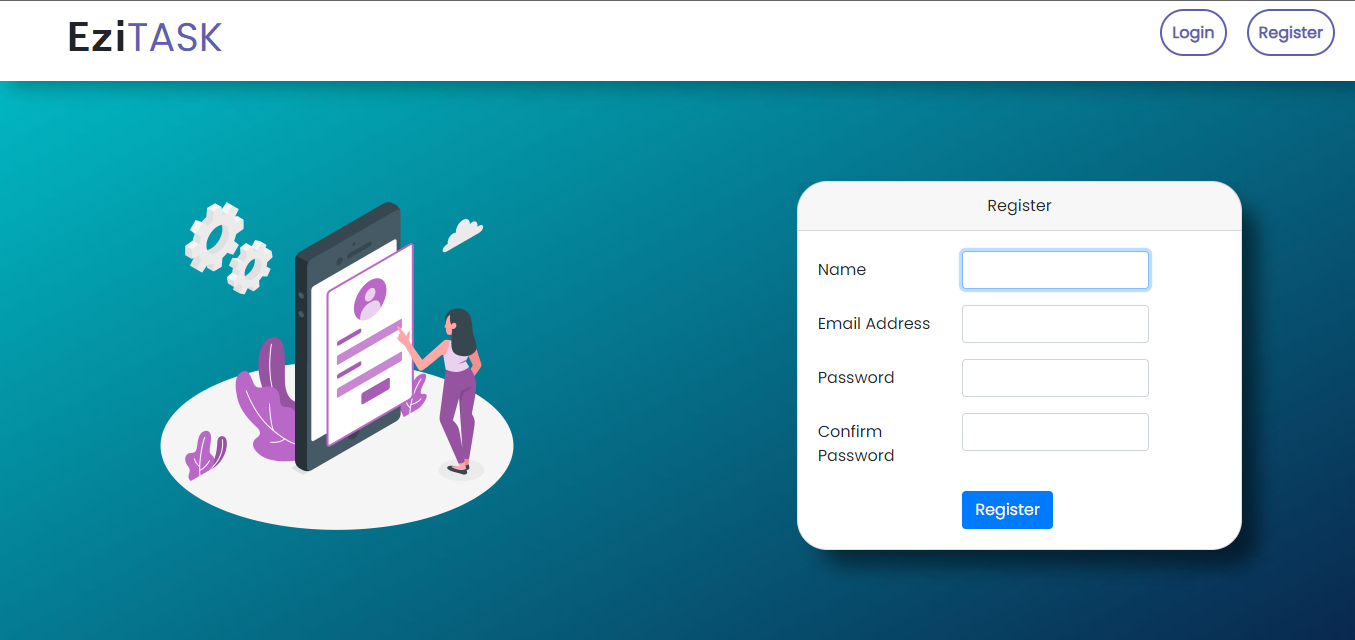
**User interfaces using Blade files**:

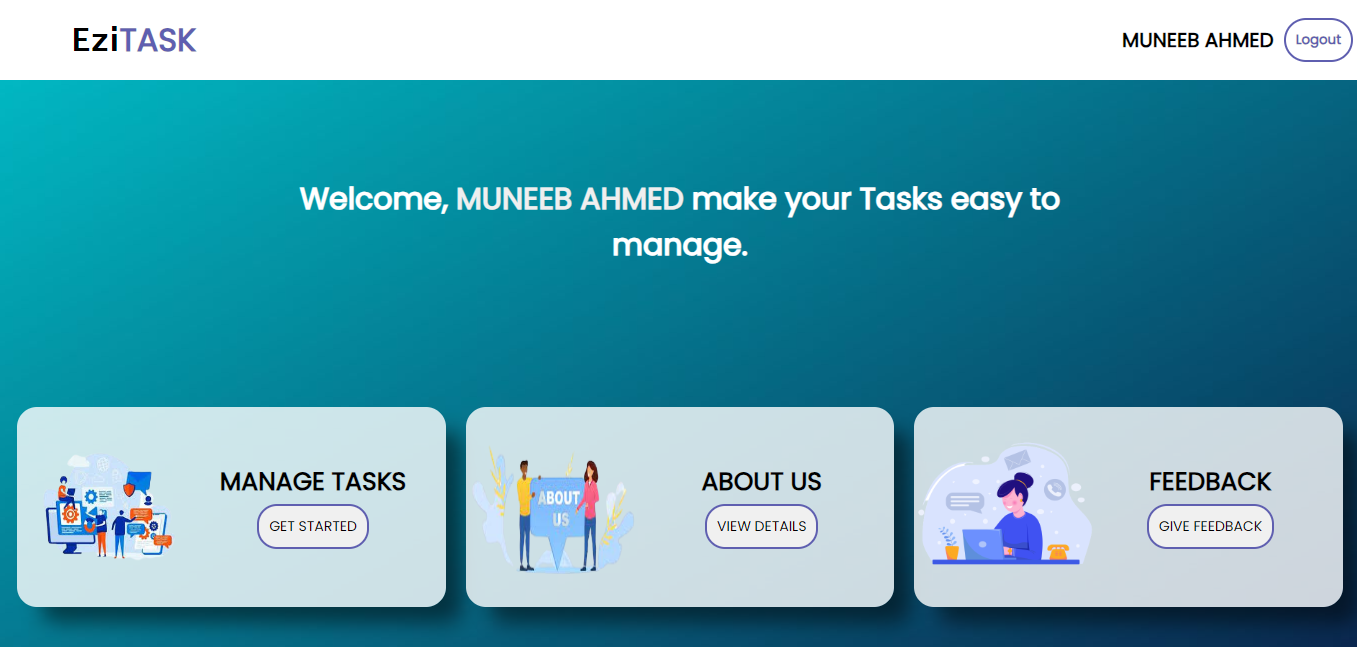
Build multiple user interfaces using laravel blade templates as for   
login/registration  
task CRUD operation

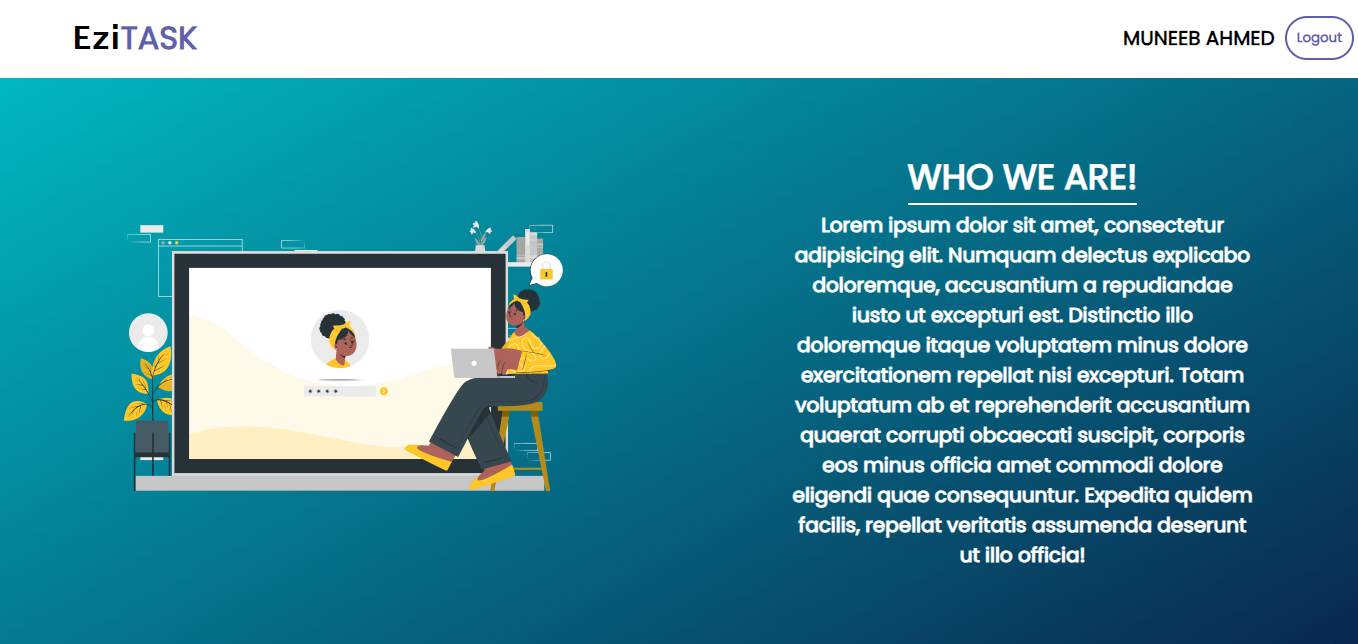
About the website page

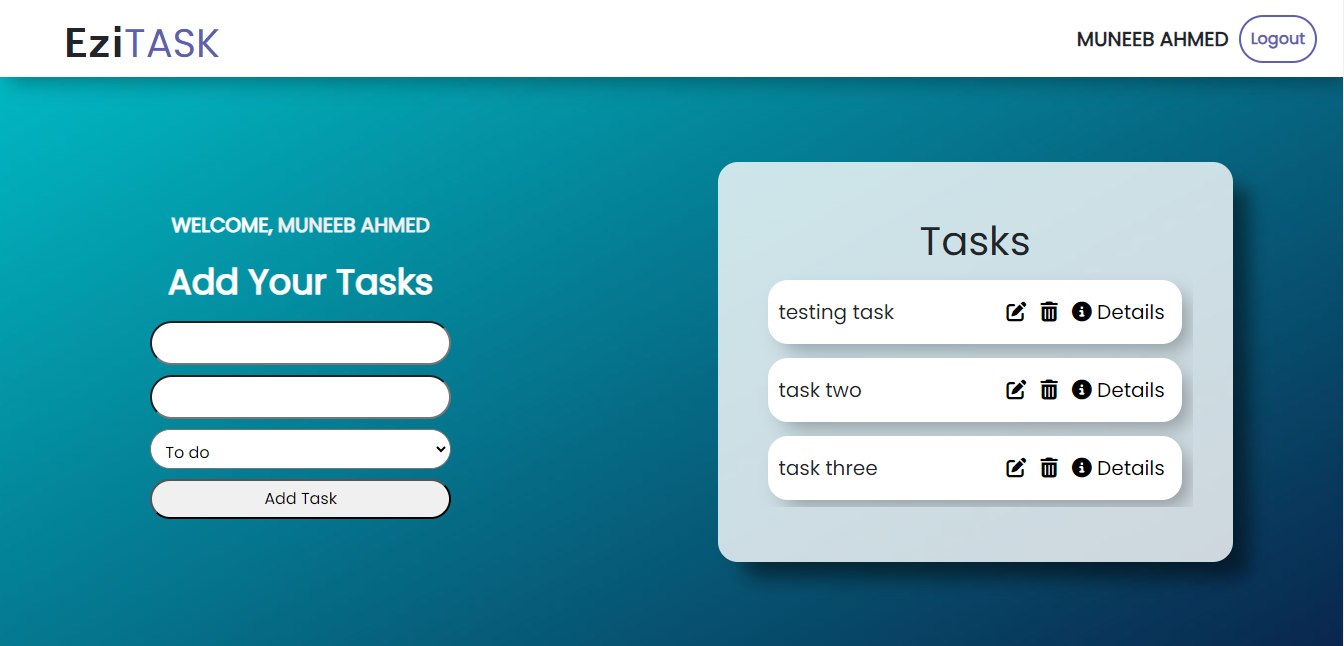
Below are some snapshots of them

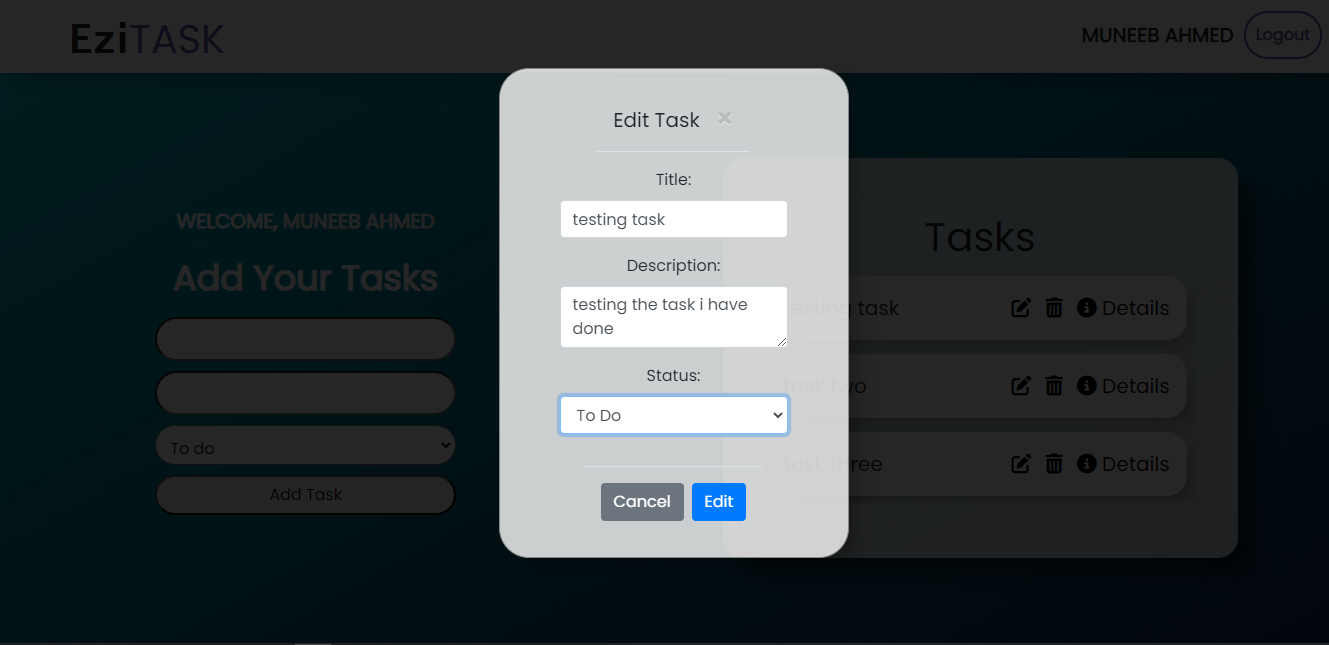


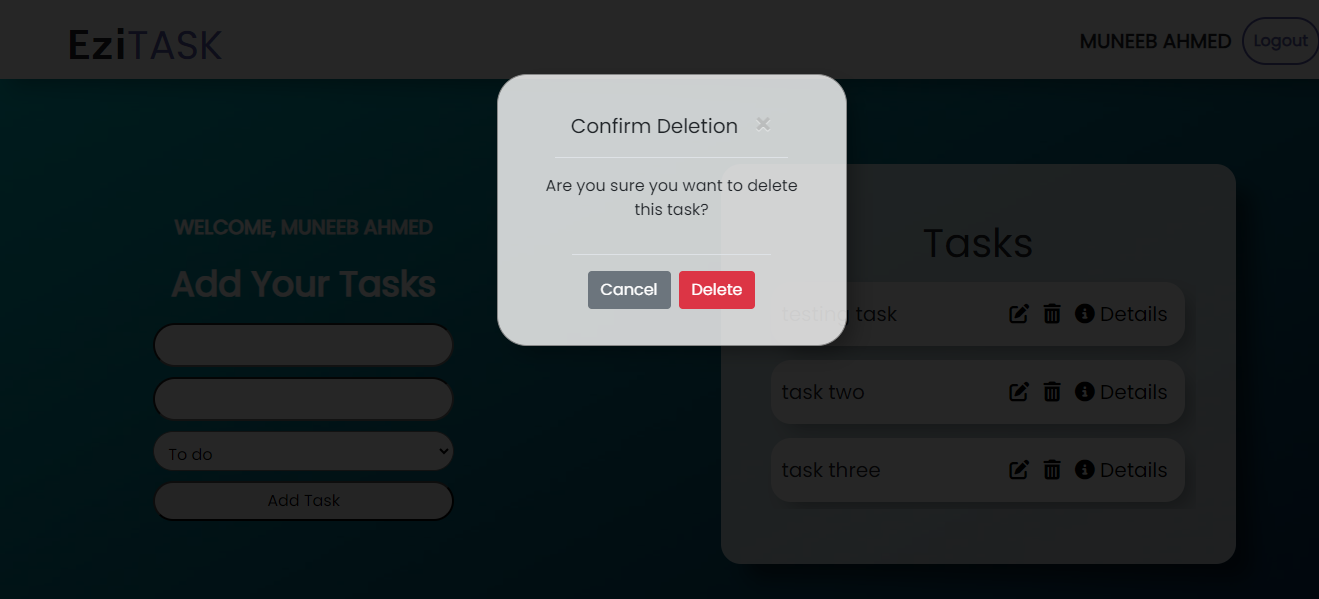


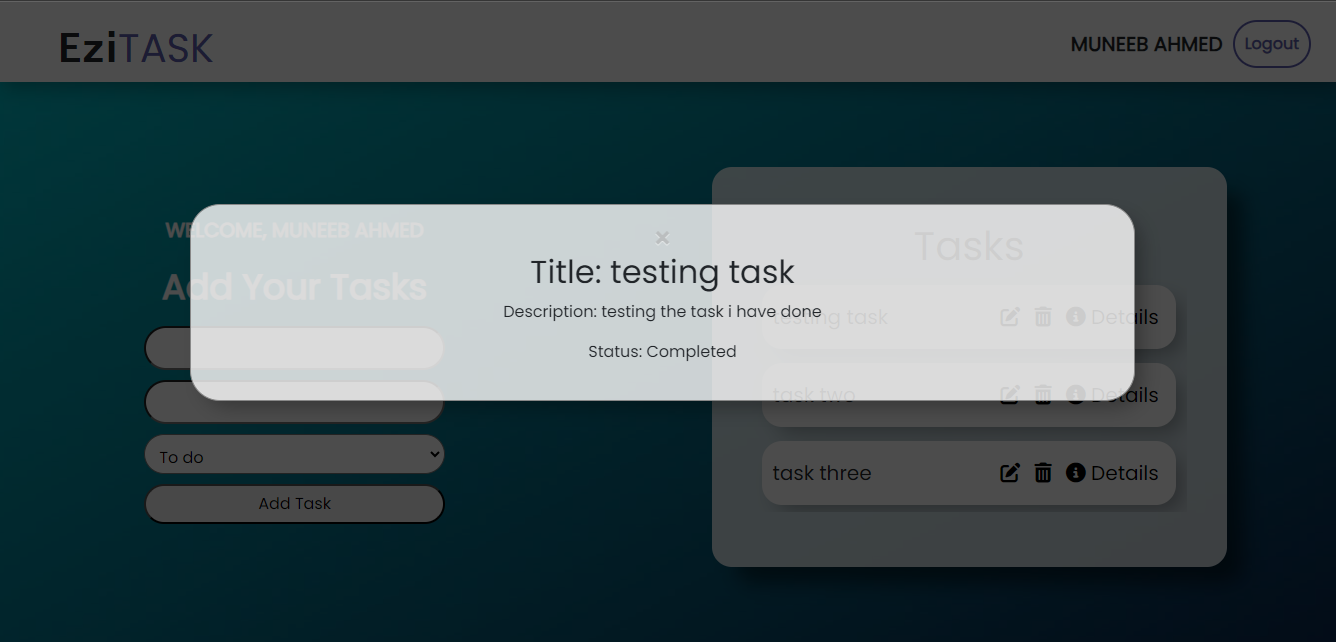








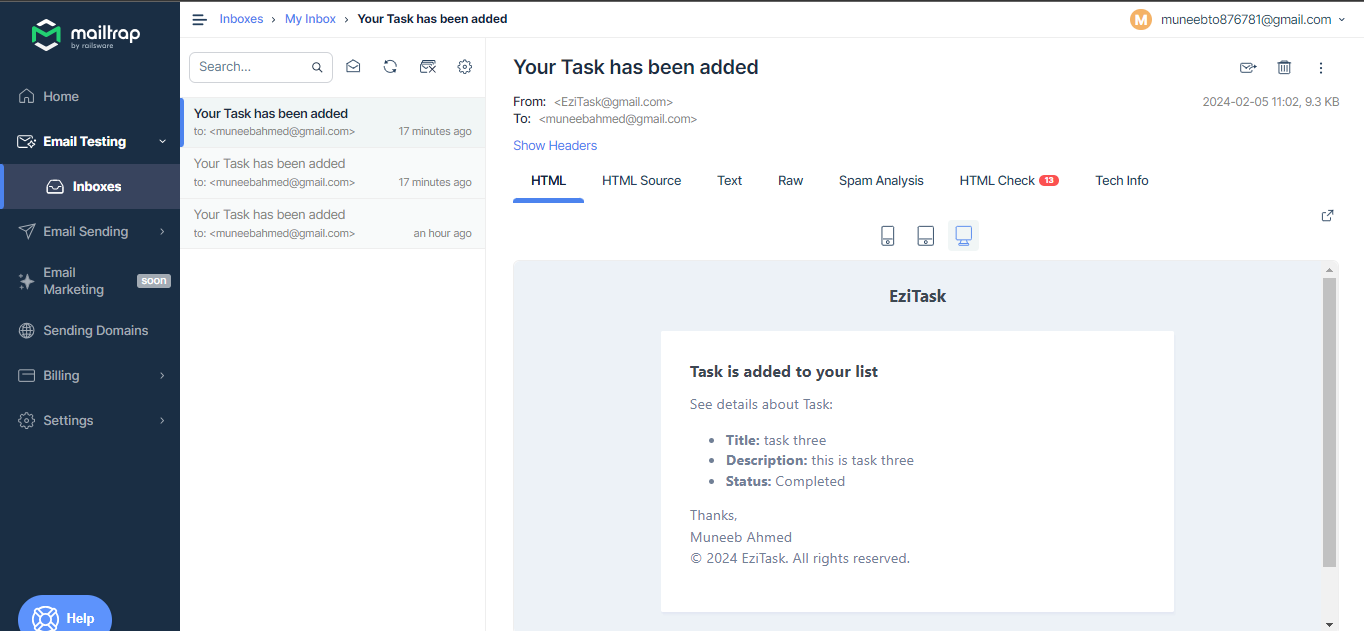




**Web hocks (E-mailing on success events):**

Use web hocks to notify the user about their activity using e-mail procedure user will be notified by e-mail when they perform some activity

Used mail trap to trap email



Write custom email message body with all info about task and send to the authorized user

**REST API for CRUD operation**

Build a REST API for user CRUD operations with all endpoints as

* GET /api/tasks
* POST /api/tasks
* PATCH /api/tasks/{id}
* DELETE /api/tasks/{id}
* Body structure {

“id" : “id”

“title” : “your\_task\_title”

“description” : “your\_task\_description”

“status” : “task\_status”  
}

Using Postman tested the validity of endpoints