# Mini Project for HCL Internship

Tasket

Vishwa R  
39111109

## **Abstract**

Tasket is a task tracking application where users can keep a track of their tasks in an effective manner. I justify the need for such an application because we often forget that we have an array of things to complete by the end of the day. We live in an era of having less attention span, and we don’t find it that easy to remember and complete all the required tasks, be it personal or work-related.

Tasket is an effective solution targeting this specific concern with simplicity and security. The increasing level of mental distractors disrupt our thought process and make us forget things often.

## **Features**

These are the core features, which will be present in the final production-level application.

### Login and Sign-up

1. Secure Authentication using JWT (JSON Web Tokens), an industry-standard RFC 7519 method, for proofing claims between request sender and receiver.
2. Password encryption using BcryptJS
3. Innovative, error-prone frontend form validation.

Dashboard

1. Easy to use Dashboard for seeing and editing the tasks.
2. All the tasks will be listed here, which are private to the authenticated user.
3. CTA (Call To Action) for important functions such as creating a task.
4. An intuitive search option to search for tasks by their title as well as description.

User Profile

1. A logout button to Logout from the application.

|  |
| --- |
| Note |
| *These are the initial set of core functionalities, however, there may be additional positive functionalities in the application.* |

## **Tech-stack used**

A modern and lightweight tech-stack is an essential factor in determining the winning of a product in the market. This project uses up-to-date, relevant technologies to deliver a smooth and efficient user experience.

### Frontend

1. React.js for building stateful UI components.
2. React-Redux for global state management throughout the frontend application.
3. React Router for SPA routing.
4. ChakraUI for styling the entire application using an enforcing design system.

### Backend

#### Server

* ExpressJS A minimalistic lightweight web-application framework for Node.js.
* Mongoose - A simple object modeling for MongoDB.
* BcryptJS - To encrypt the passwords.

#### Database

* MongoDB - A battle-tested NoSQL database.

## **Flow Diagram**

## 