

CSC301

Documentation

Shun Chit Wong, Rahul Uppalapati, David Chen, Yuvraaj Chahil, Peter Ghobrial, Daksh
Malhotra, Manjoy Malhotra

Frontend

/Timer.js:

- It is the frontend screen that provides the user to enter time for pomodoro and ability to time the user. This screen imports “@chakra-ui/react” for organizing the timer page.
- It has 5 buttons, “work” and “break” at the top for changing different times, “setting” for setting different times, “start” and “pause” for starting and pausing the timer.
- This also used useEffect to update the timer execution and useContext for setting different timer cases.
- This screen also used “CountdownAnimation.js”, which imported a countdown circle for the timer display.
- SettingContext is another function that requires a timer which helps setting up all the functions and updating the timer according to the user interaction.
- There is also a specific button class for the timer, that takes in the active button the user hits.
- It also has sound for each button click, timer ticking, and timer time ends.

/Tasks.js:

- Displays the list of tasks the user has set. The user may add and remove tasks, as well as edit them.
- This page implements components AddTask.js and TaskList.js.
- AddTask.js defines how to add a task with its specific name and due date, and a built-in unique ID.
- TaskList.js maps through the list of tasks and displays them, and includes edit/delete buttons for each task, allowing you to remove finished tasks, or edit task names by popping up a modal menu to input a new taskname. When deleteTask or editTask is called, it iterates through the list of tasks, and deletes/updates the task with the given task’s ID.

/Journal.js

- Displays journal entries. The user may add, delete, and edit entries here.
- It has two main components, a folder structure tree and a Slate editor.
- Uses useMemo and useState to update the editor contents in real time.
- Each formatting button uses Slate’s node methods to apply text formatting to user selected text

/LoginPage.js

- Login page with a login form
- Is able to authenticate the user's loginning by checking the database.
- Uses chakra UI components to create the login form.

/SignupPage.js

- Sign up page with a sign up form
- Is able to authenticate the user's information and store the information in the database.

- Uses chakra UI components to create the signup form

/LandingPage.js

- Is the index page of the application
- Has a navbar component that is mobile friendly
- Has an about us section which was built using chakra UI components.

Backend