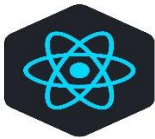








Todo Application

❖ Introduction

This is todo application for adding content, delete contents edit and view existing contents for user. Also, the application includes Login and registration part for separate contents from user to user. Also, the application provides validation and error handling for better user experience.

❖ Technology Stack

Technology	Description
	React used for developing frontend side of application.
	Typescript used for developing frontend part functionalities.
	Material UI used for enhancing frontend for better user experience.
	Ant design used for enhancing frontend for better user experience.
 	Asp.net with C# used for developing backend API for the application.
	Bcrypt used for password-hashing function in login and signup.



Sqlite used as main database of the application.

❖ Components

Login/Sign Up Part

Landing page displaying with component with left side as login and right side as signup. Users can select one side using mouse click. When user click one side, disappear two parts and appearing login or sign-up component. Both login and sign-up parts include validation and error messages.

Todo Home Page

Home page two components add content part for add new tasks and other component for show task according to user account. This page also has validations and error handling messages. Users can add titles up to 55 characters and description maximum length is 300 characters.

If the user wants to edit or delete need to click that task card and edit and delete buttons appear. If click edit button load that card data to input fields and disappearing add button and appearing save and cancel buttons. If user click cancel, clear all input fields.

Top right corner of application shows the username and if user needs to sign out need to click that user icon or username. Then automatically redirects to landing pages. Routes are protected and cannot access the home page without login.

❖ Instructions to Run

First, download both Frontend and Backend Folders. Open Frontend folder using Vscode or any ide and type npm start in terminal to frontend application. Then open **todoapi.sln** file using visual studio and start backend server. Then you can test the application.