**To-Do App Project Report**

**1. Project Overview**

This To-Do app allows users to register, log in, and manage their todos. The frontend is built using React, while the backend relies on AWS-managed services for handling API requests, authentication, and data storage.

**2. Technology Stack**

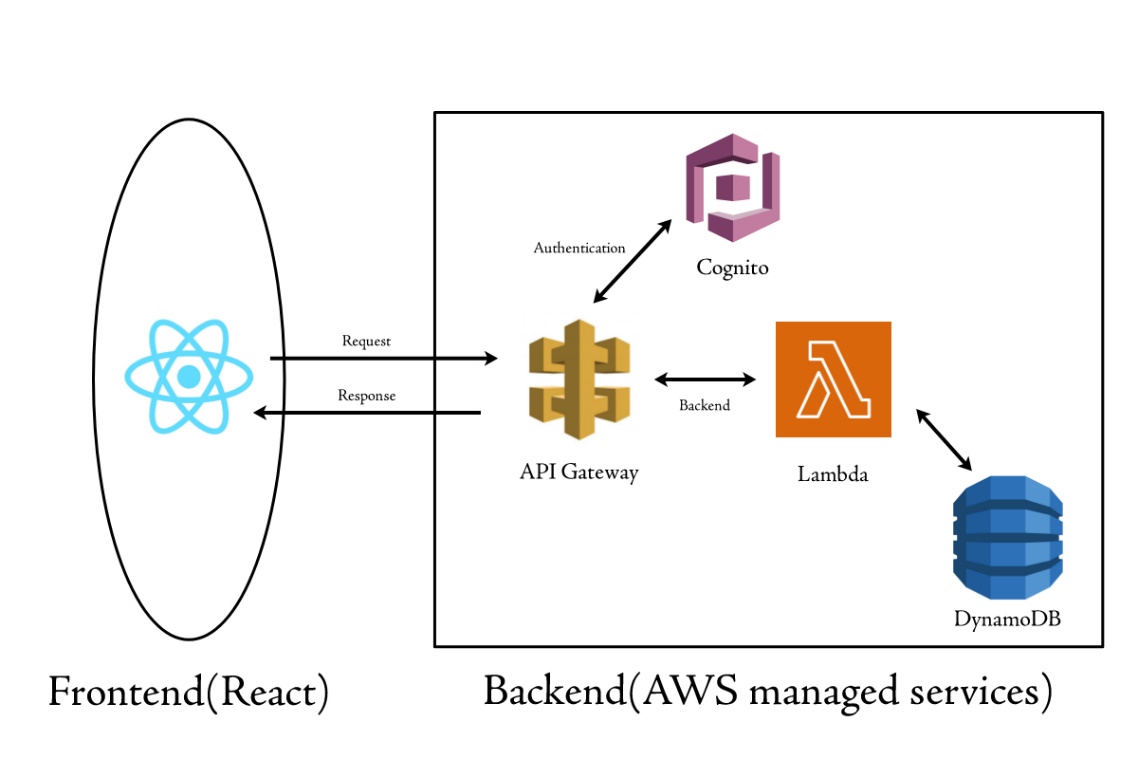
1. Frontend: React

2. Backend: AWS-managed services

* API Gateway: Handles API requests from the frontend.
* AWS Lambda (Python): Executes requests based on API calls.
* DynamoDB: Stores user todos.
* Authentication: Amazon Cognito

User authentication is managed via Cognito User Pools, which is linked to API Gateway. Users log in through the Cognito-hosted login page and are redirected to the ToDo page after successful authentication.

**3. Key Features**

****

**1. User Authentication**

* Users register and log in via Cognito.
* After login, users are redirected to their To Do page.

**2. To Do Management**

* Add: Users can add new tasks.
* View: Users can view their stored tasks.
* Update: Users can modify existing tasks.
* Delete: Users can remove completed or unnecessary tasks.

**3. Backend Processing**

* API Gateway receives requests from the frontend and routes them to Lambda.
* Lambda (Python Functions) handles:
  + GET - Fetch todos.
  + POST - Add a new todo.
  + PUT - Update an existing todo.
  + DELETE - Remove a todo.
* DynamoDB stores all user todos with necessary details.

**4. Project Flow**

1. User opens the app(React frontend).
2. User clicks login, Redirected to Cognito login page.
3. After authentication, Cognito redirects the user back to the app.
4. User accesses To Do page and manages their tasks.
5. Requests are sent via API Gateway to Lambda functions for processing.
6. Lambda interacts with DynamoDB, and the response is sent back to the frontend.