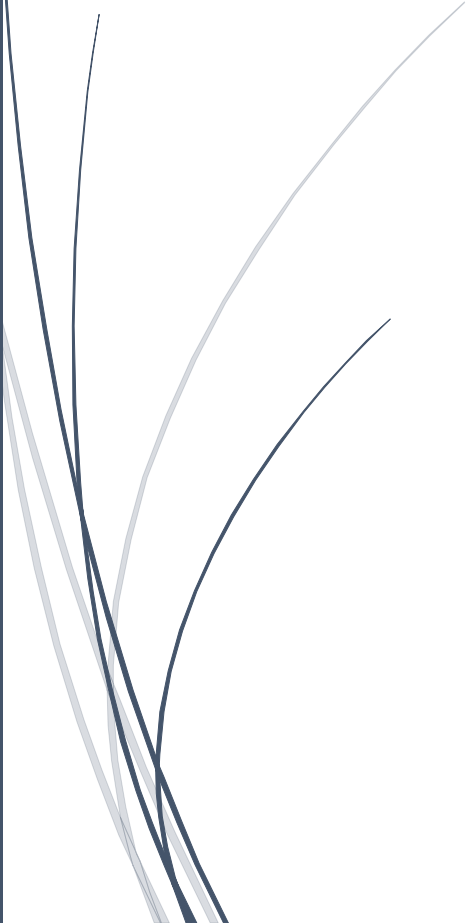


A dark blue vertical bar on the left side of the slide. A blue arrow points to the right from the bar, containing the text 'CSCI3100'.

CSCI3100

High-level Design Document – Little Twitter

Group

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

HUI Rocky	1155172012
CHOW Chun Man	1155109984
CHENG Tsz Kin	1155175870
HO Chun Wai	1155158936
Chan Siu Chung	1155157657

Table of Contents

- 1. Introduction 2
 - 1.1 Project Overview 2
 - 1.2 System Features 2
- 2. System Architecture 4
 - 2.1 Technologies 4
 - 2.2 Architecture Diagram..... 5
 - 2.3 System Components 6

1. Introduction

1.1 Project Overview

Nowadays, the Internet is commonly used in our daily life. With the help of the Internet, we can do a lot of things in an effective way including social aspect. This project is a client-server web-based social media application. The project aims to provide a web platform for people to interact with. People can share their daily life, chat with others, and make friends on this platform. With the use of this platform, people may extend their social circle. The limit of location will no longer exist, but it is hard to solve in face-to-face interaction.

1.2 System Features

There are six major features in our software system.

(1) Login and Registration

System would have a delegated pages for the users to login so that they can navigate to the tailored-made home page for users-specific contents. Besides, there would be a registration page for new users to set up a new account by typing in their own information like username and password.

(2) Users Home Page

Our system would provide a user-specific home page which contains tweets posted by their following users, the tweets they post and the recommended tweets. They can observe the likes, dislikes and some of the comments related to that tweets from the home page. They can tick on the tweet to browse the related tweet page. Besides, there is a search bar on the home page for users to search other users. Moreover, user can browse their own profile page from the home page. If the user is admin, there would be a specific function on the homepage for the admin users to navigate to their dashboard.

(3) Tweet Page

After ticking on the specific tweet, the user would be navigated to the tweet page. Apart from the like and dislike numbers. The users can view all the detailed comments about the tweet on that page, Besides, the user can also reply the comments or retweet from that page.

(4) Users Profile Page

After ticking on the self-profile from the home page, the user would access their own profile page which show all the information about them. They can also edit those information form that page. Besides, users can also browse others' profile after selecting other users on the search bar. However, the information shown is limited and it is not editable.

(5) Private Chat Box

System also allow users to share instant message via the private chat box. Besides, the

chat record would be stored and can be retrieved when the users navigated back to the chat box

2. System Architecture

2.1 Technologies

In this project, we will use MERN stack for our Little Twitter website, which stands for: MongoDB database, Express.js backend framework, React.js frontend library and Node.js server.

MERN stack is one of the most popular tech stacks for developing single page applications.

We use MongoDB Atlas, a NoSQL cloud-based database as our database for storage. By doing so, no local storage database is needed. Moreover, NoSQL database supports flexible schemas which is suitable for rapid application development.

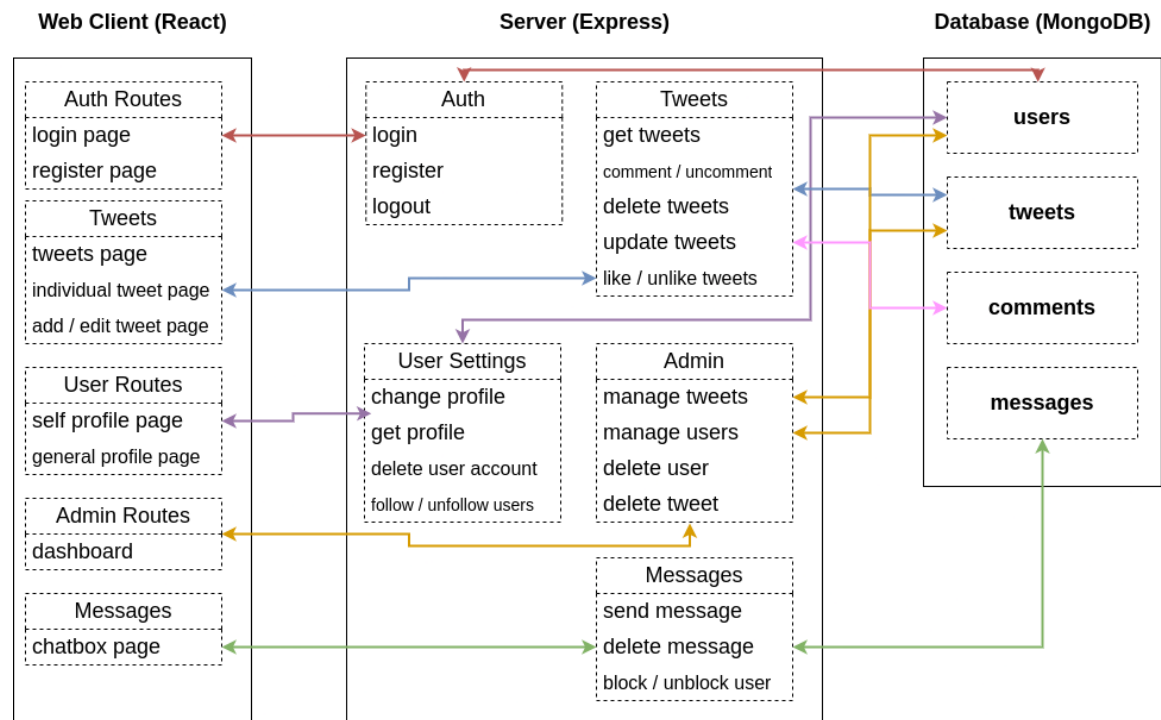
Node.js is a server environment for JavaScript code to run outside of a browser. We chose Node.js as the server because it is a powerful framework supporting many well-developed modules for various purposes.

In this project, we use Express.js module, a minimalist web developing framework to develop our website. It supports functions like route management and handling requests and response. There are also powerful middleware to facilitate our design and implement different functionality. Express.js communicates with frontend using RESTful API which enhances the uniformity of resources management.

We use React.js as our front-end library. It is a powerful library which can manipulate resources using RESTful API. It also provides the functionality to do awesome UI and animations.

The four technologies mentioned above are written in JavaScript so that the software engineers could have a flatter learning curve and the component is perfectly compatible. It increases the efficiency of development.

2.2 Architecture Diagram



(NOTE: this probably needs changes)

2.3 System Components

MERN Stack (MongoDB, Express.js, React.js, Node.js)

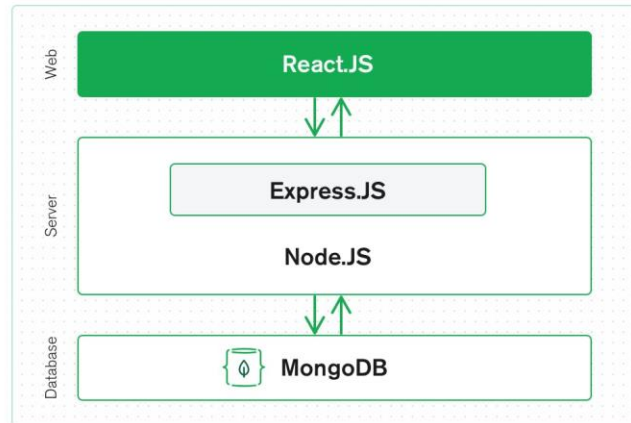


Fig 2.3.1 MERN stack structure

■ Front-End

React.js is the declarative JavaScript framework for creating dynamic client-side applications in HTML. User interface can be built by React.js with simple components that connect with the data on back-end server and rendered as HTML.

■ Back-End

Express.js is running inside a Node.js server. It has useful models for URL routing and handling HTTP requests and responses. By making XML HTTP Requests or GETs or POSTs from React front-end, Express.js can access and update data in MongoDB with MongoDB's Node.js drivers.

■ Database

MongoDB is a document-oriented database. When JSON documents created in React.js front-end, they can be sent to the Express.js server, where the data can be processed and stored directly in MongoDB.