USC Viterbi
School of Engineering

Blog App

EE 547 – FINAL PROJECT REFERENCE DECK

Yuqi Chen
Jian Dong
Bei Ming

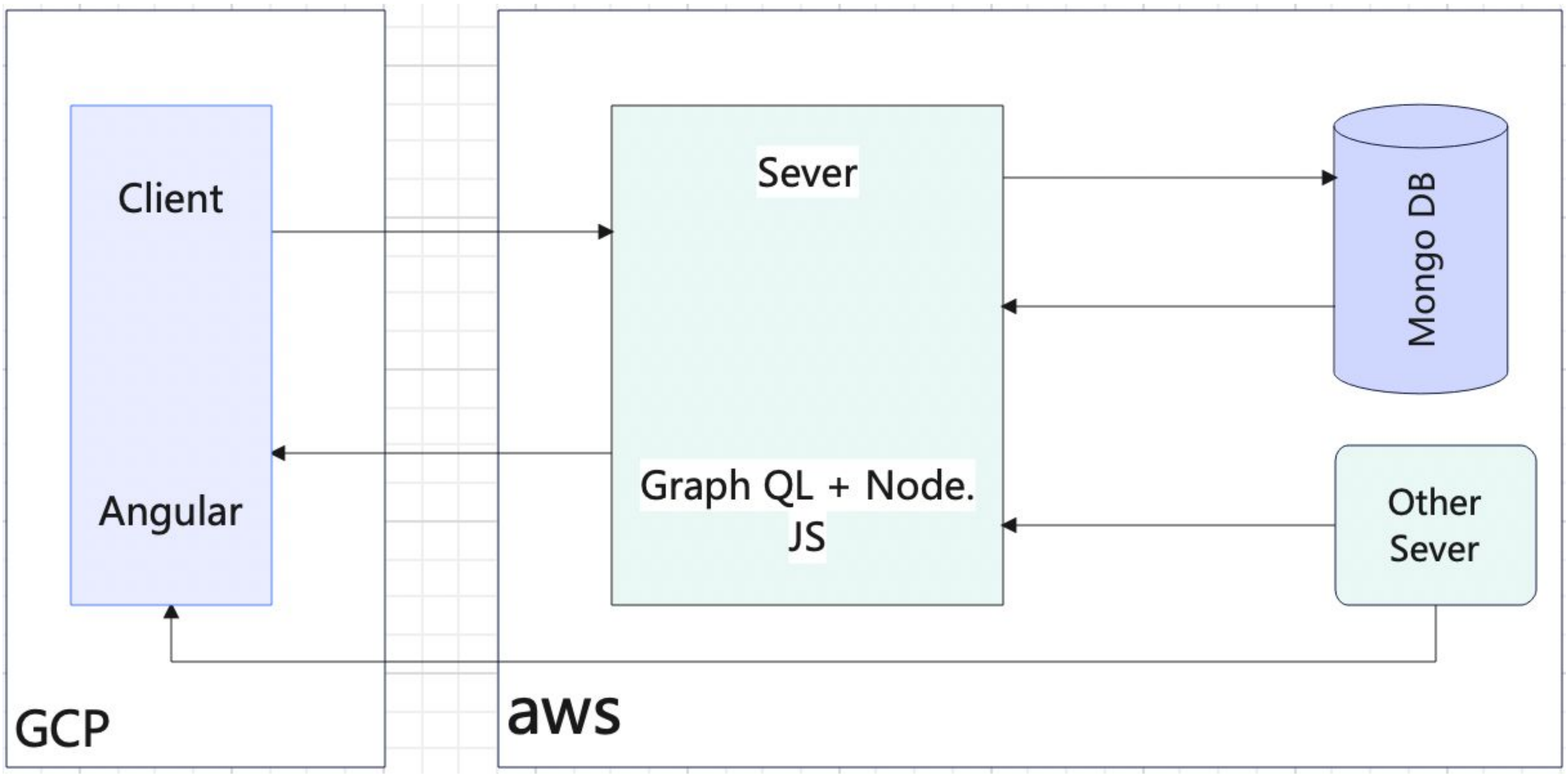
Fall 2022

PROJECT SUMMARY

- Blog platform
- Homepage and Blog page (post, profile, blog)
- Functions including : login, sign-up, blog (post,edit, delete, co-edit), profile
- Other functions: time, address, translate, joke

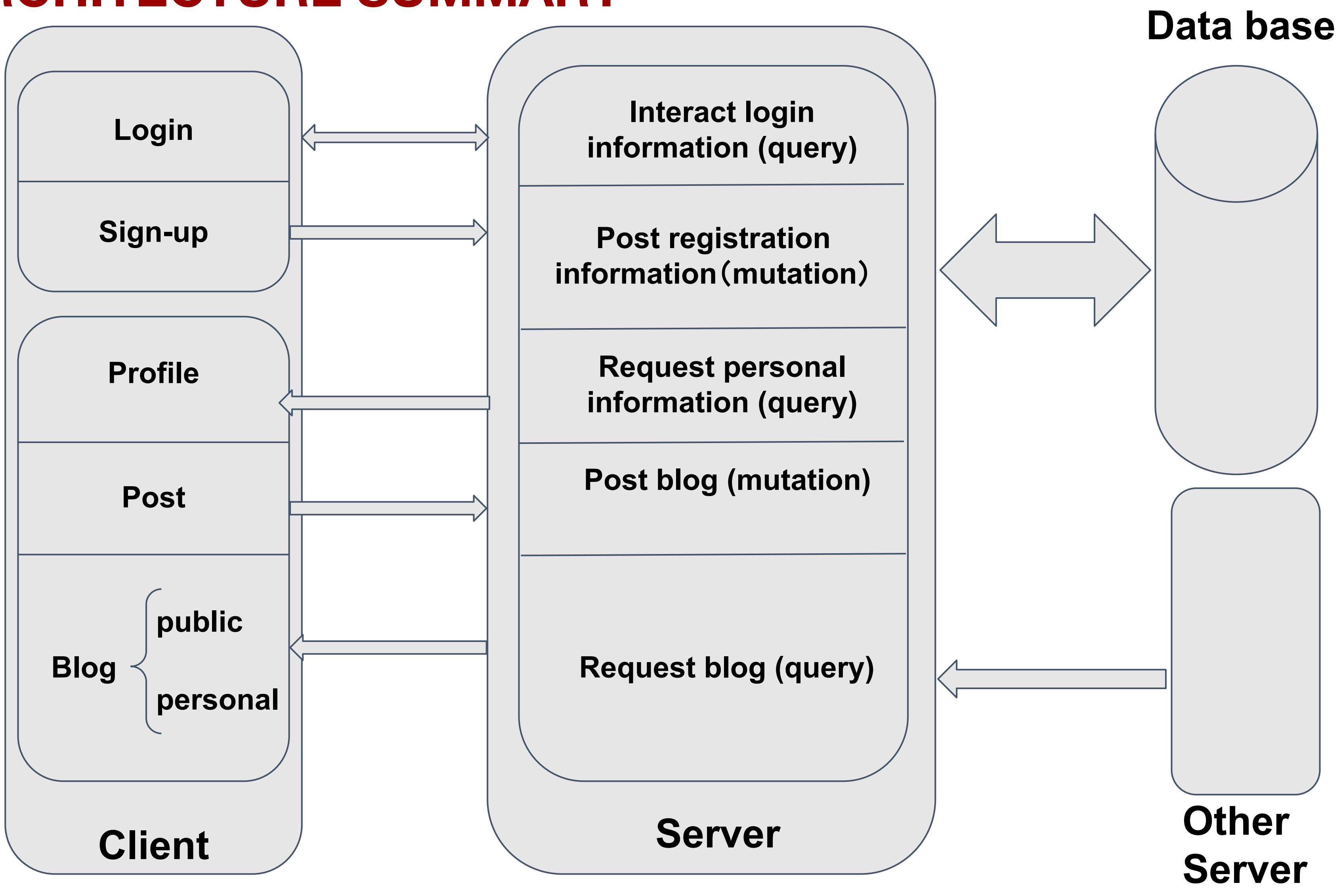
USC Viterbi
School of Engineering

ARCHITECTURE SUMMARY



USC Viterbi School of Engineering

ARCHITECTURE SUMMARY



TIMELINE – RETROSPECTIVE

November 30th: build database structure, design graphql
December 1st: develope query mutation and test
December 2nd: learn the subscription
December 3rd: rebuild graphql on apollo server instead of express-graphql
December 4th-5th: develop subscription
December 6th: test whole backend functionality
December 7th-9th: build frontend page
December 10th-11th: combine frontend and backend
December 12th: finish subscription in frontend and use external api

USC Viterbi
School of Engineering

REFERENCES

- Apollo docs: Subscriptions in Apollo Server
  https://www.apollographql.com/docs/apollo-server/data/subscriptions/
- Apollo docs: API Reference: Apollo Server
  https://www.apollographql.com/docs/apollo-server/api/apollo-server/#includestacktraceinerrorresponses
- NLP translation
  https://rapidapi.com/gofitech/api/nlp-translation
- Angular 13
  https://angular.io/docs
- Apollo-angular
  https://the-guild.dev/graphql/apollo-angular/docs

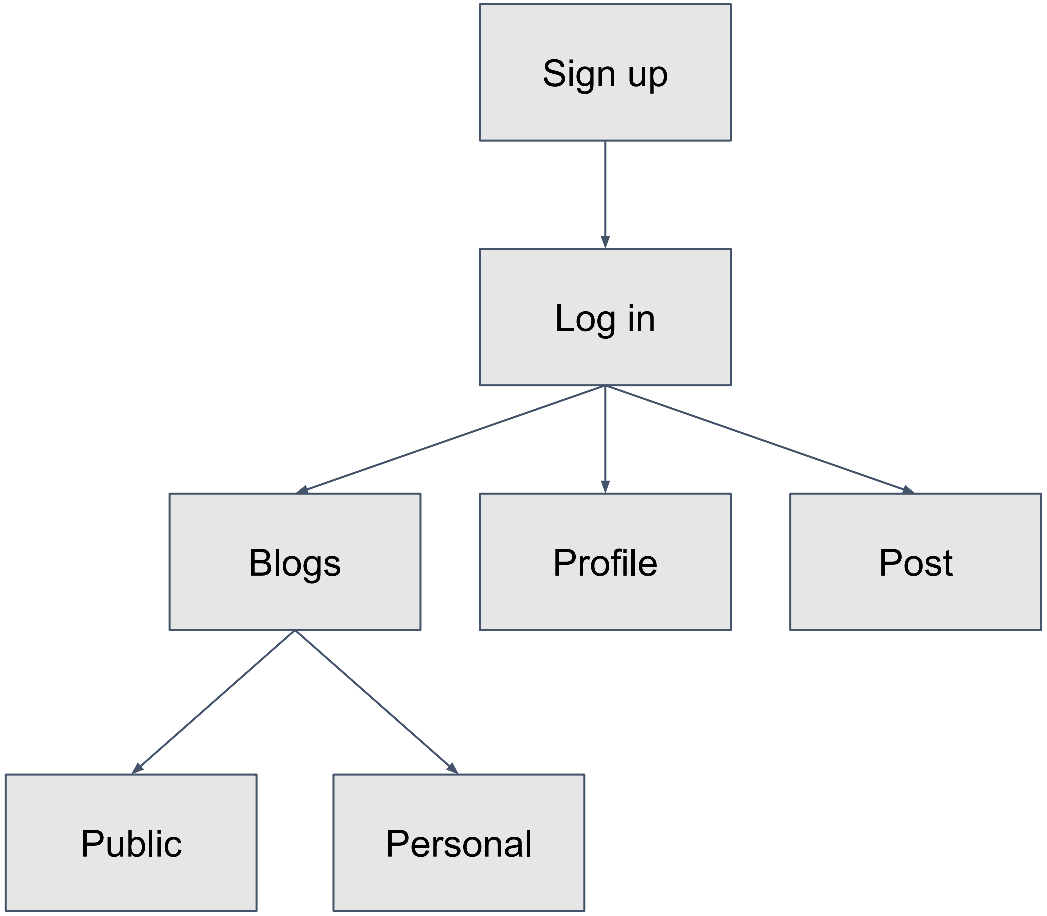
FRONTEND

FRONTEND OVERVIEW

- Framework: Angular
  single-page application by rendering HTML/CSS layout
- Modules: Apollo-angular, HttpClient
  communicate with the server
- Deployed on GCP
  http://blogapp-371504.wl.r.appspot.com/

USC Viterbi
School of Engineering

USER FLOW DIAGRAM

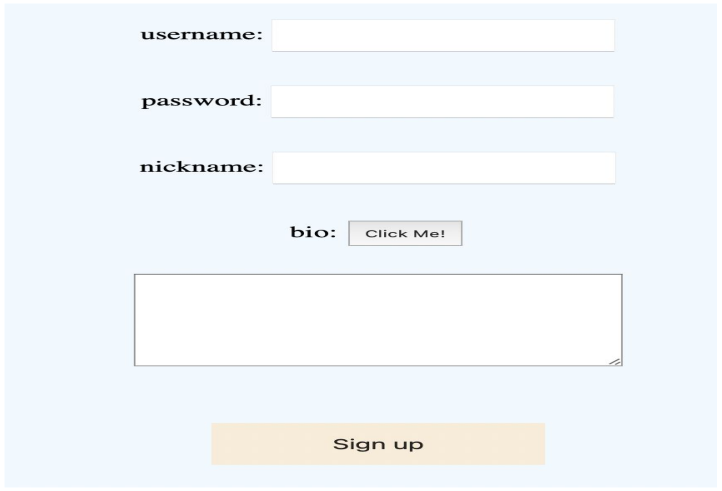


USER ROLES

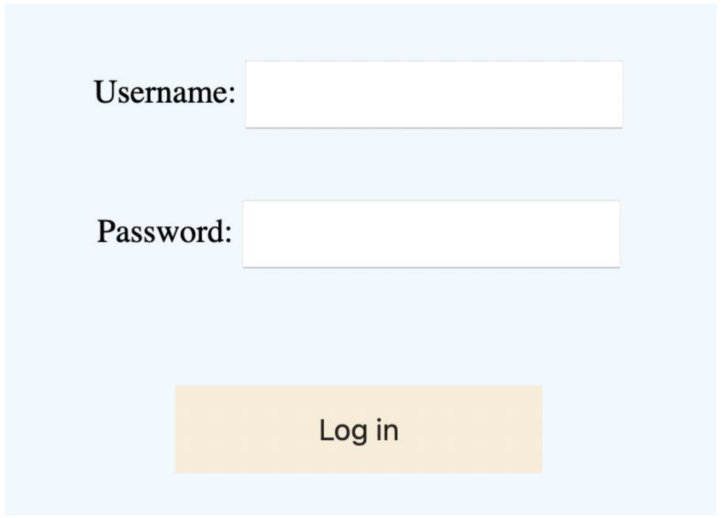
- Input necessary information to be written into backend database.
  eg. input account information on Signup page, input blog content whenever user is making a post.

- Click buttons to trigger communication with the server.
  eg. click “Log in” button to ask server for identity authentication, click “translate” checkbox to translate English to Spanish.

PAGE: Sign up & Log in

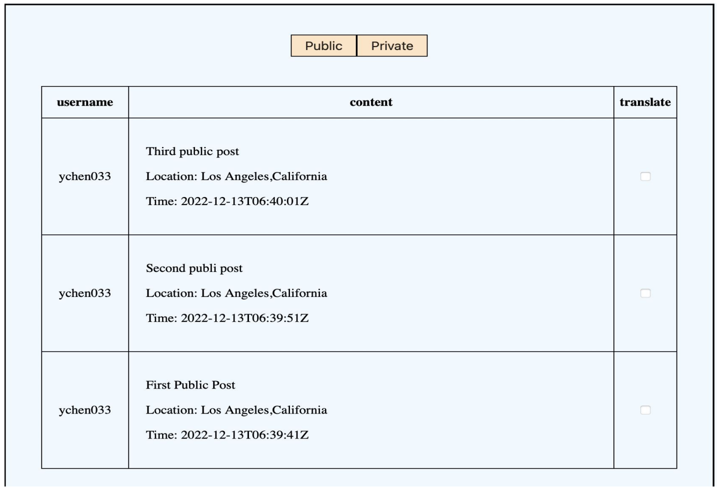


- User inputs account information
- Auto generate bio by clicking “Click Me”
- Check if the username exists in the DB
- Unique username triggers writing into the DB



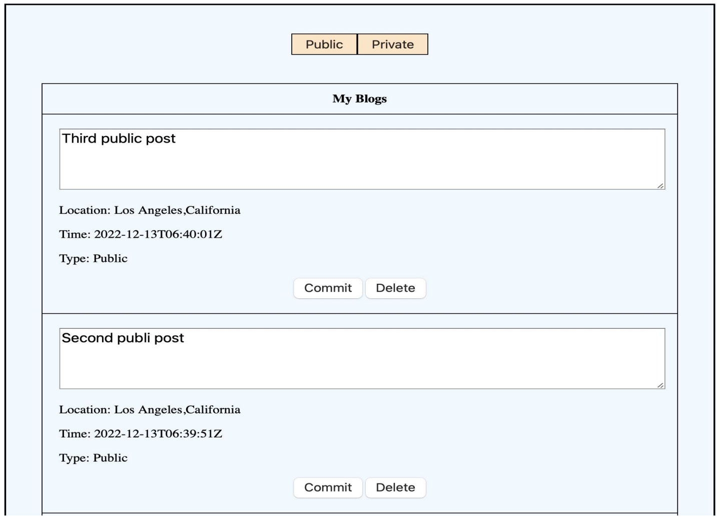
- User inputs username and password
- Check if username exists in the DB
- Check if password matches username
- User successfully log into the app

PAGE: Public & Private blogs



username	content	translate
ychen033	Third public post Location: Los Angeles,California Time: 2022-12-13T06:40:01Z	☐
ychen033	Second publi post Location: Los Angeles,California Time: 2022-12-13T06:39:51Z	☐
ychen033	First Public Post Location: Los Angeles,California Time: 2022-12-13T06:39:41Z	☐

- View all public blogs of all users
- Translate the content to Spanish after clicking the checkbox



My Blogs

Third public post

Location: Los Angeles,California

Time: 2022-12-13T06:40:01Z

Type: Public

Second publi post

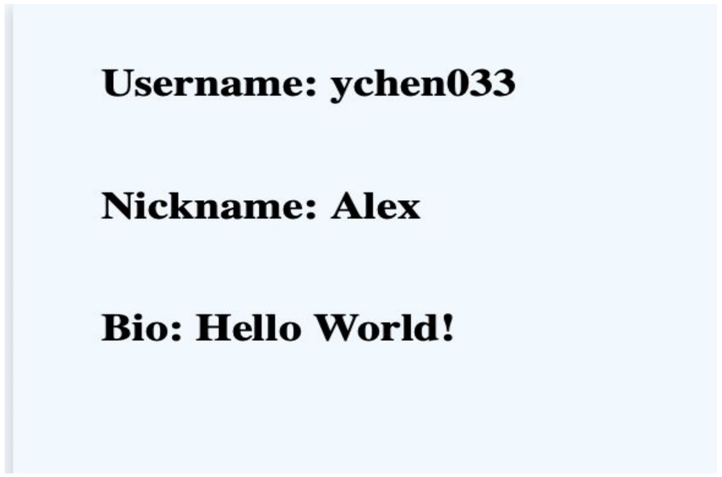
Location: Los Angeles,California

Time: 2022-12-13T06:39:51Z

Type: Public

- View all blogs posted by current user, and info of each blog
- Commit changes on blog content
- Delete blogs

PAGE: Profile & Post

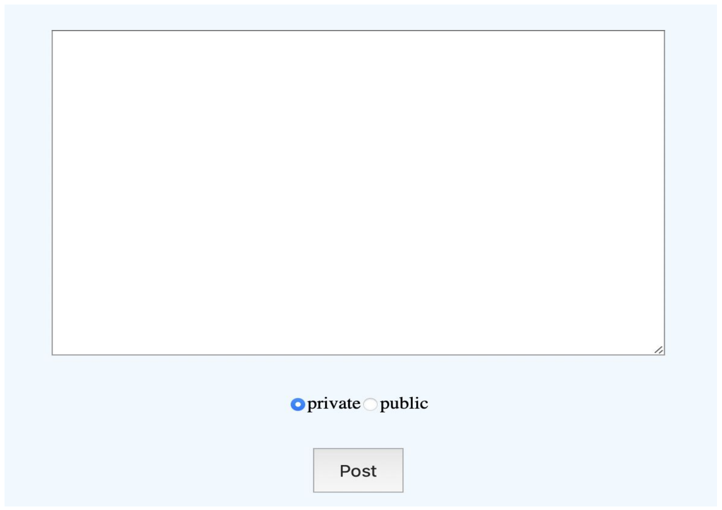


Username: ychen033

Nickname: Alex

Bio: Hello World!

- View personal information



- Input blog content
- Auto generate current address info
- Choose either private or public
- Edit post by multiple people simultaneously

USC Viterbi
School of Engineering

BACKEND

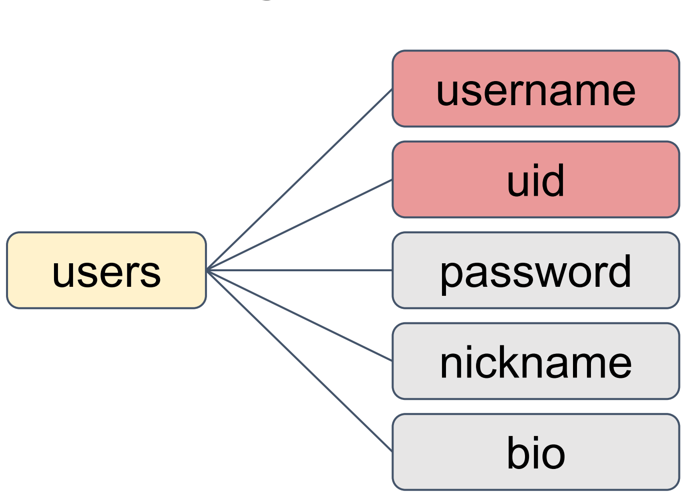
BACKEND OVERVIEW

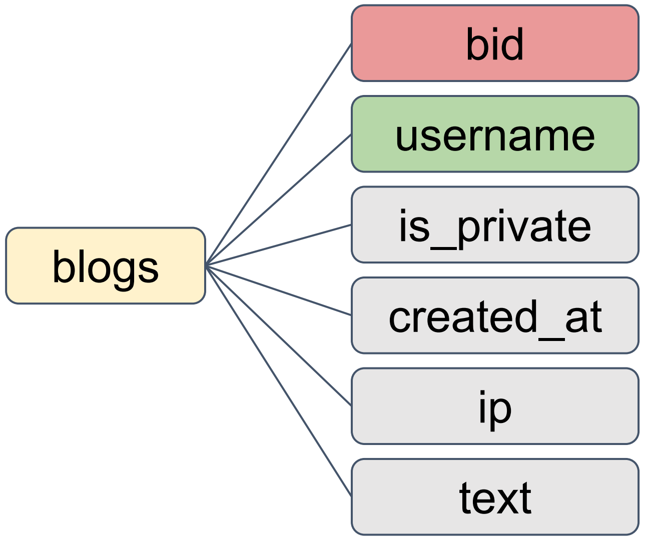
- Framework: Nodejs, Graphql
- Modules: Apollo-server, websocket, axios
- Deployed on AWS http://ec2-34-213-46-21.us-west-2.compute.amazonaws.com

USC Viterbi School of Engineering

BACKEND SUMMARY

- MongoDB :



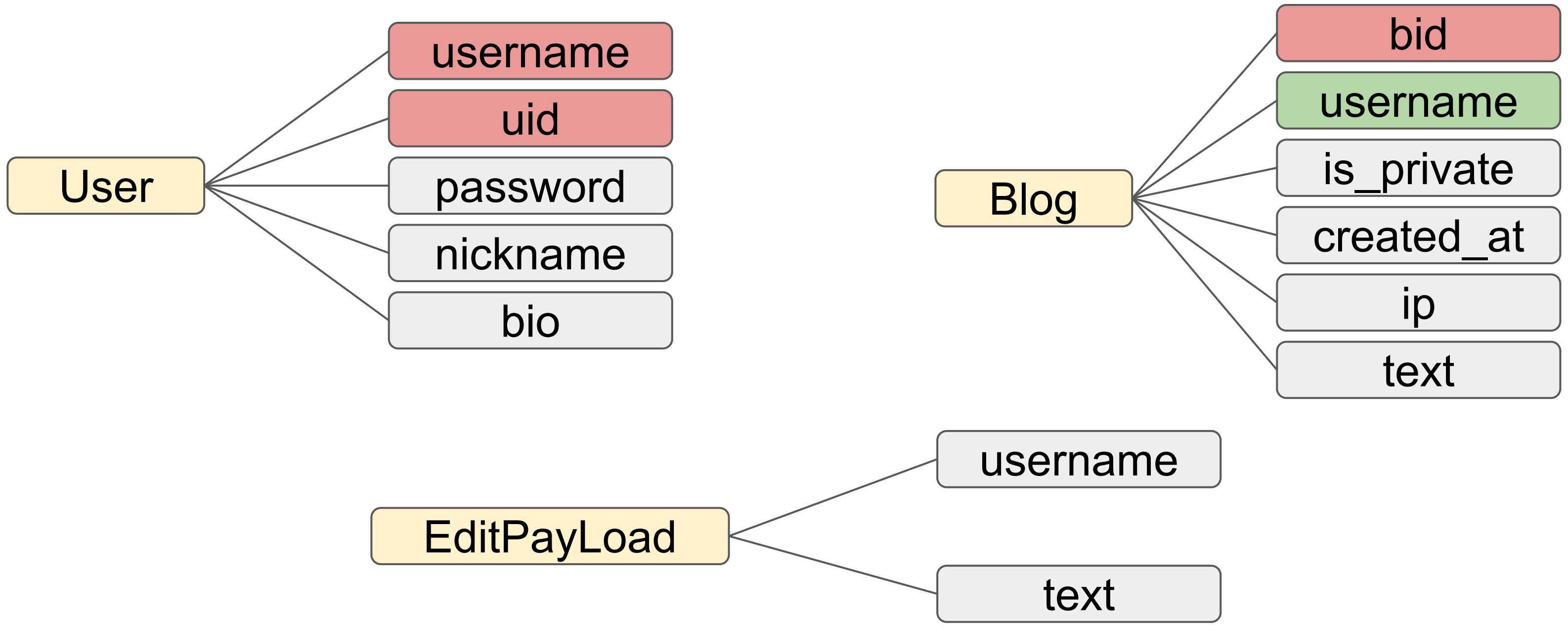


users: username and uid are unique

blogs: username is foreign key and bid is unique

BACKEND SUMMARY

- Graphql



BACKEND SUMMARY

- Graphql :

query	user	**input:** username **return:** the information of user with this username
	login	**input:** username, password **return:** success or error information
	blog	**input:** bid **return:** the information of this bid’s blog
	blogs	**input:** username, is_private, sort **return:** all the public(is_private=false)/personal(is_private=true) blog in time ascending(sort=”ascend”)/descending(sort=“descend”) order
	translate	**input:** English text **return:** Spanish text
mutation	userCreate	**input:** username, password, nickname, bio **return:** success or error information
	blogCreate	**input:** username, is_private, text, ip **return:** this blogs’ information
	blogEdit	**input:** username, text **return:** text
	blogDelete	**input:** bid **return:** success(true), fail(false)
	blogUpdate	**input:** bid, text **return:** this blogs’ information
subscription	blogEdit	**input:** username **return:** username, text when event is triggered

USC Viterbi
School of Engineering

API

API SUMMARY

- NLP Translation

  Method: GET

  End points:

  https://nlp-translation.p.rapidapi.com/v1/translate

- Official Joke API

  Method: GET

  End points:

  https://official-joke-api.appspot.com/jokes/random

- IP info

  Method: GET

  End points:

  https://ipinfo.io/json?token=c5a96995ca9e33

Outside Data source / API

- NLP Translation

  End points: https://nlp-translation.p.rapidapi.com/v1/translate

  Method: GET

  Params: {text: 'Hello, world!!', to: 'es', from: 'en'}

  Response: {

    status: 200, from: 'en', to: 'es',

    original_text: 'hello', translated_text: { es: 'Hola' },

    translated_characters: 5

  }

USC Viterbi
School of Engineering

IN CONCLUSION…

OUTCOMES AND RESULTS

- Successfully built backend using graphql, node.js, websocket, apollo server and complete all the backend functions
- Successfully built frontend using angular, apollo angular, websocket, rendered the right content on each page and realized proposed functions
- Successfully deployed backend on AWS and frontend on GCP