

Team#: V1

Term Project: Postify

April 3rd, 2022

Team Members	2
What does the application do?	3
Microservice Architecture	4
General overview of the application	5
Comment section	6
List of URLs	6
ERD.jpg	8
SQL dump	9
Link to documentation page.	11
Link to OpenAPI document (Swagger)	11

Todo for the comment section:

- ☒ ~~Provide 8 sample test uris/urls and parameters to test all of your API endpoints.~~
- ☒ ~~Url of the client app~~
- ☒ ~~API admin credentials, the admin login page (domain.com/...)~~
- ☒ ~~The url of the server app (which provides API end points)~~
- ☒ ~~Git url~~
- ☒ ~~Front-end folder (only the files relevant to this application)~~
- ☒ ~~Back-end folder (only include js files that show your server logic)~~
 - ☒ ~~No node_modules, .git, .vscode~~
- ☒ ~~DB.sql file in the root of your git repo containing the SQL dump of your DB~~
 - ☒ ~~Table structure~~
- ☒ ~~ERD.jpg~~

1. Team Members

Name	Jordan Godau
Responsibility	Front-end, back-end
Contribution	Client state-management configuration Client router configuration Client endpoint configuration Client-side response handling General Front-end development Microarchitecture design Code architecture Database Design (ERD) Minor Server bug fixes

Name	Giwoun Bae
Responsibility	Documentation, Front-end
Contribution	Creating front-end pages Client-side text storing and translation Client-side general front-end development Client-side front-end design OpenAPI final report

Name	Lingxuan(Charlie) Wei
Responsibility	Back-end, database
Contribution	Database creation Database design Server hosting SQL programming General client-side front-end development General server-side scripting Server-side request handling Implementing server, endpoints OpenAPI documentation Figma design

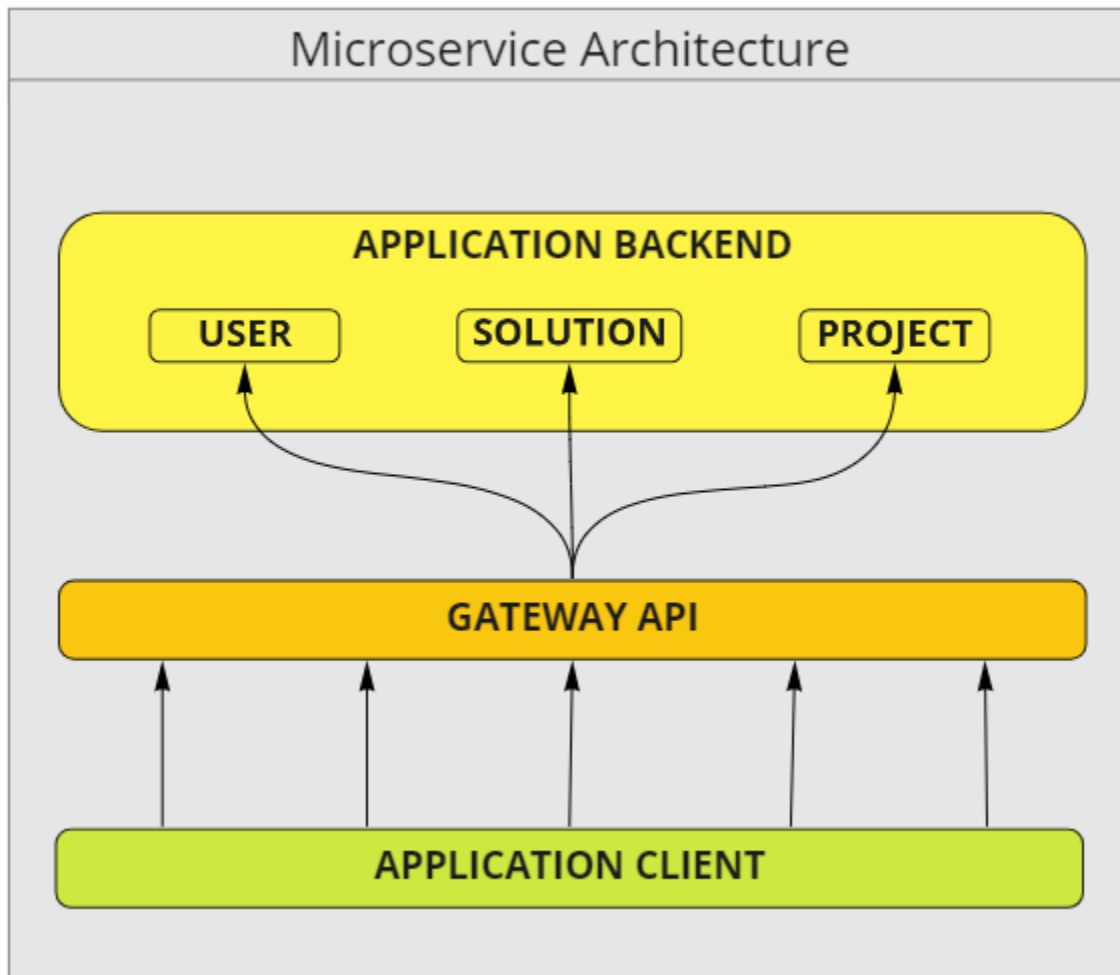
	Hosting server
--	----------------

Name	Liam Arabsky
Responsibility	Back-end, OpenAPI documentation
Contribution	JWT authentication Password Hashing Server-side user registration Server-side user login OpenAPI documentation Hosting client Figma design

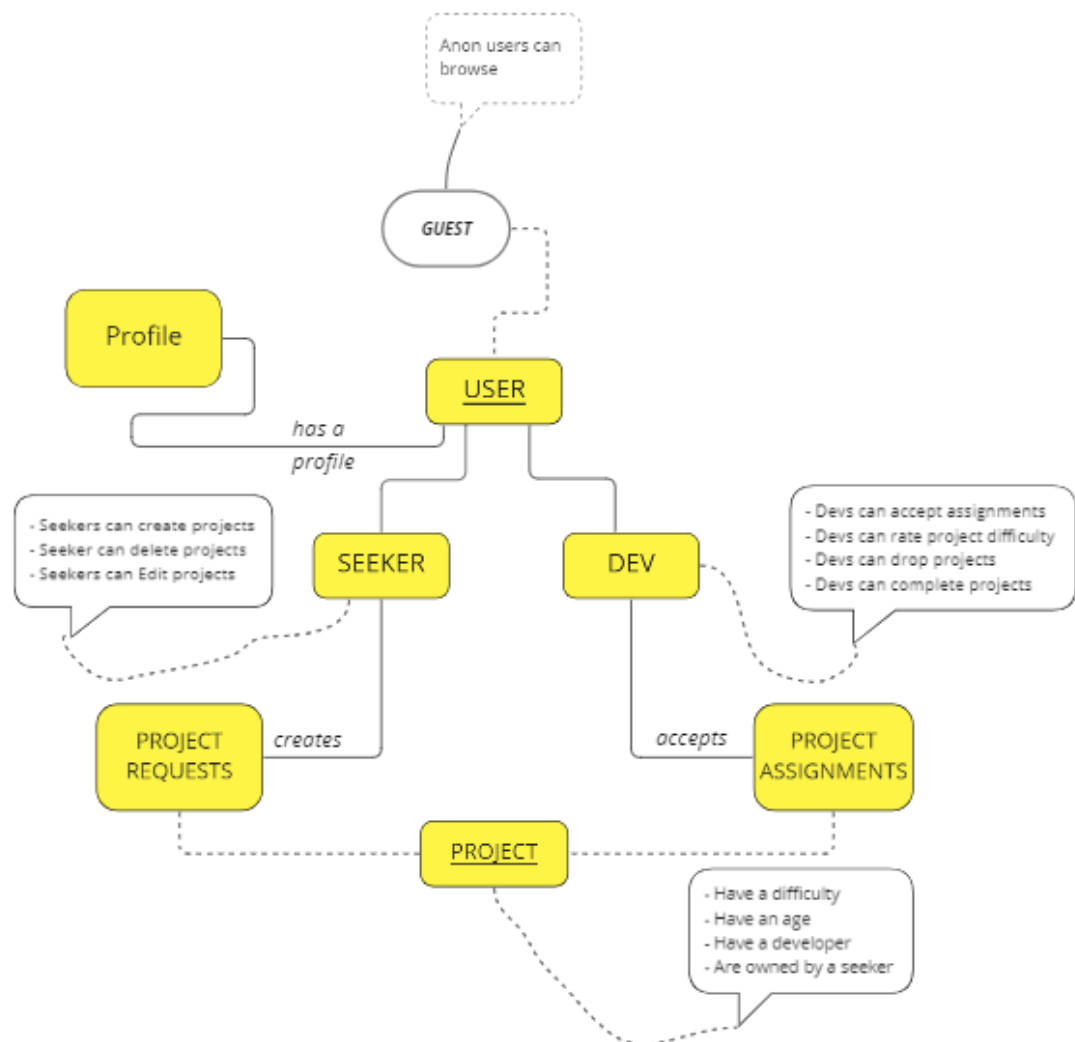
2. What does the application do?

It is a forum based web app that allows users to post a project and other users can post the project's solution.

a. Microservice Architecture



b. General overview of the application



3. Comment section

a. List of URLs

	API endpoints
Endpoint Root	/4537/termproject/api/v1/
GET user (GET)	/4537/termproject/api/v1/user
https://postify.charliewei.com/4537/termproject/api/v1/user	
GET users (GET)	/4537/termproject/api/v1/users
https://postify.charliewei.com/4537/termproject/api/v1/users	
Create user (POST)	/4537/termproject/api/v1/user
https://postify.charliewei.com/4537/termproject/api/v1/user	
Login user (POST)	/4537/termproject/api/v1/login
https://postify.charliewei.com/4537/termproject/api/v1/login	
GET project (GET)	/4537/termproject/api/v1/project
https://postify.charliewei.com/4537/termproject/api/v1/project	
GET all projects (GET)	/4537/termproject/api/v1/projects
https://postify.charliewei.com/4537/termproject/api/v1/projects	
Create project (POST)	/4537/termproject/api/v1/project
https://postify.charliewei.com/4537/termproject/api/v1/project	
Update project (PUT)	/4537/termproject/api/v1/project
https://postify.charliewei.com/4537/termproject/api/v1/project	
Delete project (DELETE)	/4537/termproject/api/v1/project
https://postify.charliewei.com/4537/termproject/api/v1/project	
GET solution (GET)	/4537/termproject/api/v1/project/solution
https://postify.charliewei.com/4537/termproject/api/v1/project/solution	

GET all solutions (GET)	/4537/termproject/api/v1/project/solutions
https://postify.charliewei.com/4537/termproject/api/v1/project/solutions	
Create solution (POST)	/4537/termproject/api/v1/project/solution
https://postify.charliewei.com/4537/termproject/api/v1project/solution	
Update solution (PUT)	/4537/termproject/api/v1/project/solution
https://postify.charliewei.com/4537/termproject/api/v1/project/solution	
Delete solution (DELETE)	/4537/termproject/api/v1/project/solution
https://postify.charliewei.com/4537/termproject/api/v1/project/solution	
Admin stats (GET)	/4537/termproject/api/v1/adminDashboard
https://postify.charliewei.com/4537/termproject/api/v1/adminDashboard	

b. Url of the client app

<https://postify.charliewei.com/4537/termproject/api/v1>

c. API admin credentials, the admin login page (domain.com/...)

Login name: charlie2

Password: charlie2

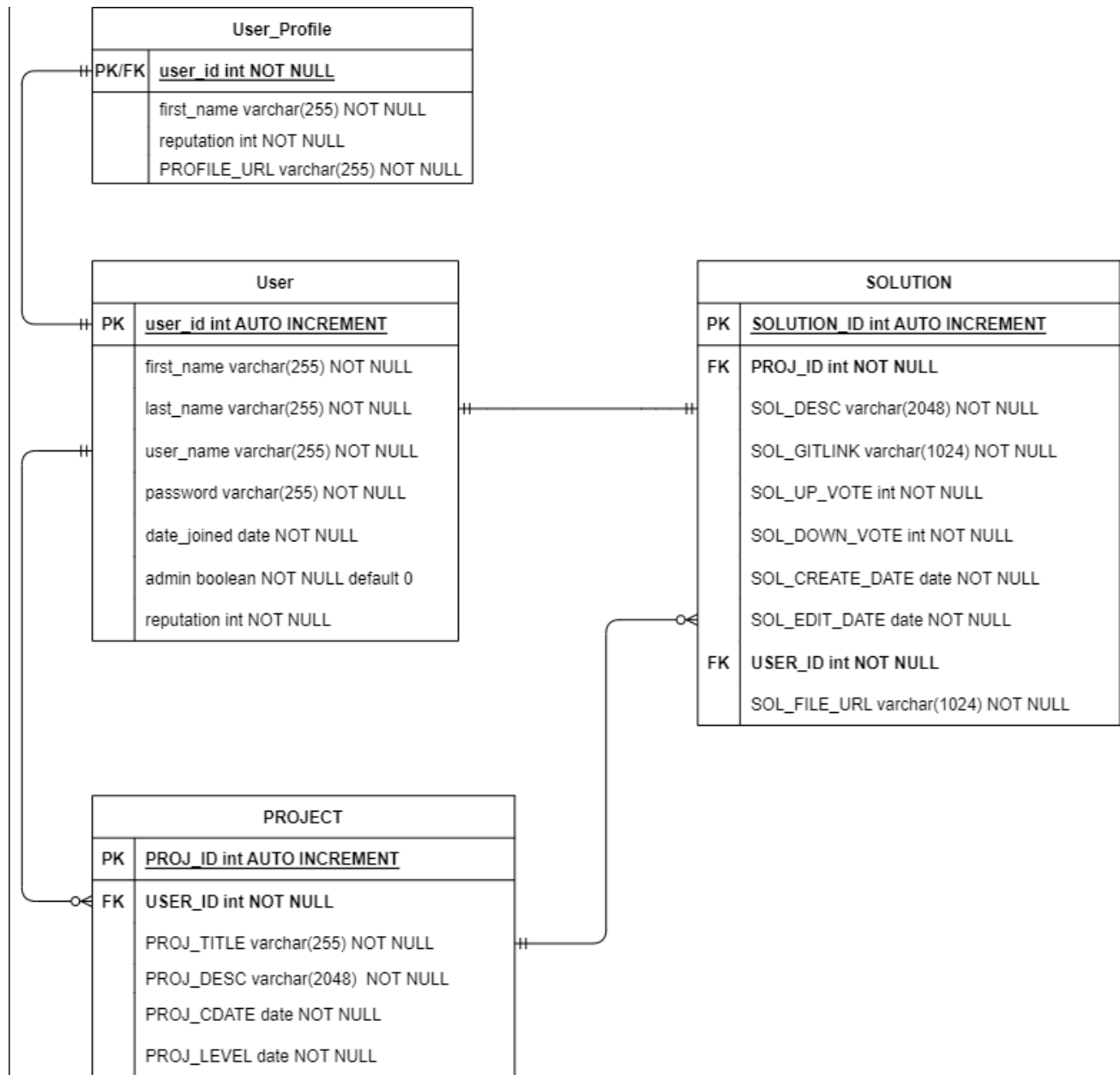
d. The url of the server app (which provides API endpoints)

<https://postify.charliewei.com/4537/termproject/api/v1>

e. Git url

<https://github.com/liamarabsky/postify/tree/dev>

f. ERD.jpg



g. SQL dump

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Apr 04, 2022 at 05:47 AM
-- Server version: 10.4.22-MariaDB
-- PHP Version: 8.1.2

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `postify`
--

--
-- Table structure for table `project`
--

CREATE TABLE `project` (
  `projID` int(11) NOT NULL,
  `userID` int(11) NOT NULL,
  `projName` varchar(255) NOT NULL,
  `projDesc` varchar(255) NOT NULL,
  `projCreationDate` date NOT NULL,
  `projLevel` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `project`
--

INSERT INTO `project` (`projID`, `userID`, `projName`, `projDesc`, `projCreationDate`, `projLevel`) VALUES
(1, 1, 'Chat Assignment in C', 'This project requires a fully implemented chat program written in C', '2022-04-02', 10),
(2, 2, 'Chat Assignment in C#', 'This project requires a fully implemented chat program written in C#', '2022-04-02', 8),
(3, 2, 'An amazing project', 'A description of my amazing project', '2022-04-02', 5),
(4, 2, 'An amazing project', 'A description of my amazing project', '2022-04-02', 5),
(5, 1, 'afdafdsfadsfdas', '2', '2022-04-03', 0),
(6, 1, 'afdafdsfadsfdas', '2', '2022-04-03', 0),
(7, 2, '12412sfafsfsf', 'fsfssd431212421421', '2022-04-03', 4),
(8, 5, 'asfsadfafasfasdfas', 'asdfasfasdfdsafasdfasdfasf12321312321321321', '2022-04-03', 4);

--
-- Table structure for table `solution`
--

CREATE TABLE `solution` (
  `solutionID` int(11) NOT NULL,
  `projID` int(11) NOT NULL,
  `userID` int(11) NOT NULL,
  `solDesc` varchar(255) NOT NULL,
  `solGitLink` varchar(255) NOT NULL,
  `solFileURL` varchar(255) NOT NULL,
  `solUpVote` int(11) NOT NULL,
  `solDownVote` int(11) NOT NULL,
  `solCreateDate` date NOT NULL,
  `solEditDate` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `solution`
--
```

```

INSERT INTO `project` (`projID`, `userID`, `projName`, `projDesc`, `projCreateDate`, `projLevel`) VALUES
(1, 1, 'Chat Assignment in C', 'This project requires a fully implemented chat program written in C', '2022-04-02', 10),
(2, 2, 'Chat Assignment in C#', 'This project requires a fully implemented chat program written in C#', '2022-04-02', 8),
(3, 2, 'An amazing project', 'A description of my amazing project', '2022-04-02', 5),
(4, 2, 'An amazing project', 'A description of my amazing project', '2022-04-02', 5),
(5, 1, 'afdafdsfadsfdfas', '2', '2022-04-03', 0),
(6, 1, 'afdafdsfadsfdfas', '2', '2022-04-03', 0),
(7, 2, '12412sfafsf', 'fsfssd431212421421', '2022-04-03', 4),
(8, 5, 'asfsadfafasfdfasd', 'asdfasfdfsafasfdasfdasf12321312321321321', '2022-04-03', 4);

--
--
-- Table structure for table `solution`
--

CREATE TABLE `solution` (
  `solutionID` int(11) NOT NULL,
  `projID` int(11) NOT NULL,
  `userID` int(11) NOT NULL,
  `solDesc` varchar(255) NOT NULL,
  `solGitLink` varchar(255) NOT NULL,
  `solFileURL` varchar(255) NOT NULL,
  `solUpVote` int(11) NOT NULL,
  `solDownVote` int(11) NOT NULL,
  `solCreateDate` date NOT NULL,
  `solEditDate` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `solution`
--

INSERT INTO `solution` (`solutionID`, `projID`, `userID`, `solDesc`, `solGitLink`, `solFileURL`, `solUpVote`, `solDownVote`, `solCreateDate`, `solEditDate`)
VALUES
(1, 1, 1, 'A fully implemented chat program written in C #1', 'chat program in C gitlink #1', 'chat program in C filelink #1', 22, 11, '2022-04-02', '2022-04-02'),
(2, 1, 2, 'An update to me amazing solution the amazing project. Im a leprechaun!', 'https://github.com/mySolutionUpdate.git', 'link.zip', 0, 0, '2022-04-02', '2022-04-02'),
(3, 2, 1, 'A fully implemented chat program written in C# #1', 'chat program in C# gitlink #1', 'chat program in C# filelink #1', 33, 1, '2022-04-02', '2022-04-02'),
(4, 2, 2, 'A fully implemented chat program written in C# #2', 'chat program in C# gitlink #2', 'chat program in C# filelink #2', 33, 1, '2022-04-02', '2022-04-02'),
(5, 1, 2, 'desc', 'link', 'file', 0, 0, '2022-04-02', '2022-04-02'),
(6, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02'),
(7, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02'),
(8, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02'),
(9, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02'),
(11, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02'),
(12, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02'),
(13, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02'),
(14, 1, 2, 'Record this API call', 'git.git.git', 'wtf.zip', 0, 0, '2022-04-02', '2022-04-02');

--
--
-- Table structure for table `statistic`
--

CREATE TABLE `statistic` (
  `statID` int(11) NOT NULL,
  `getAllUserAPI` int(11) NOT NULL,
  `getUserAPI` int(11) NOT NULL,
  `createUsersAPI` int(11) NOT NULL,
  `loginUserAPI` int(11) NOT NULL,
  `getProjectAPI` int(11) NOT NULL,
  `getAllProjectsAPI` int(11) NOT NULL,
  `createProjectAPI` int(11) NOT NULL,
  `updateProjectAPI` int(11) NOT NULL,
  `deleteProjectAPI` int(11) NOT NULL,
  `getSolutionAPI` int(11) NOT NULL,
  `getAllSolutionsAPI` int(11) NOT NULL,
  `createSolutionAPI` int(11) NOT NULL,
  `updateSolutionAPI` int(11) NOT NULL,
  `deleteSolutionAPI` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

4. Link to documentation page.

Link to OpenAPI document (Swagger)

<https://apiinfo.madface.info/spec.html>