# DB Intern Training: Student Glossary

This workshop is based on a web application that uses 3 main technologies: ReactJS, Python Flask and MySQL.

ReactJS is used to build components that provide the web user interface to the application. This communicates with a Python Flask API (Application Programming Interface) that interacts with the application data stored within a MySQL database. The Python Flask API provides endpoints that can be communicated over using HTTP (Hypertext Transfer Protocol) to call functions that extract data from the data store and return that data to be consumed by the React web front-end. The API also takes data sent to it via its endpoints and makes changes to the data stored in MySQL. MySQL is a relational database engine. It stores its data in tables.



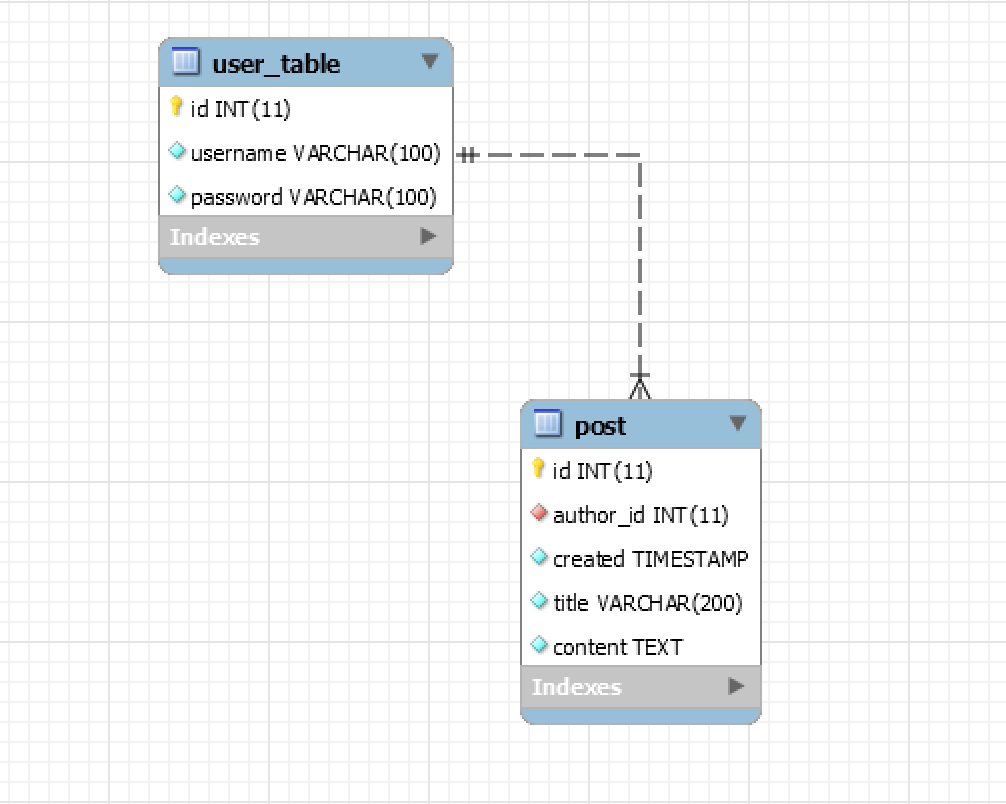
# Python & Flask

|  |  |
| --- | --- |
| Python | A high-level programming language designed to be easy to read. It is open source so is free to use. |
| Flask | Flask is a web microframework for building web apps with Python |
| app.py | In this file you create an app object which is an instance of the Flask object. It is the central configuration object for the entire application. You run this file to start the web API. |
| Routes.py | Routes are the flask application’s points of interaction (endpoints).  For example:  <http://localhost:5000/api/post>  <http://localhost:5000/api/post/update/1> |
| Middleware.py | A collection of functions that provide responses when a route is accessed. These functions interact with the data\_provider\_service.py functions which interact with the app’s data store and then provide appropriate responses as JSON data.  For example:  post\_by\_id  update\_post |
| data\_provider\_service.py | Provides connectivity to the data store and runs database queries and commands and returns the responses to the middleware which returns a web response to the client.  For example:  add\_post  get\_post\_with\_author |
| API URL | The Python Flask API can be accessed at the following URL:  <http://127.0.0.1:5000/api> |

# MySQL

MySQL is a relational database engine that stores its data in tables within a database created on a database server.

|  |  |
| --- | --- |
| Server | MySQL Server on localhost:  <http://127.0.0.1:3306/> |
| Database | blog\_flask |
| Tables | user\_table  post |



This is a very simple database that only contains two tables. Operational databases often contain hundreds or even thousands of tables.

# ReactJS

The web front-end is a ReactJS application. It is made up of user interface components that are rendered within a browser as HTML (Hypertext Mark-up Language), JavaScript and CSS (Cascading Style Sheets).

There are two main types of components: class components and function components.

The workshop uses the more modern syntax of function components.

## Example Class Component

import React, { Component } from 'react';

export default class hello\_class extends Component {

  render() {

    return (

      <div>

        <h1>Hello World!</h1>

      </div>

    );

  }

}

## Example Function Component

import React from 'react';

export default function hello\_function() {

  return (

    <div>

      <h1>Hello World!</h1>

    </div>

  );

}

# Tools

You will use various tools throughout the workshop:

|  |  |
| --- | --- |
| VS Code | An integrated development environment (IDE) for writing code with support for many languages. You will use this to create and edit your React code. |
| PyCharm Community Edition | An integrated development environment for Python programming. You will use this to edit the Python Flask API. |
| Postman | Postman is an API client that makes it easy to create and save simple and complex HTTP/s requests, as well as read their responses. |
| MySQL Workbench | MySQL Workbench is a graphical tool for database developers and database administrators (DBAs).  You can run scripts to create database objects and queries to extract data stored within the tables. |
| pip | Pip is a package management system used to install and manage software packages for Python.  Pip installs the packages under the site-packages folder by default. |
| Yarn / npm | Yarn and NPM (node package manager) are package managers used with JavaScript development. They both download packages from the NPM repository.  To use yarn or npm to create a react application:  For example, using yarn:  yarn create react-app hello-app  cd hello-app  yarn start  For example, using npm:  npx create-react-app hello-app  cd hello-app  npm start  NPM uses the npx command to execute scripts but the npm command to download packages. |