

PS-2 REPORT

PREPARED BY

Pradhuman Singh (2018btechcse011)

FACULTY GUIDE

Dr. Sonal Jain



**Department of computer science engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur**

May 2022

Business Management Web Application

Practice School-2

Submitted in partial fulfilment of the requirements

for the degree of

Bachelor of Technology in Computer Science Engineering

By:

Pradhuman Singh (2018btechcse011)

Faculty Guide:

Dr Sonal Jain



**Department of computer science engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur**

May 2022

ABSTRACT

This internship has helped me to gain real world experience. And in this report I have shown what I have learned in these 120 days.

In my internship with L and A pvt ltd, I was a backend developer in their project called gstart, I learned some basic technologies which I have previously used in college but now I gained some experience on them. PHP, Postman, NodeJS, MySQL, Sequelize, Express, Migrations, Bitbucket, RestAPI and many small things I have learned in this internship.

This report focuses on the work done during my internship. The dashboards are attached as images.

List of Figures

S.NO	FIGURE DISCRIPTION	PAGE
1	Xampp control panel	13
2	PhpMyAdmin dashboard	13
3	Postman dashboard	15
4	mySQL server working model	18
5	Migration model	21
6	Constraints of rest API	24
7	Model	27
8	Routes	27
9	Controller	28
10	Index	28
11	config	29

List of Acronyms

HTML	- Hypertext Markup Language
CSS	- Cascading Style Sheets
API	- Application Programming Interface
REST	- REpresentational State Transfer
PHP	- Hypertext Preprocessor

Contents

Topic	page number
Objective	9
Introduction	10
Technologies	11
PHP	11
Postman	13
NodeJS	15
MySQL	16
Sequalize	19
Express	19
Migrations	20
Bitbucket	22
RestAPI	23
API example	26
Conclusion	29

OBJECTIVE

Understand basic concept and structure of MySQL, creating Database, NodeJS, sequelize

Learn how to make APIs

To perceive communicational skills and organisational dynamics.

To learn office etiquettes

ABOUT L AND A PVT LTD

Lekhwani and Associates pvt ltd is a financial and accounting consultancy firm which provides services and solutions to their prestigious clients, majorly in Jaipur.

the mission is to develop a new web application which could take inputs from various sources and provide data to management for their future decisions, and help generate transparency in the system of GST and other accounting related work in the organisation.

The web application was to be started with the database they have already been using in their previous application, and start the new application from scratch.

We were to use react and nodeJS to develop the application. I was working on the backend and was using mySQL and nodejs to develop some APIs to work with on the front end.

L and A private limited is a private incorporated on 10th June 2019. It is classified as non-govt company and is registered att registrar of companies, Jaipur.

TECHNOLOGIES INTRODUCED

PHP :

PHP (hypertext preprocessor) is known as a general- purpose scripting language that can be used to develop dynamic and interactive websites. PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. PHP was created by **Rasmus Lerdorf** in **1994** but appeared in the market in 1995. It was among the first server-side languages that could be embedded into HTML, making it easier to add functionality to web pages without needing to call external files for data.

- PHP stands for Hypertext Preprocessor.
- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is faster than other scripting languages, for example, ASP and JSP.
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is an open-source scripting language.

Why should we use PHP?

PHP is a server-side scripting language, which is used to design the dynamic web applications with MySQL database.

- It handles dynamic content, database as well as session tracking for the website.
- You can create sessions in PHP.
- It can access cookies variable and also set cookies.
- It helps to encrypt the data and apply validation.
- PHP supports several protocols such as HTTP, POP3, SNMP, LDAP, IMAP, and many more.
- Using PHP language, you can control the user to access some pages of your website.
- As PHP is easy to install and set up, this is the main reason why PHP is the best language to learn.

- PHP can handle the forms, such as - collect the data from users using forms, save it into the database, and return useful information to the user. **For example** - Registration form.

I was using Xampp to host php

The screenshot displays the XAMPP Control Panel v3.3.0. The top half features a table of services with columns for Service, Module, PID(s), Port(s), and Actions. Below this is a row of buttons for Config, Netstat, Shell, Explorer, Services, Help, and Quit. The bottom half is a log window showing the following messages:

```

04/27/2021 13:00:00 [main] Initializing Control Panel
04/27/2021 13:00:00 [main] Windows Version: Home 64-bit
04/27/2021 13:00:00 [main] XAMPP Version: 8.0.15
04/27/2021 13:00:00 [main] Control Panel Version: 3.3.0 [ Compiled: Apr 6th 2021 ]
04/27/2021 13:00:00 [main] You are not running with administrator rights! This will work for
04/27/2021 13:00:00 [main] most application stuff but whenever you do something with services
04/27/2021 13:00:00 [main] there will be a security dialogue or things will break! So think
04/27/2021 13:00:00 [main] about running this application with administrator rights!
04/27/2021 13:00:00 [main] XAMPP Installation Directory: "c:\xampp\"
04/27/2021 13:00:00 [main] Checking for prerequisites
04/27/2021 13:00:00 [main] All prerequisites found
04/27/2021 13:00:00 [main] Initializing Modules
04/27/2021 13:00:00 [main] Enabling autostart for module "MySQL"
04/27/2021 13:00:00 [main] Starting Check-Timer
04/27/2021 13:00:00 [main] Control Panel Ready
04/27/2021 13:00:00 [mysql] Autostart active: starting...
04/27/2021 13:00:00 [mysql] Attempting to start MySQL app...
04/27/2021 13:00:00 [mysql] Status change detected: running
04/27/2021 13:00:00 [Apache] Attempting to start Apache app...
04/27/2021 13:00:00 [Apache] Status change detected: running
  
```

phpMyAdmin

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Design Central column

Recent Favorites

New

Database

Consolidating the users

Table	Action	Rows	Type	Collation	Size	Overhead
account_bank	Browse Search Insert Empty Copy	12	MyISAM	latin1_general_ci	50.0 x 1.0	
account_bank_associations	Browse Search Insert Empty Copy	10	MyISAM	latin1_general_ci	10.0 x 1.0	
agent_details	Browse Search Insert Empty Copy	0	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
access_mappings	Browse Search Insert Empty Copy	42	MyISAM	latin1_general_ci	10.0 x 1.0	
agents	Browse Search Insert Empty Copy	0	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
best_mappings	Browse Search Insert Empty Copy	0	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
businesses	Browse Search Insert Empty Copy	0	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
businesses_entities_branches	Browse Search Insert Empty Copy	10	MyISAM	latin1_general_ci	10.0 x 1.0	
businesses_entities	Browse Search Insert Empty Copy	10	MyISAM	latin1_general_ci	10.0 x 1.0	
businesses_entity_contact	Browse Search Insert Empty Copy	3	MyISAM	latin1_general_ci	10.0 x 1.0	
businesses_locations	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
businesses_nature	Browse Search Insert Empty Copy	0	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
businesses_nature_groups	Browse Search Insert Empty Copy	0	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
businesses_nature_group_mappings	Browse Search Insert Empty Copy	0	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
cars	Browse Search Insert Empty Copy	1	MyISAM	latin1_general_ci	10.0 x 1.0	
clients	Browse Search Insert Empty Copy	1	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
connections	Browse Search Insert Empty Copy	15	MyISAM	latin1_general_ci	10.0 x 1.0	
consumer_names	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
countries	Browse Search Insert Empty Copy	249	MyISAM	latin1_general_ci	10.0 x 1.0	
country_names	Browse Search Insert Empty Copy	249	MyISAM	latin1_general_ci	10.0 x 1.0	
deductions_types	Browse Search Insert Empty Copy	18	MyISAM	utf8mb4_general_ci	10.0 x 1.0	
default_ledgers	Browse Search Insert Empty Copy	16	MyISAM	latin1_general_ci	10.0 x 1.0	
deduction_types	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
default_ledgers	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search Insert Empty Copy	0	MyISAM	latin1_general_ci	10.0 x 1.0	
driver_details	Browse Search					

POSTMAN:

Postman is one of the most popular software testing tools which is used for API testing. With the help of this tool, developers can easily create, test, share, and document APIs. About postman:

- Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.
- While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API.
- In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

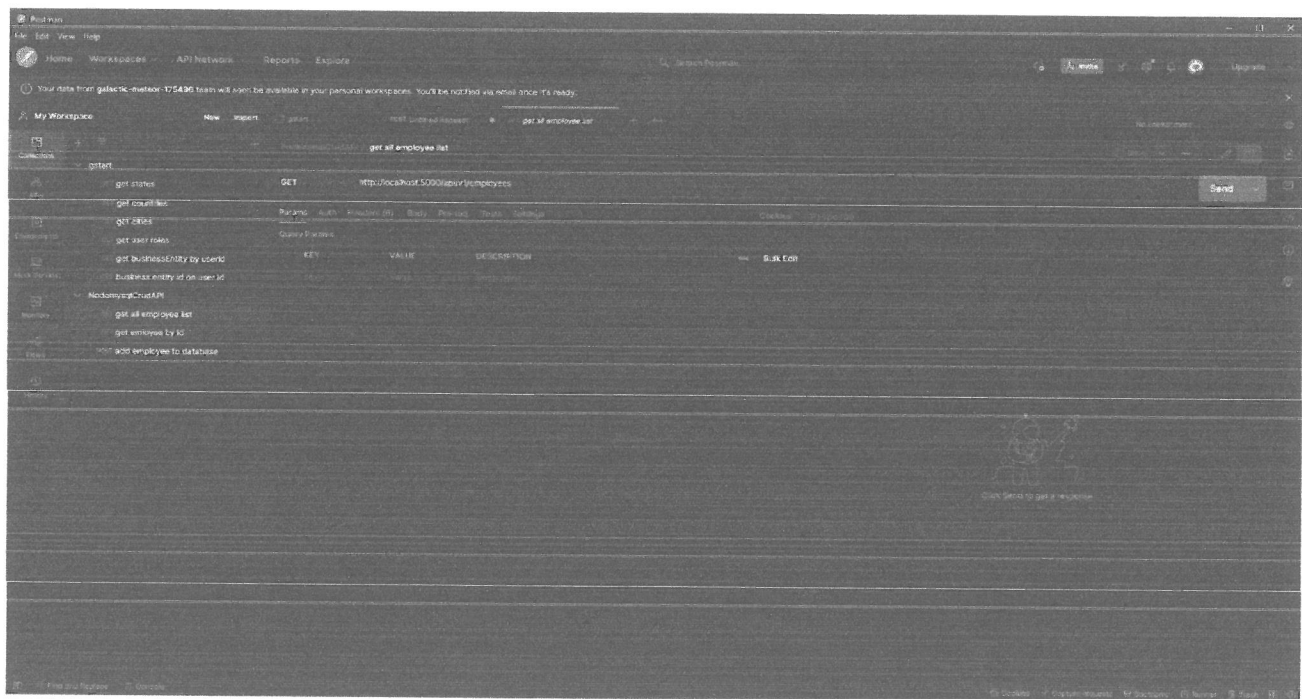
What is API?

Application Programming Interface (API) is software that acts as an intermediary for two apps to communicate with each other. We use APIs whenever we use an application like Twitter, Facebook, sending text messages, or checking the weather over the phone.

Why use postman?

1. **Accessibility-** One can use it anywhere after installing Postman into the device by simply logging in to the account.
2. **Use Collections-** Postman allows users to build collections for their API-calls. Every set can create multiple requests and subfolders. It will help to organize the test suites.
3. **Test development-** To test checkpoints, verification of successful HTTP response status shall be added to every API- calls.

4. **Automation Testing**-Tests can be performed in several repetitions or iterations by using the Collection Runner or Newman, which saves time for repeated tests.
5. **Creating Environments**- The design of multiple environments results in less replication of tests as one can use the same collection but for a different setting.
6. **Debugging**- To effectively debug the tests, the postman console helps to track what data is being retrieved.
7. **Collaboration**- You can import or export collections and environments to enhance the sharing of files. You may also use a direct connection to share the collections.
8. **Continuous integration**-It can support continuous integration.



NODEJS :

Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use.

Many of the basic modules of Node.js are written in JavaScript. Node.js is mostly used to run real-time server applications.

The definition given by its official documentation is as follows:

?Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.?

Node.js also provides a rich library of various JavaScript modules to simplify the development of web applications.

Why nodeJS?

1. **Extremely fast:** Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.
2. **I/O is Asynchronous and Event Driven:** All APIs of Node.js library are asynchronous i.e. non-blocking. So a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.
3. **Single threaded:** Node.js follows a single threaded model with event looping.
4. **Highly Scalable:** Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.
5. **No buffering:** Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.
6. **Open source:** Node.js has an open source community which has produced many excellent modules to add additional capabilities to Node.js applications.
7. **License:** Node.js is released under the MIT license.

MySQL:

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by **Oracle Company**.

What is database?

It is very important to understand the database before learning MySQL. A database is an application that stores the organized collection of records. It can be accessed and managed by the user very easily. It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly. Each database contains distinct API for performing database operations such as creating, managing, accessing, and searching the data it stores. Today, many databases are available like MySQL, Sybase, Oracle, MongoDB, PostgreSQL, SQL Server, etc.

What is MySQL?

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by **MySQL AB, a Swedish company**, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is *My Ess Que Ell*. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

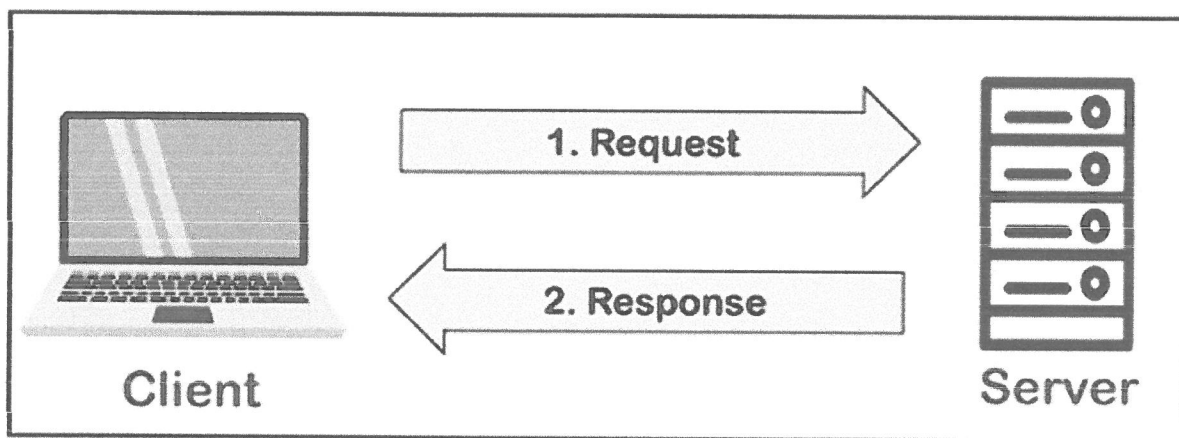
MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It allows us to implement database operations on tables, rows, columns, and indexes.
- It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
- It provides the Referential Integrity between rows or columns of various tables.
- It allows us to update the table indexes automatically.

- It uses many SQL queries and combines useful information from multiple tables for the end-users.

How does mysql works?

MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of MySQL environment is the same as the client-server model.



The core of the MySQL database is the MySQL Server. This server is available as a separate program and responsible for handling all the database instructions, statements, or commands. The working of MySQL database with MySQL Server are as follows:

1. MySQL creates a database that allows you to build many tables to store and manipulate data and defining the relationship between each table.
2. Clients make requests through the GUI screen or command prompt by using specific SQL expressions on MySQL.
3. Finally, the server application will respond with the requested expressions and produce the desired result on the client-side.

Reasons for popularity:

- MySQL is an open-source database, so you don't have to pay a single penny to use it.
- MySQL is a very powerful program that can handle a large set of functionality of the most expensive and powerful database packages.
- MySQL is customizable because it is an open-source database, and the open-source GPL license facilitates programmers to modify the SQL software according to their own specific environment.
- MySQL is quicker than other databases, so it can work well even with the large data set.
- MySQL supports many operating systems with many languages like PHP, PERL, C, C++, JAVA, etc.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL is very friendly with PHP, the most popular language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

Some commonly used queries :

SELECT ,ALTER ,DELETE ,DROP ,INSERT ,UPDATE

Some clauses used were:

WHERE, GROUP BY, ORDER BY, HAVING

I learned JOIN in mySQL

Inner join, left join., right join

Some conditions used:

AND , OR , NOT , LIKE , IS NULL , NOT NULL , BETWEEN

SEQUALIZE:

Sequelize is a dynamic and powerful JavaScript library. It is a promise-based ORM (**Object Relational Mapping**) that deals with the management versions of the database primarily SQL database. It can also support other databases like PostgreSQL, MySQL, SQLite, and MSSQL. It is also popularly known as **Object-Relation Mapper** because it maps object syntax into a database schema. Although there are no prerequisites for learning it familiarity with Node.js and JavaScript syntax used for mapping would be fruitful.

it is important to note that Sequelize is promise-based. Thus, it makes the processes easier to manage about the occurring exceptions and asynchronous functions. On the other hand, there's no support for NoSQL, and can be only done if Object Document Mapper is present on the dependencies like that of **Mongoose**. This fact depends on the ORM which can further relate mainly to the project requirement we're working on. However, Sequelize supports various types of actions be it solid transaction support, reading and writing applications, the relationship among transactions, and much more.

EXPRESS:

Express is a fast, assertive, essential and moderate web framework of Node.js. You can assume express as a layer built on the top of the Node.js that helps manage a server and routes. It provides a robust set of features to develop web and mobile applications.

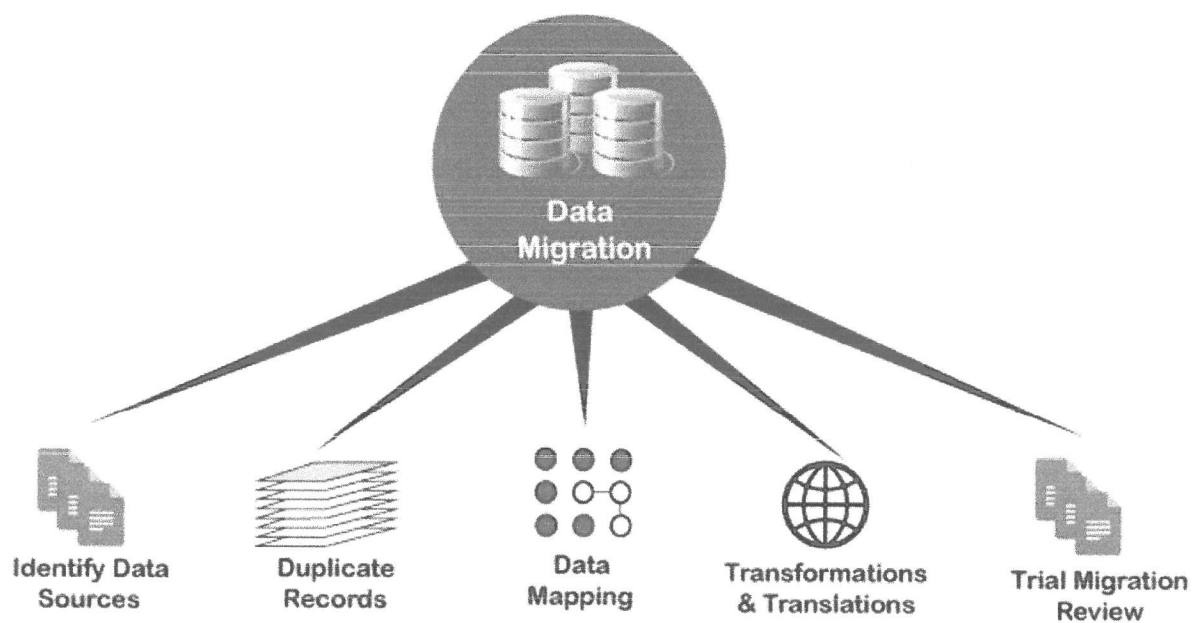
- It can be used to design single-page, multi-page and hybrid web applications.
- It allows to setup middlewares to respond to HTTP Requests.
- It defines a routing table which is used to perform different actions based on HTTP method and URL.
- It allows to dynamically render HTML Pages based on passing arguments to templates.

Why express?

- Ultra fast I/O
- Asynchronous and single threaded
- MVC like structure
- Robust API makes routing easy.

MIGRATIONS:

Data migration is a useful method of transferring data from one device to another. Although this can sound very simple, a shift in storage and database or programme is involved. Any data migration would require at least the transform and load phases in the sense of the extract/transform/load process. This method suggests that in planning, retrieved information goes through a sequence of functions and then it will be placed into a target position. Companies conduct data migrations for various purposes, such as they need to redesign an entire infrastructure, upgrade databases, build a new data warehouse, or integrate new data from an acquisition or other source. When installing another framework that sits alongside current applications, data replication is often important .



Data movers are in 3 major broad groups:

- For application-specific migrations, such as platform upgrades, replication of databases and file copying, host-based software is best.
- Software-based on arrays is mostly used to transfer data between related structures.
- Depending on their configuration, network appliances migrate volumes, files or blocks of data.

Migrating data is not to be confused with data integration. Data transfer requires transferring or copying data from one device to another, or from one environment to another, while the transformation of data deals with the movement of data between different programmes and structures. Administrators that incorporate data should be familiar with the technology of extraction, processing and loading. This method has its own underlying layers of complexity, and each one has its own data model. Applications are not intended to be handheld. Control tools, operating systems and implementations of virtual machines will all vary from those in the context in which the programme has been created or implemented. In order to overcome technological gaps, successful programme migration can involve the use of middleware goods.

Database relocation is completed if there is a need to switch storage providers, upgrade the database architecture or migrate a database to the cloud. In this type of transition, the underlying data will shift and impact the application layer when a protocol or user language update occurs. Replication of server data deals with the modification of data without changing the schema. Determining the archive's size to decide how much room is available, reviewing codes, and protecting the document's confidentiality are some of the key activities. During the migration process, compatibility problems may occur, so it is necessary to first check the process.

Storage Migration: Via infrastructure upgrades, storage transfer is justified, and the process is seen as a perfect time to verify and minimize data by detecting outdated or corrupt data. The method involves transferring data blocks and files, whether on the disc, tape or the cloud, from one storage device to another. There are various items and instruments for storage relocation that help smooth the process. Storage relocation also provides the ability to fix any orphaned storage or inefficiencies.

Bitbucket:

Bitbucket is a repository management tool which is mainly designed for expert teams and professionals. Bitbucket is a Git repository management software that serves as a central hub for all the Git repositories. We use Bitbucket for access control, workflow control, pull request and integration for full rest API.

Bitbucket was launched in 2008 as an independent startup in Australia specializing in Mercurial project hosting. It was acquired by Atlassian, a fellow Australian corporation, in 2010, and nearly a year later added support for Git repository.

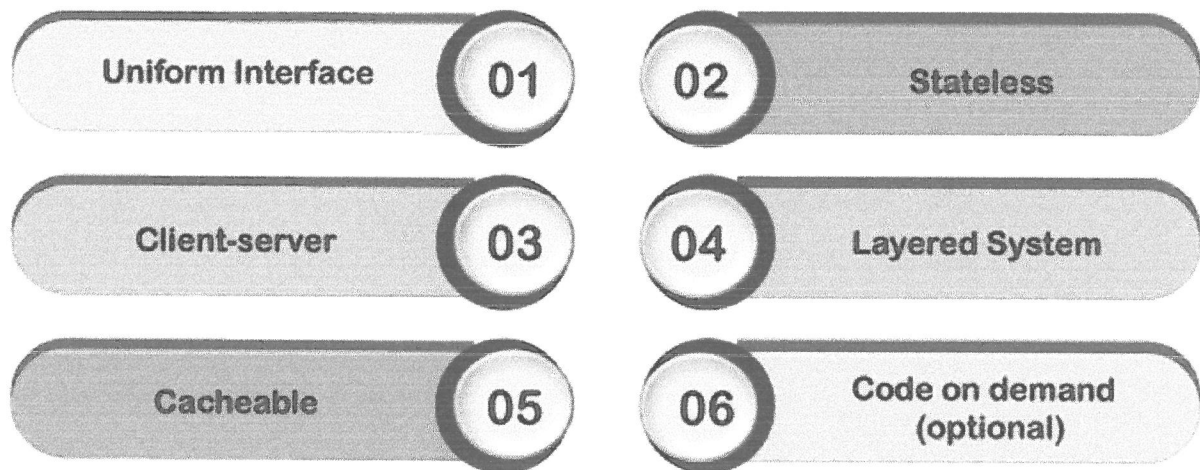
Features:

- In bitbucket, we can create unlimited private repositories.
- Bitbucket has a strong integration with Trello and permits third-party API support.
- It supports Git Large File Storage (LFS) in order to reduce clone time.
- Explicitly manages branch permissions and allows the developer to restrict branch access.
- Bitbucket for Enterprises, known as Stash.
- History view with branch comparison.
- Pull requests and code reviews.
- Integration with tools such as Jenkins, Bamboo, Jira, Crucible.
- Built-in continuous delivery, issue tracking, and wikis.
- Simple integration with Bamboo and Confluence, as well as Atlassian's Jira Software Cloud for further customization and hosting.
- Bitbucket snippets let you build and manage a wide range of multi-line snippets.
- Bitbucket can be integrated into many aspects of our current workflow using third-party integrations, making the entire development process far more effective.

REST API :

REpresentational State Transfer (REST) is a software architectural style that defines the constraints to create web services. The web services that follows the **REST** architectural style is called **RESTful Web Services**. It differentiates between the computer system and web services. The REST architectural style describes the six barriers.

CONSTRAINTS OF REST ARCHITECTURE



1. Uniform interface

The Uniform Interface defines the interface between **client** and **server**. It simplifies and decomposes the architecture which enables every part to be developed. The Uniform Interface has four guiding principles:

- **Resource-based:** Individual resources are identified using the URI as a resource identifier. The resources themselves are different from the representations returned to the customer. For example, the server cannot send the database but represents some database records expressed to **HTML**, **XML** or **JSON** depending on the server request and the implementation details.
- **Manipulation of resources by representation:** When a client represents a resource associated with metadata, there is information on the server to modify or delete it.
- **Self-Descriptive Message:** Each message contains enough information to describe how the message is processed. For example, the parser can be specified by the Internet media type (known as the **MIME** type).

- **As the engine of Hypermedia Application State (HATEOAS):** Customers provide states by query-string parameters, body content, request headers, and requested URIs. The services provide customers with the state by **response codes, response headers and body content**. It is called hypermedia (**hyperlink within hypertext**).
- In addition to the above description, HATEOS also means that, where necessary, the object or itself is contained in the linked body (or header) to supply the URI for retrieving the related objects.
- The same interface that any **REST services** provide is fundamental to the design.

2. Client-server

A client-server interface separates the client from the server. For Example, the separation of concerns not having an internal relationship with internal storage for each server to improve the portability of customer's data codes. Servers are not connected with the user interface or user status to make the server simpler and scalable. Servers and clients are independently replaced and developed until the interface is changed.

3. Stateless

Stateless means the state of the service doesn't persist between subsequent requests and response. It means that the request itself contains the state required to handle the request. It can be a query-string parameter, entity, or header as a part of the **URI**. The URI identifies the resource and state (or state change) of that resource in the unit. After the server performs the appropriate state or status piece (s) that matters are sent back to the client through the header, status, and response body.

- Most of us in the industry have been accustomed to programming with a container, which gives us the concept of "**session**," which maintains the status among multiple HTTP requests. In **REST**, the client may include all information to fulfil the server's request and multiple requests in the state. Statelessness enables greater scalability because the server does not **maintain, update, or communicate** any session state. The resource state is the data that defines a resource representation.

Example, the **data stored** in a database. Consider the application state of having data that may vary according to client and request. The resource state is constant for every customer who requests it.

4. Layered system

It is directly connected to the end server or by any intermediary whether a client cannot tell. Intermediate servers improve the system **scalability** by enabling **load-balancing** and **providing** a shared cache. Layers can enforce security policies.

5. Cacheable

On the World Wide Web, customers can cache responses. Therefore, responses clearly define themselves as unacceptable or prevent customers from **reusing** stale or **inappropriate data** to further requests. **Well-managed** caching eliminates some client-server interactions to improving scalability and performance.

6. Code on demand

The server temporarily moves or optimizes the functionality of a client by logic that it executes. Examples of compiled components are **Java applets** and **client-side scripts**.

Compliance with the constraints will enable any distributed hypermedia system with desirable contingency properties such as **performance**, **scalability**, **variability**, **visibility**, **portability**, and **reliability**.

API is the abbreviation of the term ***Application Programming Interface***. It is the software responsible for the connection for the communication and information exchange between two apps. API connects two devices or programs in order to facilitate the exchange of information between them. It is the interface that serves the other parts of the software. The API specifications are the standards or documents designed to describe the creation of such connections. If a computer system meets these standards, then it is said to expose an API. The specification or implementation both are known as the API

API example

get API is used to get information from model to controller.

Example of this API is

```
src > models > JS employee.model.js > [e] Employee
1  var dbConn = require('../config/db.config');
2
3  var ... = function(Employee) {}
4      this.first_name = Employee.first_name;
5      this.last_name  = Employee.last_name;
6      this.email      = Employee.email;
7      this.phone      = Employee.phone;
8      this.organisation = Employee.organisation;
9      this.designation = Employee.designation;
10     this.salary      = Employee.salary;
11     this.status      = Employee.status ? Employee.status : 1;
12     this.created_at  = new Date();
13     this.updated_at  = new Date();
14 }
15
16 //get all employees
17 .getAllEmployees = (result) =>{
18     dbConn.query('SELECT * FROM employees', (err, res)=>{
19         if(err){
20             console.log('error while fetching employees', err);
21             result(null, err);
22         }else{
23             console.log('employees fetched successfully')
24             result(null, res);
25         }
26     })
27 }
28
```

This is the model

```
src > routes > JS employee.route.js > ...
1  const express = require('express');
2  const { route } = require('express/lib/application');
3  //const { router } = require('express/lib/application');
4  const router = express.Router();
5
6  const EmployeeController = require('../controllers/employee.controller');
7  // get all employees
8  router.get('/', EmployeeController.getEmployeeList);
9
10 //get employee by id
11 router.get('/:id', EmployeeController.getEmployeeById);
12
13 //create new employee
14 router.post('/', EmployeeController.createNewEmployee);
15
16 module.exports = router;
```

This is the routes

```

src > controllers > JS employee.controller.js > getEmployeeByid > getEmployeeByid
1  const      = require('../models/employee.model');
2
3  // get all employee list
4  exports.getEmployeeList = (req,res)=>{
5      //console.log('here all employee list');
6      .getAllEmployees((err, employees) =>{
7          console.log('we are here');
8          if(err)
9              res.send(err);
10             console.log('Employees', employees);
11             res.send(employees)
12         })
13     }
14
15 //get employee by id
16 exports.getEmployeeByid = (req,res)=>{
17     //console.log('get emp by id');
18     .getEmployeeByid(req.params.id, (err, employee) =>{
19         //console.log('employee by id is here');
20         if(err)
21             res.send(err);
22         console.log('single Employee data', employee);
23         res.send(employee)
24     })
25 }

```

This is the controller

```

JS: index.js > ...
1  const express = require('express');
2  const bodyParser = require('body-parser');
3
4  // create express app
5  const app = express();
6
7  //setup server port
8  const port = process.env.PORT || 5000;
9
10 //parse request data content type application/x-www-form-urlencoded
11 app.use(bodyParser.urlencoded({extended: false}));
12
13 //parse request data content type application/json
14 app.use(bodyParser.json());
15
16 // define root route
17 app.get('/', (req,res)=>{
18     res.send('HELLO WORLD');
19 });
20
21 //import employee routes
22 const employeeRoutes = require('./src/routes/employee.route');
23 const bodyParser = require('body-parser');
24
25 //create employee routes
26 app.use('/api/v1/employees', employeeRoutes);
27
28
29 //listen to the port
30 app.listen(port, ()=>{
31     console.log('Express server is running at port ${port}');
32 });
33

```

This is the index file

```
config > JS db.config.js > ...
1  const mysql = require('mysql');
2  // create here mysql connection
3  const dbConn = mysql.createConnection({
4    host: 'localhost',
5    user: 'root',
6    password: '',
7    port: '3307',
8    database: 'node_mysql_crud_db'
9  });
10
11  dbConn.connect(function(error){
12    if(error) throw error;
13    console.log('database connected succesfully!!!');
14  })
15
16  module.exports = dbConn;
```

This is the config file

There are 2 APIs here.

1. get all employees
2. get employee by employee id

the first API will get the whole list of data

the second api will take input from us to get specific data from the whole list.

Conclusion

The internship program has been an excellent and rewarding experience. I can conclude that there have been a lot I've learnt from my work at L and A. needless to say, the technical aspect of the work I've done are not flawless and could be improved provided enough time. As someone with no prior experience with NodeJs whatsoever I believe my time spent in research and discovering it was well worth it and contributed to finding an acceptable solution to build a fully functionable web service. Two main things I've learned the importance of are time-management skills and self-motivation.

L AND A PRIVATE LIMITED

CIN U52602RJ2019PTC065223

Basement, 112/16, Pratap Nagar Sanganer, Jaipur 302033

Contact 9549946999 Email info.alandae@gmail.com

INTERNSHIP CERTIFICATE

Date: [22/05/2022]

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Pradhuman Singh has done his internship in full stack web application at L and A pvt ltd, jaipur, from 17/01/2022 to 17/05/2022 for the duration of 4 months .

He has worked on a project titled Gstart. This project was aimed at creating a web application using react and nodejs. As part of the project, he has done work in full stack.

During his internship he has demonstrated his skills with self-motivation to learn new skills. His performance exceeded our expectations and he was able to complete the project on time.

We wish him all the best for his upcoming career.

Company name: L & A Pvt. Ltd.

Issuing person name: Nishant Mathur

Designation: Project manager

LAND A PRIVATE LIMITED

Nishant

Authorised Signatory

[Signature][Seal of Company]