

An Industrial Internship Report

Submitted in partial fulfillment for the award of the degree of

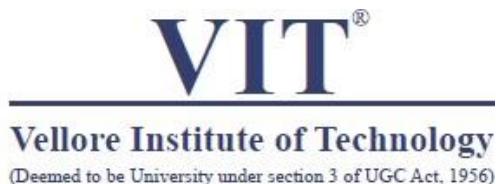
B. Tech

in

Information Technology

by

Priyal Bhardwaj 18BIT0272



December, 2021

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled "**INDUSTRIAL INTERNSHIP REPORT**" submitted by me to School of Information Technology & Engineering, Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the degree of **B.Tech (Information Technology)** is a record of bonafide **Industrial Internship – (ITE3099)** work carried out by me. I further declare that the work reported in this **Industrial Internship report** has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore



Date: 1st December, 2021

Priyal Bhardwaj



VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the Industrial Internship report entitled "**INDUSTRIAL INTERNSHIP**" submitted by **Priyal Bhardwaj (18BIT0272)** to School of Information Technology & Engineering, Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the degree of **B.Tech (Information Technology)** is a record of bonafide **Industrial Internship – (ITE3099)** work carried out by her in **Optum – UnitedHealth Group**. The **Industrial Internship** project fulfills the requirements as per the regulations of this Institute.

A handwritten signature in blue ink, appearing to read "Manjula".

Examiner – Panel In Charge

(Name and Signature)

Dr.V.Manjula

Date 01-12-2021

COMPANY CERTIFICATE XEROX TO BE ENCLOSED HERE



To Whom It May Concern

Date: 3-Sep-2021

Name: Priyal Bhardwaj

Unique ID: 001688970

This is to certify that **Priyal Bhardwaj D/o UNMESH BHARDWAJ** has completed her internship with **Optum Global Solutions (India) Pvt. Ltd** as an Intern with Technology division from **16-Jun-2021 to 15-Aug-2021**. Her conduct during the aforesaid period was found to be satisfactory.

We wish her all the best in her future endeavor.

For Optum Global Solutions (India) Pvt. Ltd

A handwritten signature in blue ink, appearing to read "Sumek Gopal".

Sumek Gopal

Vice President | Human Capital

ACKNOWLEDGEMENT

I would like to thank Mr. Andrew Witty, CEO of Optum - UnitedHealth Group, for giving me the opportunity to work there. I would also like to express my sincere gratitude towards Mr. Rajat Bansal (Sr. Manager), Mr. Venkat Reddy (Manager), Mr. Gaurav Nagpal (Mentor) and Mr. Sonu Mehta (Quality Consultant), at Optum - UnitedHealth Group, for their valuable and constructive suggestions during my time there. I would also like to thank my fellow teammates for working with me and helping me throughout the project.

Special thanks to the School of Information Technology, VIT, and its faculties for providing me with the knowledge required to work at Optum - UnitedHealth Group. I would also like to express my sincere gratitude towards my parents and friends for believing in me and supporting me throughout my journey.

TABLE OF CONTENTS

Chapter	Contents
1	The Company - Optum - UnitedHealth Group
2	Responsibilities
3	Languages, Frameworks, Tools and Services Used
4	Development Process
5	Database Management
6	Spring Rest API
7	Additional Responsibilities
8	Conclusion



THE COMPANY: OPTUM – UNITEDHEALTH GROUP

ABOUT

1.11 Introduction

Optum - UnitedHealth Group, is an American pharmacy benefit manager and health care provider. It is a subsidiary of UnitedHealth Group since 2011. UHG formed Optum by merging its existing pharmacy and care delivery services into the single Optum brand, comprising three main businesses: OptumHealth, OptumInsight and OptumRx. In 2017, Optum accounted for 44 percent of UnitedHealth Group's profits and as of 2019, Optum's revenues have surpassed \$100 billion. Optum has been operating in India since 2006 and has teams supporting health care operations, product development, automation, analytics and data solutions.

1.12 Work

Optum is a **health information technology and services firm** that is part of UnitedHealth Group. It provides technological, operational and consulting solutions and services to individuals, healthcare organizations, pharmaceutical companies as well as the federal and state governments.

1.13 Product

Earlier, a lot of companies followed Waterfall methodologies where releasing apps were scheduled once per year or even on longer intervals. But with the evolution of technologies, Agile development started to be adopted. The release cycles for most apps got shorter and shorter - from 1-2 weeks to a couple of

times per day. Earlier, following waterfall methodology the apps were manually tested. But with the shortening of the cycles over and over again, it became difficult to keep up. This is also so, because with time old applications got bigger and bigger. The companies needed more testers or had to release their apps with a higher risk of bugs which reduces reliability as already mentioned. For this primary problem, the test automation serves as a solution. ETAAS stands for enterprise test automation as a service. The aim of this project is to run tests automatically with little or no manual intervention in order to make the overall development and testing process easy and efficient. The key factor of ETAAS is automation. The basic expectation of ETAAS is to create common methods for all 4 types testing which can be used to test a wide range of projects and the tester could also perform complex testing scenarios like API validation, UI check and performance check in a single scenario. Finally, automation enables the tester/developer to focus on other issues such as customer needs, functionalities and improvements.

1.14 Benefit of the Product

The key factor of ETAAS is: automation. The testing process is automated as the user simply has to interact with the UI to create projects and testcases. All this can be done only after the user has successfully onboarded and details regarding his projects and testcases will be saved in one single space. Also, the tests can be recycled and reused for other projects so we do not have to go through the whole process again. The aim of ETAAS could be creating Generic methods required for testing UI using Selenium, API using RestAssured, Front-end performance using SauceLabs and Security using Zap alongside creating common methods for all 4 types testing which can be used to test a wide range

of projects and the tester could also perform complex testing scenarios like API validation, UI check and performance check in a single scenario.

DEVELOPMENT TEAM

1.21 Scrum Master

Mr. Lalit Gandhi

He was the Scrum Master for our team and conducted daily scrum meetings in which we had to update our tasks and discuss and technical issues we had faced.

1.22 Senior Manager

Mr. Rajat Bansal

He took care of any Virtual Desktop Infrastructure (VDI) issues and regularly gave insights on the project. He helped immensely while preparing for reviews with the Senior Directors.

1.23 Senior Software Developer

Mr. Gaurav Nagpal

All tasks were to be discussed with him. He was to be contacted in case of any doubts. He conducted daily progress meetings to check the functionality and progress and assign new tasks.

Mr. Venkat Reddy

In Gaurav Sir's absence he was to be approached for any issues.

1.24 Stakeholder

Mr. Sonu Mehta

He was the main stakeholder for the project and was to be approached for any doubts regarding the whole idea of the project.

1.25 Frontend Team

At the time of my joining, the Frontend Team had 2 members who were also fellow interns.

1.26 Backend Team

The Backend Team did not have anyone apart from myself though I was encouraged to include other interns as well if needed.

RESPONSIBILITIES

2.11 Post

I joined the team as a Software Development Intern.

2.12 Duration

My internship started on 16th June, 2021, for a period of 2 months. I officially completed my internship on 15th August, 2021

2.13 Working Hours

The working hours were flexible and the working days were from Monday to Friday. Sometimes due to a deadline, it got extended to Saturdays as well.

MEANS OF COMMUNICATION

2.21 Meetings

Regular meetings were conducted on Microsoft Teams, at least twice day.

2.22 Agenda of the meetings

In the meetings, I had to give a detailed description of what I was working on, the extent of completion, by when will it be completed and what is the progress of the project. I was also told about the next functionality I have to develop after my current work has been completed. In the meeting, further functionalities to be built were discussed.

2.23 Other communication

Apart from meetings, quick calls were also done at times to discuss any bugs of utmost importance.

After the deployment of the project, meetings were conducted to discuss bugs, additional features or to conduct a Usability Test.

The screenshot shows the Rally software interface with the title 'Vikings Standup'. The main view displays the 'Iteration Status' board for 'R31 Sprint 1'. Key metrics shown include 'Planned Velocity' (102% of 40 Points), 'Iteration End' (5 of 10 days left), and 'Accepted' (48/75 of 61 Points). The board lists various tasks and their details, such as 'Shib Automobile Backend UI to create and publish rally template...', 'CBV API Prod and Metal LB config for dev and PROD...', and 'Report Utilization Action planning - 5 reports...'. On the right side, there is a 'Team' section showing profiles for team members like RJ, Gandh, Lilit, Neeraj, Reddy, PL, SS, LP, GS, and PB, each with a small profile picture and some status indicators. The bottom of the screen shows the Windows taskbar with icons for various applications and the system tray.

DAY TO DAY ACTIVITIES

- Create ER Diagram for the modules as per the user requirement for the project.
- Develop the REST APIs using Spring Boot, a Java Framework, and any SQL Database (preferable MySQL).
- Test the modules, using any software testing tools, like Postman and send it to the supervisor.
- Create proper documentation of the REST APIs developed and deliver it to the Frontend Team.
- Assist in the integration of the REST APIs to Front-end, using AngularJS.
- Help in the deployment of the web-app using Docker and Kubernetes.
- Attend meetings regularly through Microsoft Teams.
- Meet deadlines and be available whenever required by the team.
- Ensure that the Backend Tasks are completed on time and ready for smooth integration with Front-end.

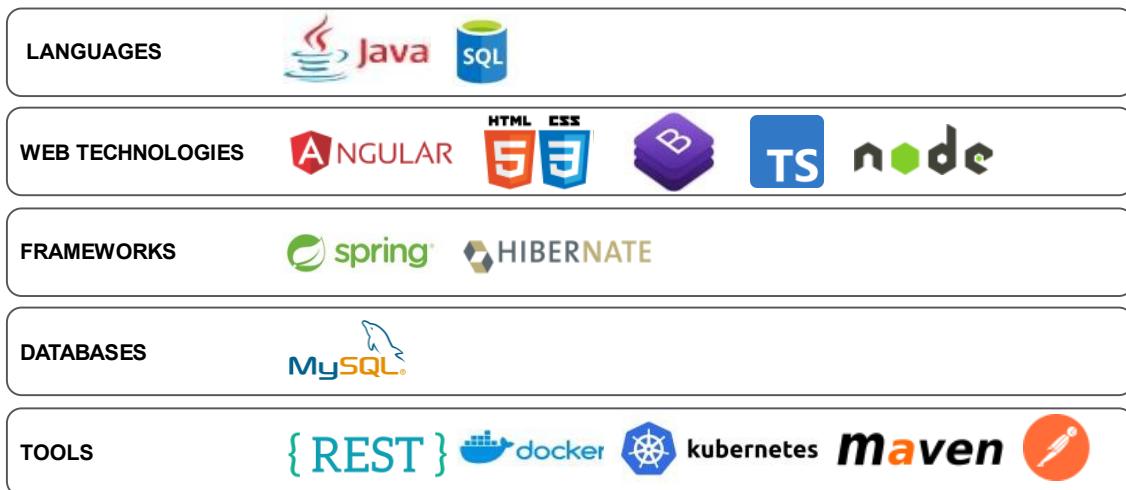
MODULES DEVELOPED

- Spring Rest API for LDAP Login
- Spring Rest API for JWT authorization
- ER Diagrams
- Implemented ER diagrams in MySQL
- Spring Rest API using Hibernate for Projects page and Testcases

- API testing using Postman
- Integration of APIs with Frontend
- Deployment of APIs on Docker and Kubernetes via GitHub

LANGUAGES, FRAMEWORKS, TOOLS AND SERVICES USED

Tech Stack



Confidential property of Optum. Do not distribute or reproduce without express permission from Optum.

5

DEVELOPMENT PHASE

3.11 Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is used as the server-side language for most back-end development projects, including those involving big data and Android development. It's free and open source.

3.12 Spring Boot: Open-Source Java Framework

It is used to create a micro service. It provides infrastructure for developing Java applications.

3.13 MySQL: Database used locally.

MySQL is an open-source relational database management system.

3.14 Hibernate: Object Relational Mapping (ORM) Java Framework

It simplifies the development of Java application to interact with the database. It implements the specifications of Java Persistence API (JPA) for data persistence.

3.15 Rest API: Application Programming Interface

Representational state transfer is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. REST defines a set of constraints for how the architecture of an Internet-scale distributed hypermedia system, such as the Web, should behave

TESTING PHASE

3.21 Git and GitHub:

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning-fast performance. It is used for tracking changes in source code during software

development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

GitHub provides hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration and wikis for every project.

Through this internship, I learnt the version control mechanisms very well. I learnt how to collaborate with the development team and work together on a project. I learnt the ins and outs of Git and the various command line arguments like, push, pull, rebase, merge and other commands used for collaboration and version control.

3.22 Postman

Postman is a great tool to test RESTful APIs I had made. It offers a sleek user interface with which to make HTTP requests, without the hassle of writing a bunch of code just to test an API's functionality.

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so to help create better APIs—faster.

Postman has various features, but I learnt how to test the developed Software (Rest API) on it, using the API Client. It provides a very easy interface to test any HTTP Request—get, post, put, patch, delete or more. A number of custom features are available like setting different body types or

formatting the response.

Postman also helps in documenting the REST APIs for the easy integration in the Frontend. Postman's API documentation tool generates beautiful, machine-readable documentation for the API and automatically keeps it up to date. While testing itself, proper documentation takes place without extra efforts.

I learnt how to use Postman as an API Testing tool very well.

DEPLOYMENT PHASE

3.31 Docker

Container deployment is **a method for quickly building and releasing complex applications**. Docker container deployment is a popular technology that gives developers the ability to construct application environments with speed at scale.

3.32 Kubernetes

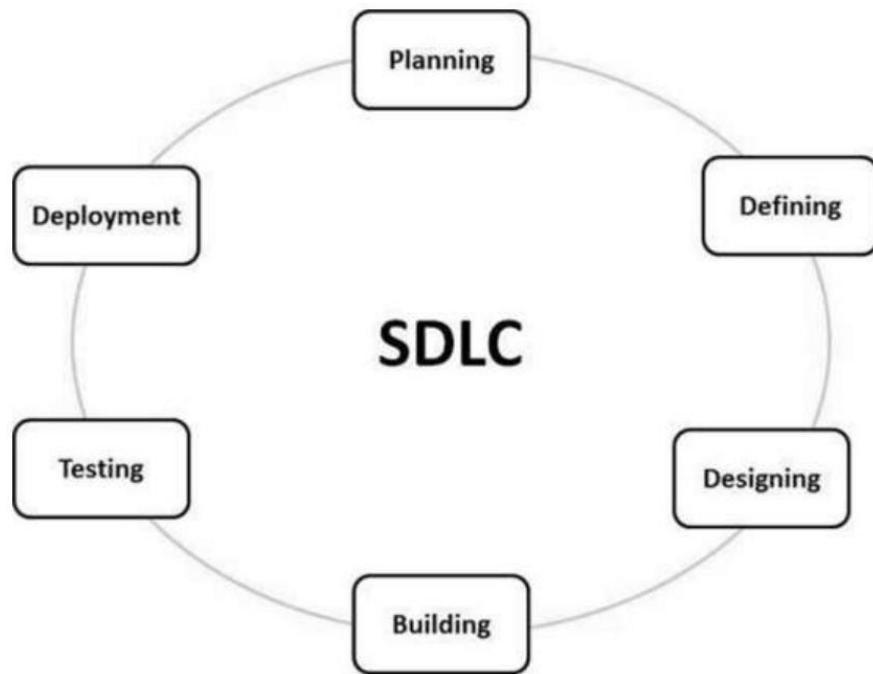
Kubernetes is an **open-source system** which allows you to run containers, manage them, automate deploys, scale deployments, create and configure ingresses, deploy stateless or stateful applications, and many other things.

I went through the basics of Docker and Kubernetes hosting to assist the team during deployment of the application. It was not a major part of my job.

PROCESS OF DEVELOPMENT

SOFTWARE DEVELOPMENT LIFE CYCLE

The complete development of the Software followed the Software Development Life Cycle strictly



4.11 Planning: My mentor planned on what had to be done.

4.12 Defining: On the basis of the plan required, I decided what tools are required

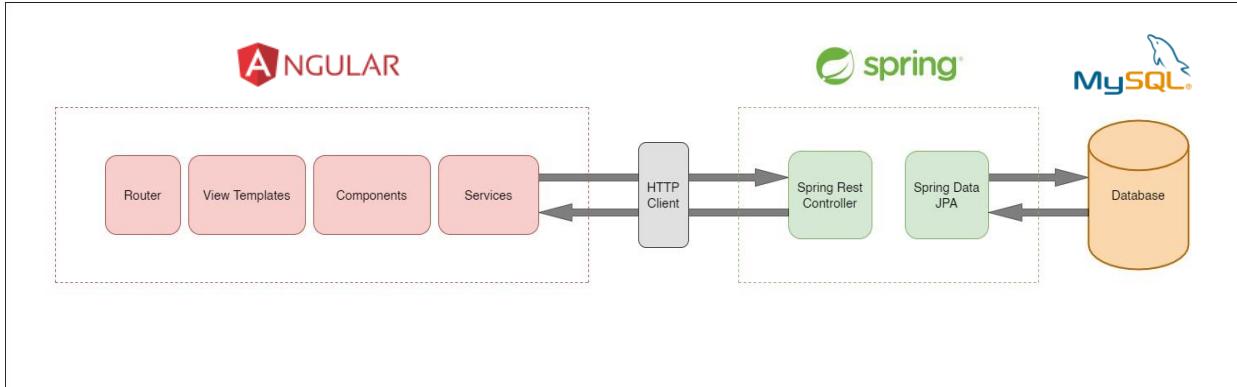
4.13 Designing: I developed the ER model.

4.14 Building: This was the coding phase

4.15 Testing: The APIs were tested using Postman.

ARCHITECTURE

Architecture

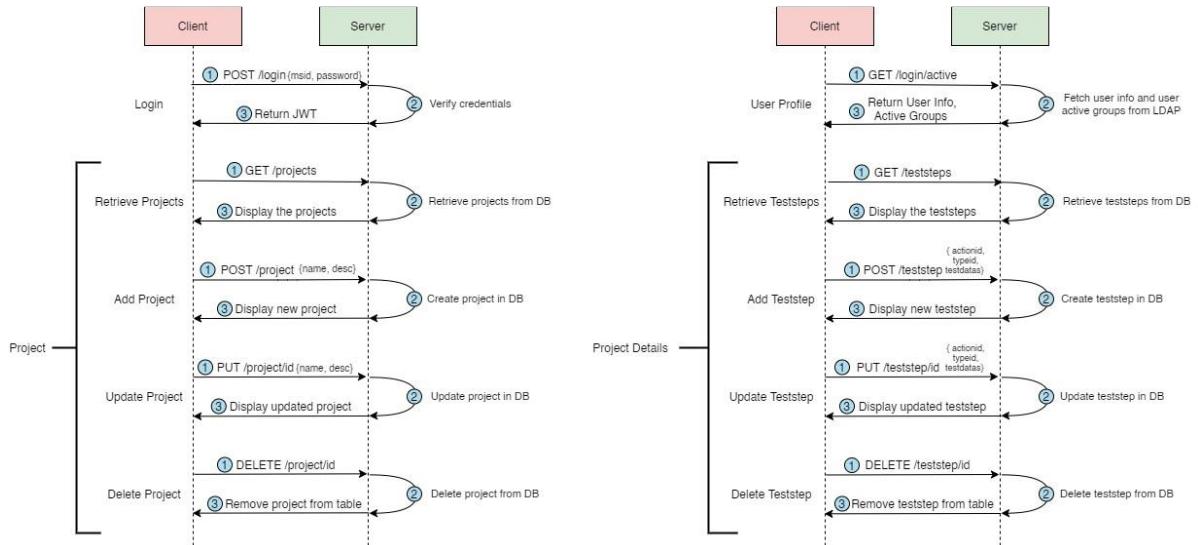


Confidential property of Optum. Do not distribute or reproduce without express permission from Optum.

4

WORKFLOW OF PROJECT

Workflow of the Project



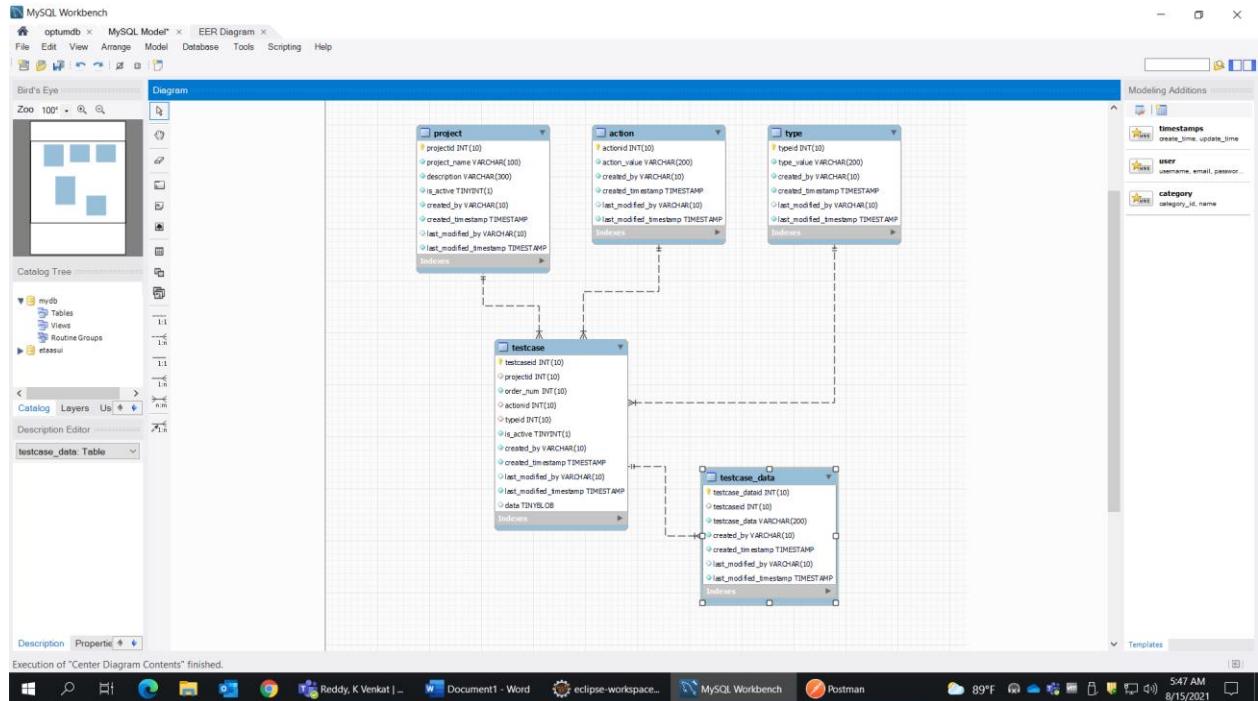
Confidential property of Optum. Do not distribute or reproduce without express permission from Optum.

10

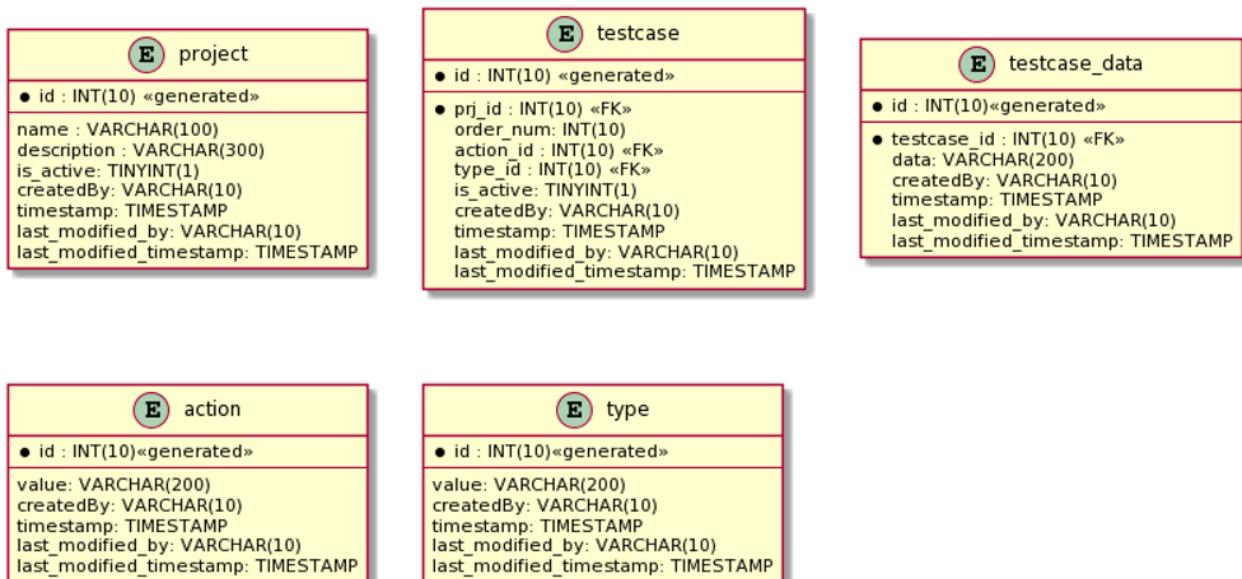
18

DATABASE MANAGEMENT

ER DIAGRAM



The complete ER model was not developed at once. The ER model was developed functionality wise. For example, the Projects model was developed, then the Testcase and Testcase_data models were developed.



Spring Rest API

MEASURES TAKEN

6.1.1 Json Web Tokens

I learnt about JSON WEB TOKENs (JWT) for Token Authentication. JWTs are used for securely transmitting information between parties as a JSON object.

JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA. Signed tokens can verify the integrity of the claims contained within it, while encrypted tokens hide those claims from other parties. When tokens are signed using public/private key pairs, the signature also certifies that only the party holding the private key is the one that signed it.

JWTs contain 3 parts separated by a '.'

- Header – Containing basic
- Payload
- Signature

I used SHA256 to digitally sign this signature.

6.1.2 Programming Measures

- The code was developed using Object Oriented Programming Concepts like Classes, Objects and Inheritance.
- Proper Functions were used to handle the different http requests like GET, POST, etc.

- The code was developed in such a way that it does not throw any arbitrary error.
- Proper Error Handling was used.
- Try - Except blocks were used in almost the complete code to minimize chances of error.
- Proper Status Codes were sent as Response.

```

eclipse-workspace - ETAAS_API/src/main/resources/application.properties - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer Testcase.java TestcaseData... Type.java ActionRepos... ProjectRepo... TestcaseData... TestcaseRepo... TypeRepo...
application... Data Source Configuration, URL, UserName and Password for the Database
1 # Data Source Configuration, URL, UserName and Password for the Database
2 spring.datasource.url = jdbc:mysql://rn000037817:3306/etaasui
3 spring.datasource.username = aestools
4 spring.datasource.password = 5DxOp[b<
5
6 # Hibernate Properties
7 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQLInnoDBDialect
8
9 # Hibernate ddl auto property
10 spring.jpa.hibernate.ddl-auto = update
11
12 spring.jpa.show-sql = true
13
14 logging.level.org.hibernate.SQL = DEBUG
15 logging.level.org.hibernate.type = TRACE

```

Markers Java Application Servers Data Source Explorer Snippets Console

Application (1) Java Application C:\Program Files\AdoptOpenJDK\jdk-8.0.282.8-hotspot\bin\javaw.exe (Aug 15, 2021, 4:54:56 AM)

```

2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration materialized_nlob -> org.hib...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration serializable -> org.hib...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration object -> org.hibernate.t...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration java.lang.Object -> org.h...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration imm_date -> org.hibernate...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration imm_time -> org.hibernate...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration imm_timestamp -> org.hib...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration imm_datetime -> org.hil...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration imm_calendar_date -> org.h...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration imm_binary -> org.hibernate...
2021-08-15 05:40:56.656 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration imm_serializable -> org.h...
2021-08-15 05:40:56.998 INFO 11620 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-3 - Starting...
2021-08-15 05:40:56.998 INFO 11620 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-3 - Start completed.
2021-08-15 05:40:56.916 DEBUG 11620 --- [ restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.My...
2021-08-15 05:40:57.009 DEBUG 11620 --- [ restartedMain] o.h.type.spi.TypeConfigurationsScope : Scoping TypeConfiguration [org.hibernate.type.spi.TypeConfiguration]
2021-08-15 05:40:57.484 INFO 11620 --- [ restartedMain] o.h.e.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.l...
2021-08-15 05:40:57.484 TRACE 11620 --- [ restartedMain] o.h.e.j.p.i.JtaPlatformInitiator : Handling #sessionFactoryCreated from [org.hibernate...
2021-08-15 05:40:57.484 INFO 11620 --- [ restartedMain] j.localContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persisten...
2021-08-15 05:40:57.484 WARN 11620 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. The ...
2021-08-15 05:40:58.035 INFO 11620 --- [ restartedMain] o.s.b.d.a.OptionalLocalLoadServer : LiveReload server is running on port 35729
2021-08-15 05:40:58.035 INFO 11620 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context...
2021-08-15 05:40:58.063 INFO 11620 --- [ restartedMain] com.orojectdetails.Application : Started Application in 3.054 seconds (JVM running: 8.152)

```

The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - ETAAS_API/src/main/resources/application.properties - Eclipse". The main window displays the application logs for a Spring Boot application. The logs show the application starting up, bootstrapping Spring Data JPA repositories, and initializing Tomcat. Numerous DEBUG-level log entries from org.hibernate.type.BasicTypeRegistry are visible, detailing the registration of various Java types to their corresponding Hibernate types.

```

2021-08-15 05:40:55.078 INFO 11620 --- [ restartedMain] com.projectdetails.Application      : Starting Application using Java 1.8.0_282 on vetsvra00015697 with PID 11620 (C:\Users\pbhardw\ETAAS_API\src\main\resources\application.properties)
2021-08-15 05:40:55.078 INFO 11620 --- [ restartedMain] com.projectdetails.Application      : No active profile set, falling back to default profiles: default
2021-08-15 05:40:55.473 INFO 11620 --- [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2021-08-15 05:40:55.513 INFO 11620 --- [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 42 ms. Found 5 JPA repository interfaces.
2021-08-15 05:40:56.382 INFO 11620 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2021-08-15 05:40:56.388 INFO 11620 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-08-15 05:40:56.388 INFO 11620 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.48]
2021-08-15 05:40:56.552 INFO 11620 --- [ restartedMain] o.a.c.c.[Tomcat].localhost:[]/ : Initializing Spring embedded WebApplicationContext
2021-08-15 05:40:56.552 INFO 11620 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1464 ms
2021-08-15 05:40:56.560 DEBUG 11620 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000396: Processing PersistenceUnitInfo [name: default]
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration boolean -> org.hibernate.type.BooleanType@1eb05cf0
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration java.lang.Boolean -> org.hibernate.type.BooleanType@1eb05cf0
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration numeric_boolean -> org.hibernate.type.NumericBooleanType@44a157a1
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration true_false -> org.hibernate.type.TrueFalseType@1de84e7
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration yes_no -> org.hibernate.type.YesNoType@68da57a3
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration byte -> org.hibernate.type.ByteType@26984f84
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration java.lang.Byte -> org.hibernate.type.ByteType@26984f84
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration character -> org.hibernate.type.CharacterType@2c158a62
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration char -> org.hibernate.type.CharacterType@2c158a62
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration java.lang.Character -> org.hibernate.type.CharacterType@2c158a62
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration short -> org.hibernate.type.ShortType@8852657
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration short -> org.hibernate.type.ShortType@8852657
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration integer -> org.hibernate.type.IntegerType@e1750b9
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration int -> org.hibernate.type.IntegerType@e1750b9
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration java.lang.Integer -> org.hibernate.type.IntegerType@7e1750b9
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration long -> org.hibernate.type.LongType@60001b52
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration long -> org.hibernate.type.LongType@60001b52
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration float -> org.hibernate.type.FloatType@671151f2
2021-08-15 05:40:56.640 DEBUG 11620 --- [ restartedMain] org.hibernate.type.BasicTypeRegistry : Adding type registration float -> org.hibernate.type.FloatType@671151f2

```

REST API DEVELOPMENT

As mentioned earlier, the APIs were tested on Postman. About 120+ APIs were developed by me. I mainly developed the APIs for the School Entity. While, the rest of the team developed it for Tutor and Institution.

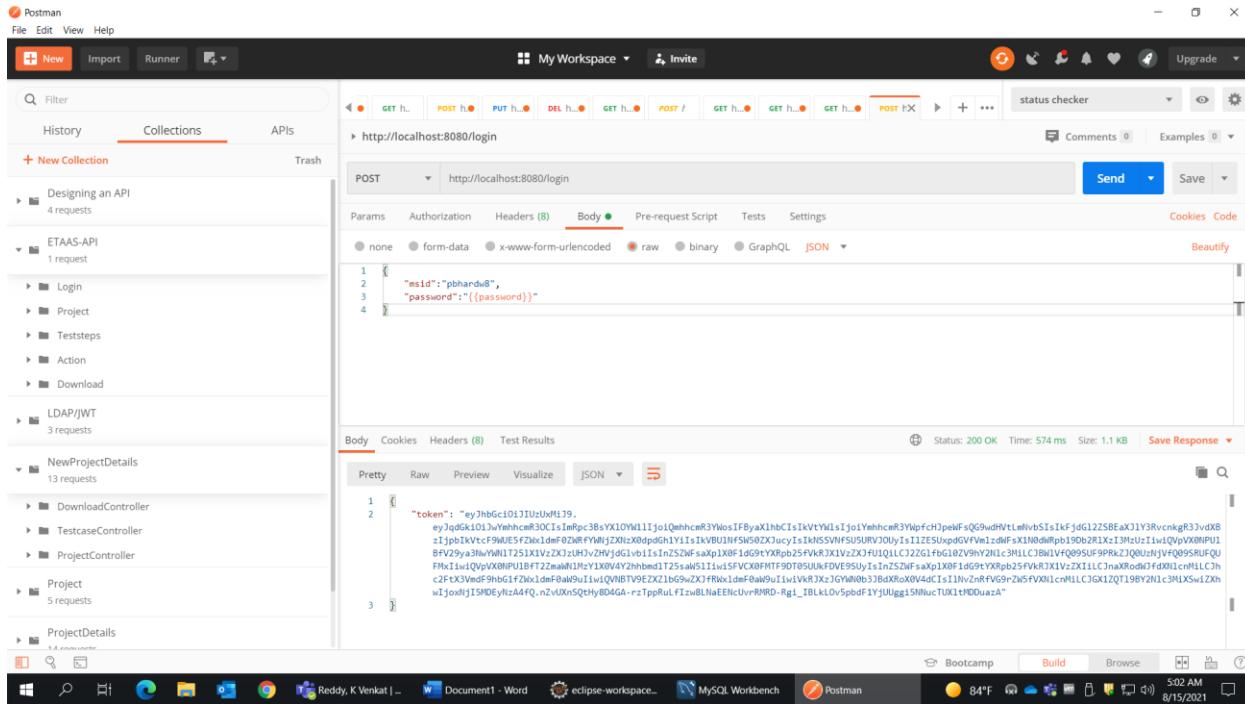
The HTTP methods used were post, get, delete, put and patch. The response of the APIs are in JSON format.

The main functionalities that I have developed have been mentioned above.

REST API TESTING

Some of the tests of the REST APIs are present below

Login post

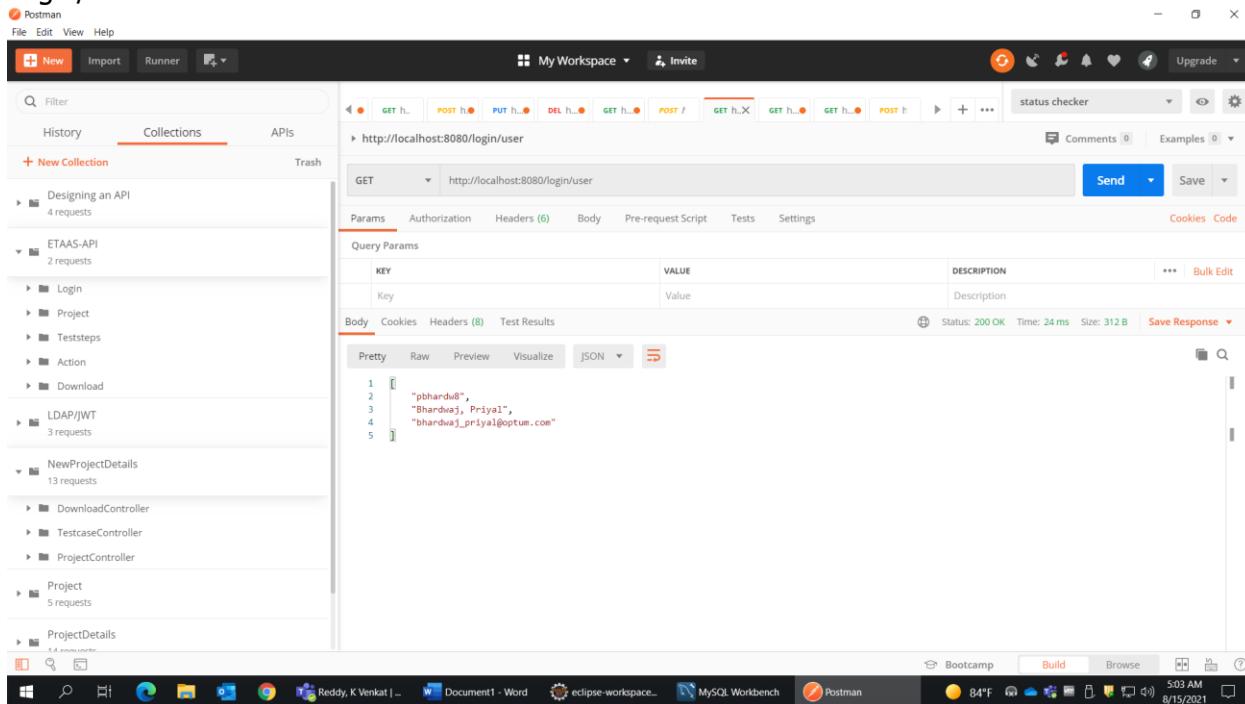


The screenshot shows the Postman interface with a collection named "Designing an API". A POST request is made to `http://localhost:8080/login`. The request body is set to "raw" JSON:

```
1 {
2   "msid": "pbhardw@",
3   "password": "({password})"
4 }
```

The response status is 200 OK, and the token is displayed in the response body.

Login/user

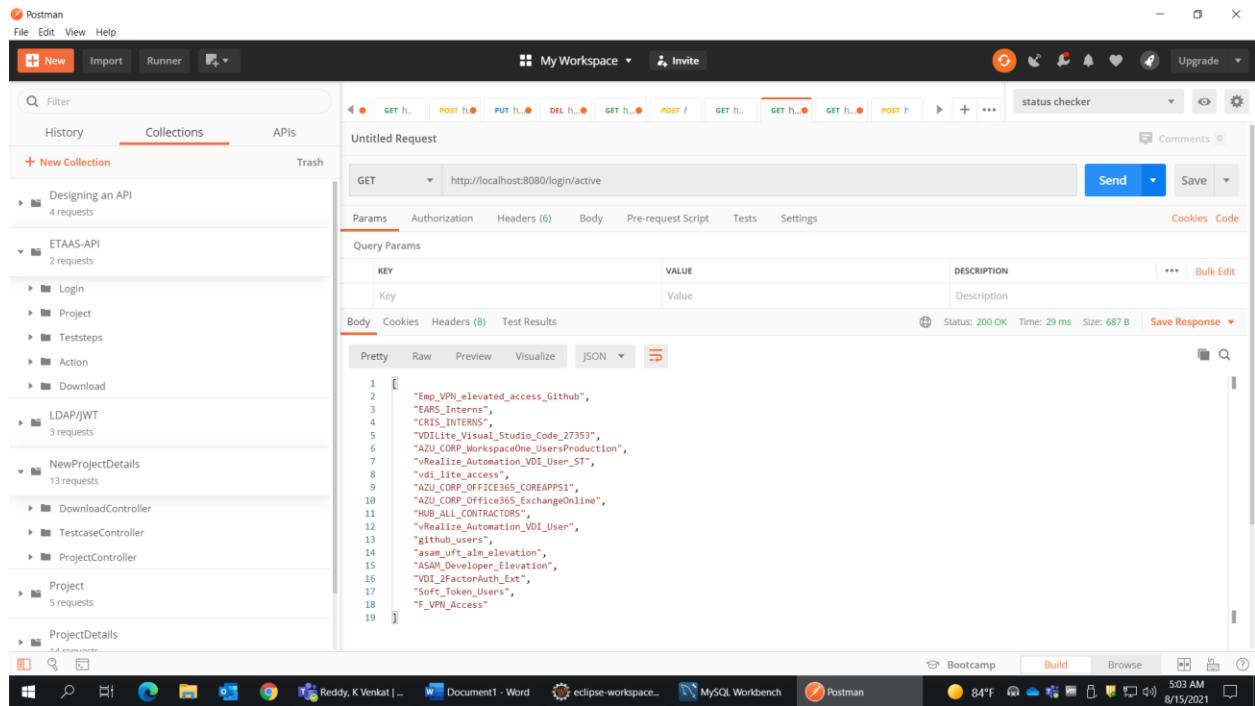


The screenshot shows the Postman interface with a collection named "Designing an API". A GET request is made to `http://localhost:8080/login/user`. The request includes query parameters:

KEY	VALUE	DESCRIPTION
Key	Value	Description

The response status is 200 OK, and the JSON object is displayed in the response body.

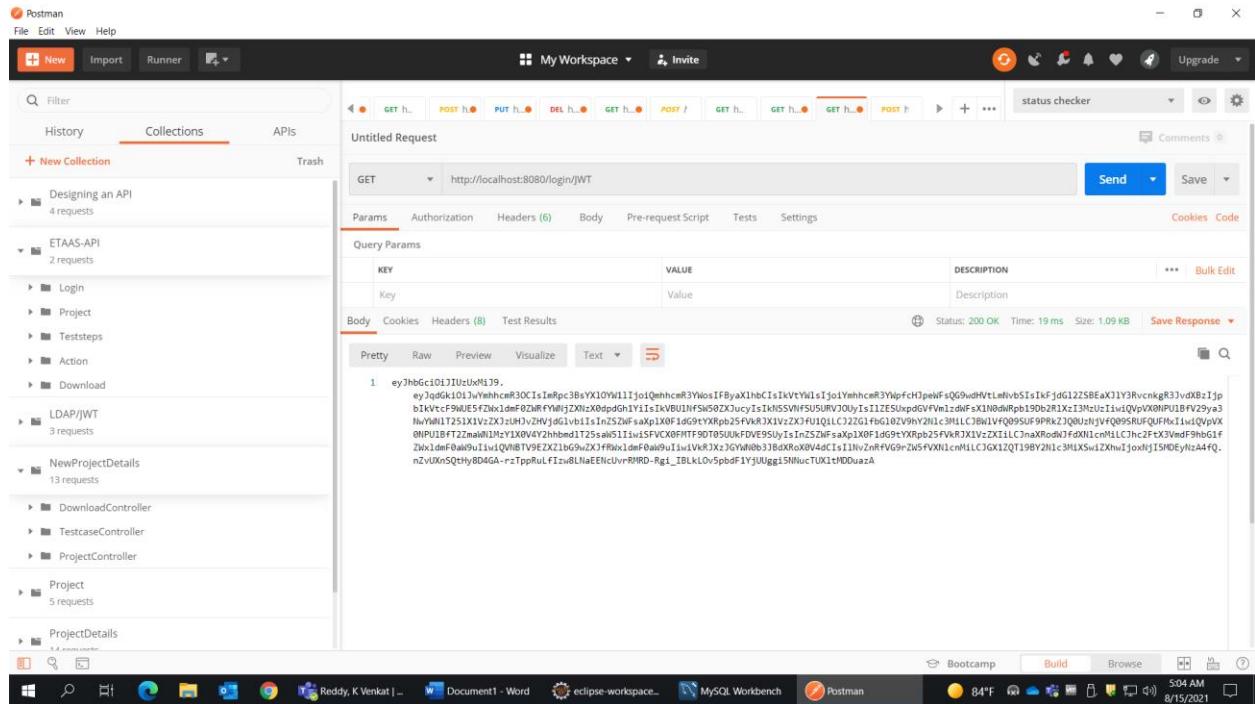
Login/active



The screenshot shows the Postman application interface. On the left, there's a sidebar with collections like 'Designing an API', 'ETAA-AP1', 'LDAP/JWT', and 'NewProjectDetails'. The main area shows an 'Untitled Request' with a GET method to 'http://localhost:8080/login/active'. The response body is a JSON object with many keys, some of which are highlighted in red:

```
1 "Emp_VPN_elevated_access_Github",
2 "EARS_Interns",
3 "CRIS_INTERNALS",
4 "AZU_Corp_EspaceOne_UsersProduction",
5 "AZU_Corp_EspaceOne_VDI_User_ST",
6 "AZU_Lite_Access",
7 "AZU_Corp_Office365_COREAPP$1",
8 "AZU_Corp_Office365_ExchangeOnline",
9 "HUB_ALL_CONTACTORS$",
10 "vRealize_Automation_VDI_User",
11 "github_users",
12 "asam_uft_aim_elevation",
13 "ASAM_Developer_Elevation",
14 "VDI_2FactorAuth_Ext",
15 "Soft_Token_Users",
16 "F_VPN_Access"
```

Login/JWT



The screenshot shows the Postman application interface. The sidebar has the same collection structure as the previous screenshot. The main area shows an 'Untitled Request' with a GET method to 'http://localhost:8080/login/JWT'. The response body is a large JSON object containing a JWT token, which is partially visible:

```
1 eyJhbGciOiJIUzI1NiJ9.
eyJqdGkiOiJyYmhhcmR3OCIsImRpC3BsYXLOW1lIjo1QemhhcmR3YkNfQdhVtLmIvbSIsIkfJdG12ZSEaXj1Y3RvcnkR3jvdxBzIjp
b1kYcfcF9wUEF2Nx3dmF02RIrYmijZXOx0dpdGh1Y1IsIkVBUHfSw5Q2XJu2c1iKNSVfUSU5URV0U1s11ZE5pxdgVw12wf>xINodwRpb19b2rIx1z3MzUz1w1QvpxX0PUbV29ya3
NyYmU1T251X1Vz2X1zHv2HjG1vb1s1n5z2W>iAxp1X0f1g9jYRpb2p8VFRX1VzXfU1Q1L122Gfb182V9HY2NL13HL1CjB1WVf095UF0PRk2zQ0uJvFQ095RUFQfHx1w1lQvpX
0fPU1BfT2zmaW1lM1Y1XW4Yzhbm1725a0w511iSFVX0PMT90705UkF0DE95Uy1s1n5z2W>faXp1X0f1g9jYRpb25FVRX1VzXfU1naRwd7FdK11cn1L1Ch2F>x3wdf9bG1f
2Xh1dmF0awu11w1QWb1T9E2X1b69wXfRwx1dmf0au9u11w1vKxJx2jOGWb0j38dKRxx04dC1s11w1ZnfEVG9rZw5fVXN1cmHLLCjZK12Q19BY2H1c3M1XSw1ZKhw1joxhj1SM0EylzA4fQ.
nZv0Xn0SQthHyB04GA-rzTpRuLF1zw8LNaeERcvrRMHD-Rg1_IBLkLoV5pdF1Yj0Ugg15NuvtUX1tH00uaazA
```

Actions

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' selected, displaying various API endpoints under categories like 'Designing an API', 'ETAS-API', 'Login', 'Project', 'Teststeps', 'Action', 'Download', 'LDAP/JWT', 'NewProjectDetails', and 'DownloadController'. The main workspace shows a 'GET' request to 'http://localhost:8080/actions'. The response body is a JSON array:

```

1 [
2   {
3     "value": "Launch_Browser",
4     "key": 1
5   },
6   {
7     "value": "Click",
8     "key": 2
9   },
10  {
11    "value": "SetValue",
12    "key": 3
13  },
14  {
15    "value": "send Post Request",
16    "key": 4
17  },
18  {
19    "value": "Close Browser",
20    "key": 5
21 },

```

Types

The screenshot shows the Postman application interface. The sidebar is identical to the previous one, showing various API endpoints. The main workspace shows a 'GET' request to 'http://localhost:8080/types'. The response body is a JSON array:

```

1 [
2   {
3     "value": "UI",
4     "key": 1
5   },
6   {
7     "value": "API",
8     "key": 2
9   }
10 ]

```

Postman

File Edit View Help

New Import Runner + My Workspace Invite Upgrade

Filter Collections APIs + New Collection History Trash

ETAAS-API 7 requests

Login

POST http://localhost:8080/login
GET http://localhost:8080/login/user
GET http://localhost:8080/login/JWT
GET http://localhost:8080/login/active

Project Teststeps Action Download LDAP/JWT NewProjectDetails DownloadController

GET http://localhost:8080/generate/35

GET http://localhost:8080/actions
GET http://localhost:8080/types

Download GET http://localhost:8080/generate/35

LDAP/JWT 3 requests

NewProjectDetails 13 requests

DownloadController

PUT http://localhost:8080/generate/35

My Workspace Invite status checker

Comments Examples Save

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize Text

```

1 package com.app.tests.utest;
2
3 import org.testng.annotations.Test;
4
5 import com.app.base.BaseTestCase;
6
7 public class TestCase extends BaseTestCase{
8
9     @Test(enabled=true, groups = { "utest" },
10          description="Validate user with valid credentials is able to login application")
11    public void login_checkDelete() {
12        uiLaunch("http://www.google.com");
13        uiClick("form");
14        uiSetValue("name","xyz");
15        api_sendPostRequest("sendPost1","sendPost2","sendPost3","sendPost4");
16        uiCloseDriver();
17    }
18 }
19

```

Status: 200 OK Time: 147 ms Size: 917 B Save Response

Bootcamp Build Browse

83°F 5:12 AM 8/15/2021

Projects

Postman

File Edit View Help

New Import Runner + My Workspace Invite Upgrade

Filter Collections APIs + New Collection History Trash

Designing an API 4 requests

ETAAS-API 8 requests

Login

POST http://localhost:8080/login
GET http://localhost:8080/login/user
GET http://localhost:8080/login/JWT
GET http://localhost:8080/login/active

Project Teststeps Action Download LDAP/JWT NewProjectDetails DownloadController

GET http://localhost:8080/actions
GET http://localhost:8080/types

Download GET http://localhost:8080/generate/35

LDAP/JWT 3 requests

NewProjectDetails 35 requests

PUT http://localhost:8080/projects

My Workspace Invite status checker

Comments Examples Save

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2     {
3         "createdDate": "2021-07-26T07:52:11.000+00:00",
4         "lastModifiedDate": "2021-07-26T07:52:11.000+00:00",
5         "projectId": 27,
6         " projectName": "Project 10",
7         "description": "Project Details 10",
8         "active": false
9     },
10    {
11        "createdDate": "2021-07-26T09:01:06.000+00:00",
12        "lastModifiedDate": "2021-07-26T09:01:06.000+00:00",
13        "projectId": 34,
14        " projectName": "Project 16",
15        "description": "Project Details 16",
16        "active": false
17    },
18    {
19        "createdDate": "2021-07-27T00:16:36.000+00:00",
20        "lastModifiedDate": "2021-07-27T00:16:36.000+00:00",
21        "projectId": 35,

```

Status: 200 OK Time: 423 ms Size: 1.37 KB Save Response

Bootcamp Build Browse

83°F 5:14 AM 8/15/2021

Project/id

Postman

File Edit View Help

New Import Runner + My Workspace Invite Upgrade status checker

Filter History Collections APIs + New Collection Trash

Designing an API 4 requests

ETAAS-API 9 requests

Login

POST http://localhost:8080/login

GET http://localhost:8080/login/user

GET http://localhost:8080/login/jwt

GET http://localhost:8080/login/active

Project

GET http://localhost:8080/projects

GET http://localhost:8080/project/35

Teststeps

Action

Download

LDAP/JWT

GET http://localhost:8080/generate/35

GET http://localhost:8080/project/35

Send Save

Comments Examples Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2     "createdDate": "2021-07-27T00:16:36.000+00:00",
3     "lastModifiedDate": "2021-07-27T00:16:36.000+00:00",
4     "projectId": 35,
5     "projectName": "Project 3",
6     "description": "Project Details 3",
7     "active": false
8 }
```

Status: 200 OK Time: 53 ms Size: 441 B Save Response

Bootcamp Build Browse

83°F 5:16 AM 8/15/2021

Project/id/teststeps

Postman

File Edit View Help

New Import Runner + My Workspace Invite Upgrade status checker

Filter History Collections APIs + New Collection Trash

Designing an API 4 requests

ETAAS-API 10 requests

Login

POST http://localhost:8080/login

GET http://localhost:8080/login/user

GET http://localhost:8080/login/jwt

GET http://localhost:8080/login/active

Project

GET http://localhost:8080/projects

GET http://localhost:8080/project/35

GET http://localhost:8080/project/35/teststeps

Teststeps

Action

Download

GET http://localhost:8080/generate/35

GET http://localhost:8080/project/35/teststeps

Send Save

Comments Examples Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

2 {
3     "createdDate": "2021-07-27T00:19:04.000+00:00",
4     "lastModifiedDate": "2021-07-27T00:19:04.000+00:00",
5     "testcaseId": 11,
6     "ordenum": 1,
7     "testcasedata": [
8         {
9             "createdDate": "2021-07-27T00:23:17.000+00:00",
10            "lastModifiedDate": "2021-07-27T00:23:17.000+00:00",
11            "testcaseId": 15,
12            "testData": "http://www.google.com",
13            "testcaseid": 11
14        }
15    ],
16    "projectId": 35,
17    "actionId": 1,
18    "typeId": 1,
19    "active": false
20 },
21 {
22     "createdDate": "2021-07-27T00:20:37.000+00:00",
23     "lastModifiedDate": "2021-07-27T00:20:37.000+00:00",
}
```

Status: 200 OK Time: 162 ms Size: 2.95 KB Save Response

Bootcamp Build Browse

83°F 5:16 AM 8/15/2021

Project post

The screenshot shows the Postman application interface. On the left, the sidebar lists collections and APIs, including 'Designing an API', 'ETAA5-API' (11 requests), 'Login' (4 requests), 'Project' (4 requests), 'Teststeps' (1 request), 'Action' (2 requests), and 'Download' (1 request). The main workspace shows a POST request to 'http://localhost:8080/project'. The 'Body' tab is selected, showing the following JSON payload:

```

1 {
2   "projectName": "Project Report",
3   "description": "ETAA5 Report Format"
4 }

```

The response status is 200 OK, with a time of 259 ms and a size of 447 B. The response body is displayed as:

```

1 {
2   "createdDate": "2021-08-14T23:47:40.649+00:00",
3   "lastModifiedDate": "2021-08-14T23:47:40.649+00:00",
4   "projectId": 0,
5   " projectName": "Project Report",
6   "description": "ETAA5 Report Format",
7   "active": false
8 }

```

The screenshot shows the MySQL Workbench application interface. The left sidebar includes sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). The central area has a 'sample SQL*' editor window with the following query:

```

171
172 • select * from project;
173
174
175
176
177
178
179
180
181
182
183
184
185
186

```

Below the editor is a 'Result Grid' window displaying the results of the query:

projectid	project_name	description	is_active	created_by	created_timestamp	last_modified_by	last_modified_timestamp
27	Project 10	Project Details 10	0	aestools	2021-07-26 13:22:11	aestools	2021-07-26 13:22:11
34	Project 16	Project Details 16	0	aestools	2021-07-26 14:31:06	aestools	2021-07-26 14:31:06
35	Project 3	Project Details 3	0	aestools	2021-07-27 05:46:36	aestools	2021-07-27 05:46:36
46	Project 4	Project Details 4	0	aestools	2021-07-30 10:12:02	aestools	2021-07-30 10:12:02
71	Project Test	Project Details Test	0	aestools	2021-08-09 00:11:08	aestools	2021-08-09 00:47:29
73	Project 5	Project Details 5	0	aestools	2021-08-09 00:13:27	aestools	2021-08-09 00:13:27
77	Project Report	ETAA5 Report Format	0	aestools	2021-08-15 05:17:41	aestools	2021-08-15 05:17:41

The right side of the interface shows a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.' The bottom status bar shows the date and time as 5:18 AM 8/15/2021.

Project/id put

Postman

File Edit View Help

New Import Runner + My Workspace Invite Upgrade

Filter Collections APIs + New Collection History Trash

PUT http://localhost:8080/project/77

Params Authorization Headers (8) Body (8) Pre-request Script Tests Settings

Body (8) Cookies Headers (8) Test Results

```

1 {
2   "projectName": "Project Report",
3   "description": "ETAAS Report Format Changes"
4 }
    
```

Status: 200 OK Time: 87 ms Size: 456 B Save Response

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "createdDate": "2021-08-14T23:47:41.000+00:00",
3   "lastModifiedDate": "2021-08-14T23:49:20.041+00:00",
4   "projectId": 77,
5   "projectName": "Project Report",
6   "description": "ETAAS Report Format Changes",
7   "active": false
8 }
    
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize Text

Bootcamp Build Browse 8:19 AM 8/15/2021

Project/id delete

Postman

File Edit View Help

New Import Runner + My Workspace Invite Upgrade

Filter Collections APIs + New Collection History Trash

DELETE http://localhost:8080/project/77

Params Authorization Headers (8) Body (8) Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize Text

1

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize Text

Bootcamp Build Browse 8:20 AM 8/15/2021

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane lists various database objects like MANAGEMENT, INSTANCE, and PERFORMANCE. The central area contains a SQL editor with the following query:

```
171
172 • select * from projects;
173
174
175
176
177
178
179
180
181
182
183
184
185
186
```

The Result Grid displays the following data:

project_id	project_name	description	is_active	created_by	created_timestamp	last_modified_by	last_modified_timestamp
27	Project 10	Project Details 10	0	aestools	2021-07-26 13:22:11	aestools	2021-07-26 13:22:11
34	Project 16	Project Details 16	0	aestools	2021-07-26 14:31:06	aestools	2021-07-26 14:31:06
35	Project 3	Project Details 3	0	aestools	2021-07-27 05:46:36	aestools	2021-07-27 05:46:36
46	Project 4	Project Details 4	0	aestools	2021-07-30 10:12:02	aestools	2021-07-30 10:12:02
71	Project Test	Project Details Test	0	aestools	2021-08-09 00:11:08	aestools	2021-08-09 00:12:29
73	Project 5	Project Details 5	0	aestools	2021-08-09 00:13:27	aestools	2021-08-09 00:13:27

On the right side of the interface, there is a note: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below the interface, the taskbar shows icons for Postman, MySQL Workbench, and other applications.

Teststeps/projectid

The screenshot shows the Postman application interface. The left sidebar displays a collection named "ETAS-API" containing several requests under categories like "Login", "Project", and "Teststeps". The main workspace shows a "Teststeps" collection with a single GET request to "http://localhost:8080/teststeps/35". The "Body" tab of the request configuration is selected, showing a JSON response:

```

1 [
2   {
3     "testcaseid": 11,
4     "action_value": "Launch_Browser",
5     "Data": [
6       {
7         "Data": [
8           "http://www.google.com"
9         ],
10        "order_num": 1,
11        "type_value": "UI"
12      },
13      {
14        "testcaseid": 12,
15        "action_value": "Click",
16        "Data": [
17          {
18            "form": [
19              {
20                "order_num": 2,
21                "type_value": "UI"
22              }
23            ]
24          }
25        ]
26      }
27    ]
28  ]
29 }
```

The status bar at the bottom indicates "Status: 200 OK Time: 197 ms Size: 889 B Save Response". The taskbar at the bottom shows icons for Postman, MySQL Workbench, and other applications.

Teststep post

Postman

File Edit View Help

New Import Runner + My Workspace Invite Upgrade status checker Comments Examples

Filter Collections APIs

+ New Collection

ETAA-API 15 requests

Login

POST http://localhost:8080/login

GET http://localhost:8080/login/user

GET http://localhost:8080/login/JWT

GET http://localhost:8080/login/active

Project

GET http://localhost:8080/projects

GET http://localhost:8080/project/35/teststeps

POST http://localhost:8080/project

PUT http://localhost:8080/project/77

DEL http://localhost:8080/project/77

Teststeps

GET http://localhost:8080/teststeps/35

POST http://localhost:8080/teststep

Action

GET http://localhost:8080/actions

POST http://localhost:8080/teststep

Body Headers (8) Test Results

Params Authorization Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2   "projectId": 73,
3   "ordenum": 1,
4   "actionid": 4,
5   "typeid": 1,
6   "data": ["Dummy data 1","Dummy data 2"]
7

```

Status: 200 OK Time: 102 ms Size: 456 B Save Response

Bootcamp Build Browse

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

sample SQL*

select * from testcase;

Result Grid

testcased	projectid	ordenum	actionid	typeid	is_active	created_by	created_timestamp	last_modified_by	last_modified_timestamp	data
11	35	1	1	0	aestools	2021-07-27 05:49:04	aestools		2021-07-27 05:49:04	
12	35	2	2	1	0	aestools	2021-07-27 05:50:37	aestools	2021-07-27 05:50:37	
13	35	5	3	1	0	aestools	2021-07-27 05:50:37	aestools	2021-08-08 19:05:05	
14	35	4	4	2	0	aestools	2021-07-27 05:50:37	aestools	2021-07-27 05:50:37	
64	35	5	2	2	0	aestools	2021-07-30 10:28:08	aestools	2021-07-30 10:28:08	
70	27	1	1	1	0	aestools	2021-08-02 03:49:37	aestools	2021-08-02 03:49:37	
71	27	5	30	1	0	aestools	2021-08-02 03:49:37	aestools	2021-08-10 15:50:41	
72	27	3	3	1	0	aestools	2021-08-02 03:49:37	aestools	2021-08-02 03:49:37	
73	27	4	4	2	0	aestools	2021-08-02 03:49:37	aestools	2021-08-02 03:49:37	
91	27	5	2	1	0	aestools	2021-08-03 19:21:06	aestools	2021-08-03 19:21:06	
95	35	5	5	1	0	aestools	2021-08-03 21:37:31	aestools	2021-08-03 21:37:31	
145	27	5	2	1	0	aestools	2021-08-08 21:37:12	aestools	2021-08-08 21:37:12	
146	27	5	2	1	0	aestools	2021-08-08 21:40:09	aestools	2021-08-08 21:40:09	
147	46	5	2	1	0	aestools	2021-08-08 21:57:09	aestools	2021-08-08 21:57:09	
149	46	5	3	1	0	aestools	2021-08-08 21:58:09	aestools	2021-08-08 21:58:09	
150	46	5	3	1	0	aestools	2021-08-08 22:53:07	aestools	2021-08-08 22:53:07	
186	46	5	5	1	0	aestools	2021-08-08 22:54:34	aestools	2021-08-08 22:54:34	
215	73	1	4	1	0	aestools	2021-08-15 05:26:14	aestools	2021-08-15 05:26:14	

testcase 5

Output

Action Output

- Time Action
- 4 05:26:58 select * from testcase_data LIMIT 0, 1000
- 5 05:31:56 select * from testcase_data LIMIT 0, 1000
- 6 05:33:53 select * from testcase LIMIT 0, 1000

Message Duration / Fetch

- 34 row(s) returned 0.000 sec / 0.000 sec
- 34 row(s) returned 0.000 sec / 0.000 sec
- 18 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

The screenshot shows the MySQL Workbench interface. In the center, there is a 'Result Grid' window titled 'sample SQL*'. It displays a table with columns: testcase_idat, testcased, testcase_data, created_by, created_timestamp, last_modified_by, and last_modified_timestamp. The data consists of multiple rows of test case entries. To the right of the grid, a vertical toolbar has several icons: 'Form Editor', 'Field Types', 'Query Test', and 'Execution Plan'. A message box on the right says: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.' At the bottom of the window, there are buttons for 'Apply', 'Revert', 'Context Help', and 'Snippets'. Below the main window, the taskbar shows icons for Postman, MySQL Workbench, and other applications.

Teststep/id put

The screenshot shows the Postman application interface. On the left, the 'Collections' tab is selected, showing a list of collections like 'Designing an API', 'ETaaS-API', and 'Login'. In the center, a 'PUT' request is being made to 'http://localhost:8080/teststep/215'. The request body contains the following JSON:

```

1 {
2     "projectId": 73,
3     "ordenum": 1,
4     "actionId": 4,
5     "typeId": 1,
6     "Data": ["Test Data 1", "Test Data 2"]
7 }

```

Below the request, the 'Body' tab shows the JSON response received from the server. The response is identical to the request body. The status bar at the bottom indicates: Status: 200 OK, Time: 176 ms, Size: 777 B, Save Response.

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database 'optimedb' is selected. The main area displays a query results grid for a table named 'testcase_data'. The grid has columns: testcase_dated, testcased, testcase_data, created_by, created_timestamp, last_modified_by, and last_modified_timestamp. The results show multiple rows of data, with the last two rows highlighted in blue: 'Test Data 1' and 'Test Data 2'. The status bar at the bottom right shows the date and time as '8/15/2021 5:32 AM' and the temperature as '83°F'.

Teststep/id delete

The screenshot shows the Postman application interface. On the left, the 'Collections' tab is selected, displaying a list of API endpoints under the 'ETTAS-API' collection. In the center, a 'DELETE' request is being made to the URL 'http://localhost:8080/teststep/215'. The 'Body' tab shows a single parameter 'key' with the value 'Test Data 2'. The response status is '200 OK' with a duration of '79 ms'. The status bar at the bottom right shows the date and time as '8/15/2021 5:35 AM' and the temperature as '89°F'.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: sample SQL*+ x

MANAGEMENT

- Server Status
- Client Connections
- User and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Log
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas Information

No object selected

testcase 7 x

Output

Action Output

- # Time Action
- 6 05:33:53 select * from testcase LIMIT 0, 1000
- 7 05:35:57 select * from testcase LIMIT 0, 1000
- 8 05:36:19 select * from testcase LIMIT 0, 1000

Message

18 row(s) returned

17 row(s) returned

17 row(s) returned

Duration / Fetch

0.000 sec / 0.000 sec

0.000 sec / 0.000 sec

0.000 sec / 0.000 sec

Object Info Session

Teststep/projectid delete

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: sample SQL*+ x

MANAGEMENT

- Server Status
- Client Connections
- User and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Log
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas Information

No object selected

testcase 7 x

Output

Action Output

- # Time Action
- 6 05:33:53 select * from testcase LIMIT 0, 1000
- 7 05:35:57 select * from testcase LIMIT 0, 1000
- 8 05:36:19 select * from testcase LIMIT 0, 1000

Message

18 row(s) returned

17 row(s) returned

17 row(s) returned

Duration / Fetch

0.000 sec / 0.000 sec

0.000 sec / 0.000 sec

0.000 sec / 0.000 sec

Object Info Session

Postman

File Edit View Help

New Import Runner

My Workspace Invite

status checker

Comments Examples

Send Save

DELETE http://localhost:8080/teststeps/46

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize Text

Status: 200 OK Time: 46 ms Size: 212 B Save Response

Bootcamp Build Browse

89°F 5:38 AM 8/15/2021

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

sample SQL*

173
174 • select * from testcase;
175
176
177
178
179
180

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content |

testcaseid	projectid	order_num	actionid	typed	is_active	created_by	created_timestamp	last_modified_by	last_modified_timestamp	data
11	35	1	1	1	0	aestools	2021-07-27 05:49:04	aestools	2021-07-27 05:49:04	
12	35	2	2	1	0	aestools	2021-07-27 05:50:37	aestools	2021-07-27 05:50:37	
13	35	3	3	1	0	aestools	2021-07-27 05:50:37	aestools	2021-08-08 19:05:05	
14	35	4	4	2	0	aestools	2021-07-27 05:50:37	aestools	2021-08-08 19:05:05	
64	35	5	2	2	0	aestools	2021-07-30 10:28:08	aestools	2021-07-30 10:28:08	
70	27	1	1	1	0	aestools	2021-08-01 03:49:37	aestools	2021-08-02 03:49:37	
71	27	5	30	1	0	aestools	2021-08-02 03:49:37	aestools	2021-08-10 15:50:41	
72	27	3	3	1	0	aestools	2021-08-02 03:49:37	aestools	2021-08-02 03:49:37	
73	27	4	4	2	0	aestools	2021-08-02 03:49:37	aestools	2021-08-02 03:49:37	
91	27	5	2	1	0	aestools	2021-08-06 16:49:47	aestools	2021-08-08 23:50:21	
95	35	5	5	1	0	aestools	2021-08-08 19:21:06	aestools	2021-08-08 19:21:06	
145	27	5	2	1	0	aestools	2021-08-08 21:37:12	aestools	2021-08-08 23:50:38	
146	27	5	2	1	0	aestools	2021-08-08 21:37:31	aestools	2021-08-08 23:51:00	

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

testcase 8

Action Output

#	Time	Action	Message	Duration / Fetch
7	05:35:57	select * from testcase LIMIT 0, 1000	17 row(s) returned	0.000 sec / 0.000 sec
8	05:36:19	select * from testcase LIMIT 0, 1000	17 row(s) returned	0.000 sec / 0.000 sec
9	05:38:23	select * from testcase LIMIT 0, 1000	13 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Bootcamp Build Browse

89°F 5:38 AM 8/15/2021

ADDITIONAL RESPONSIBILITIES

Front-End Developer

I was transferred to front-end team towards the end of the internship since the mentors wanted me to have experience working on that area as well before the internship ended. I was still answerable to the same people.

My additional tasks included:

- Fixing and design issues in the webpages
- All APIs are well integrated with the Frontend and the web page displays information as required
- The code developed is clean and will not create any bugs.

CONCLUSION

Through this internship for 61 days, I learnt how to develop the Server Side of any application. I applied the concepts that I learnt in my core subjects like DBMS, Web Technologies, Software Engineering, OOPs, to name a few. Software Testing tools and Docker/Kubernetes gave an insight to Deployment of applications as well. I also used Usability Testing of Human – Computer Interaction to judge the UX of the developed application. I learnt how to develop scalable applications to meet the industry requirements.

Apart from technical skills, I also learned how to work within a team. I developed leadership skills by leading the Backend team. Apart from that I portrayed excellent time management skills for which I was appreciated in the group.