Internship Report
FEB 2021-JUNE 2021

IN
COMPUTER SCIENCE AND ENGINEERING

By

Pranjal Saxena (171254)

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology
Waknaghat, Solan ,173234, Himachal Pradesh

DECLARATION

I hereby declare that this submission is my own work carried out at Paymentus Corporation(PAXCOM), Mohali from Feb, 2021 to June, 2021 and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Pranjal Saxena

Name: Pranjal Saxena
Date: 17-06-2021

ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted cooperation and have helped me in completing this internship successfully.

First of all, I would like to thank Mr. Gaurav, who mentored me, guided me and challenged me.

I also thank my family and friends who greatly supported me during the course of the Internship.

Last but not the least, I would like to thank our founders for considering me a part of the organization and provide such a great Platform to learn and enhance my skills

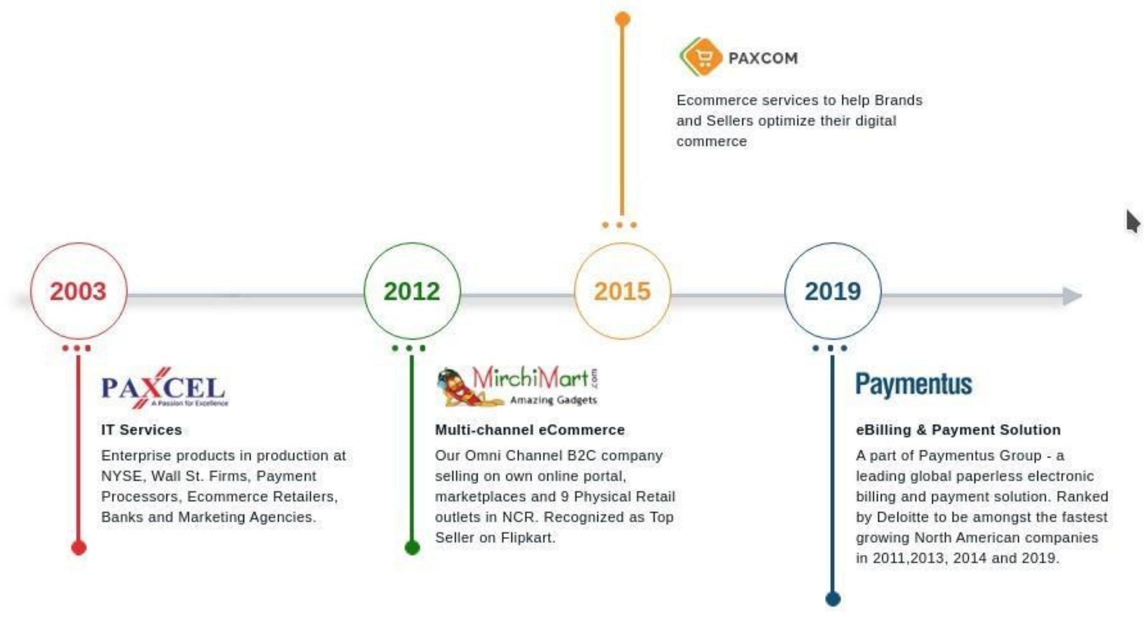
Pranjal Saxena
171254
Jaypee University of Information Technology

Chapter-1

Introduction

Paxcom:

We are a team of 200+ Ecommerce enthusiasts, passionate about using technology to simplify Digital Commerce and Payments for brands across the globe. Paxcom is a part of the Paymentus group - a leading global paperless electronic billing and payment solution provider.



Ecommerce services to help Brands and Sellers optimize their digital commerce

Enterprise products in production at NYSE, Wall St. Firms, Payment Processors, Ecommerce Retailers, Banks and Marketing Agencies.

Our Omni Channel B2C company selling on own online portal, marketplaces and 9 Physical Retail outlets in NCR. Recognized as Top Seller on Flipkart.

A part of Paymentus Group - a leading global paperless electronic billing and payment solution. Ranked by Deloitte to be amongst the fastest growing North American companies in 2011,2013, 2014 and 2019.

speaking.That characterized way empowers the making of a uniform code base over an association, withoutagonizingovercharacterizingcertainprinciples.

Angular as of now accompanies these principles out of the crate. Tailing them doesn't just build the consistency and hence the nature of the code. It makes your application more obvious, as well. That is, in the event that you are as of now acquainted with angular. This exacting methodology likewise proves to be useful across collaboration fringes. It empowers new engineers to incorporate into another group rapidly, as a result of the high commonality with the code.

What I need to state is, you should follow the angular structure rules to capitalize on the angular system. It will make your life significantly simpler, when coming into new undertakings and will expand the nature of your code consequently.
A typical activity, when learning angular is to put everything into the application module. Obviously, that your application will advance into a total chaos. Modules are there which is as it should be!

Modules help to sort out your code into littler groups to make discovering things simpler. Yet, they are not just a corrective thing. With the assistance of lethargic stacked modules, you can likewise build the client experience by just downloading the pieces of the application, that are required at that point.

Angular is written in typescript by Google development team and is maintained by them regularly. It’s first initial release was in beginning of 2016. Very first version was called as Angular-JS so in future updates to make it distinguish from other they named it to just Angular and removed the JS. 11 stable versions has been released by the google team along with the community support as of date 11 Nov 2020. To do material design in anular, it provides angular material library.

GitHub

<u>Paymentus:</u>

Paymentus is a North Carolina based software company providing complete billing solutions.

I worked at Paymentus India Pvt Ltd at Mohali branch. I worked with the Hybrid-Billers Team and was successfully able to understand their working structure and pattern. I also learned soft skills like communicating within a corporate firm and working with a team. I was successfully able to understand various coding paradigms used by a company to build its application. I got a thorough understanding of some of the company's existing products and some of the upcoming products.

The working culture of the company is great. I thoroughly enjoyed myself working there. Paymentus has a typical blend of work and fun. Although I didn't get to spend much time in the company office and started working from home after the coronavirus pandemic lockdown, I really enjoyed the weekend football and various parties at the company. And even after the lockdown the communication between me and my team was good through various meeting platforms like skype and google meets.

There are numerous things I love about my internship - the experience/information I've acquired, the balance between fun and serious activities climate, my director and colleagues, and surprisingly the gathering of people I exercise with on my mid-day breaks. In any case, in the event that I needed to put it on a certain something, what I like most is, at last, the way that this chance truly opened the entryway into my profession. The most stunning thing about this temporary position experience is that it truly overcame any barrier between learning things and really applying a portion of this information into a reality, AND getting paid for it! The way that I presently have an entry-level position straightforwardly identified with what I concentrate on it makes me study more diligently and makes me need to sort out how I can apply a greater amount of what I'm realizing in an internship. It's interesting, in light of the fact that things I've learned in college help me at work

Chapter-2

DESCRIPTIONOFTHEINDUSTRY

Paymentus

In 2004, Paymentus was born from a desire to improve the way bills get paid. Vision, innovation and exemplary service have propelled Paymentus to become the leading paperless electronic billing and payment solution on the market, resulting in 1,300 clients including some of the largest billers in North America.

We know that in order to keep our solutions current and relevant, we need people with the know-how, drive and proclivity for fostering a supremely happy customer experience. Our highly committed, creative employees turned an idea into a secure, SAAS-based Customer Engagement and Payment Platform; one that enables direct-bill organizations to provide a unified customer experience and boost adoption of cost-saving electronic billing and payment services.

Recognized by Deloitte to be among the fastest growing North American companies in 2011, 2013, 2014, and 2016, Paymentus consistently strives to develop better, faster, more secure, cost-efficient billing and payment platforms. We continually seek higher value for our customers, in both solutions and service.

It’s what has led to our remarkable growth in the last decade. We succeed when our clients succeed. They succeed when their customer relationships are enhanced and, in turn, their customers participate in these cost-saving electronic services at high rates.

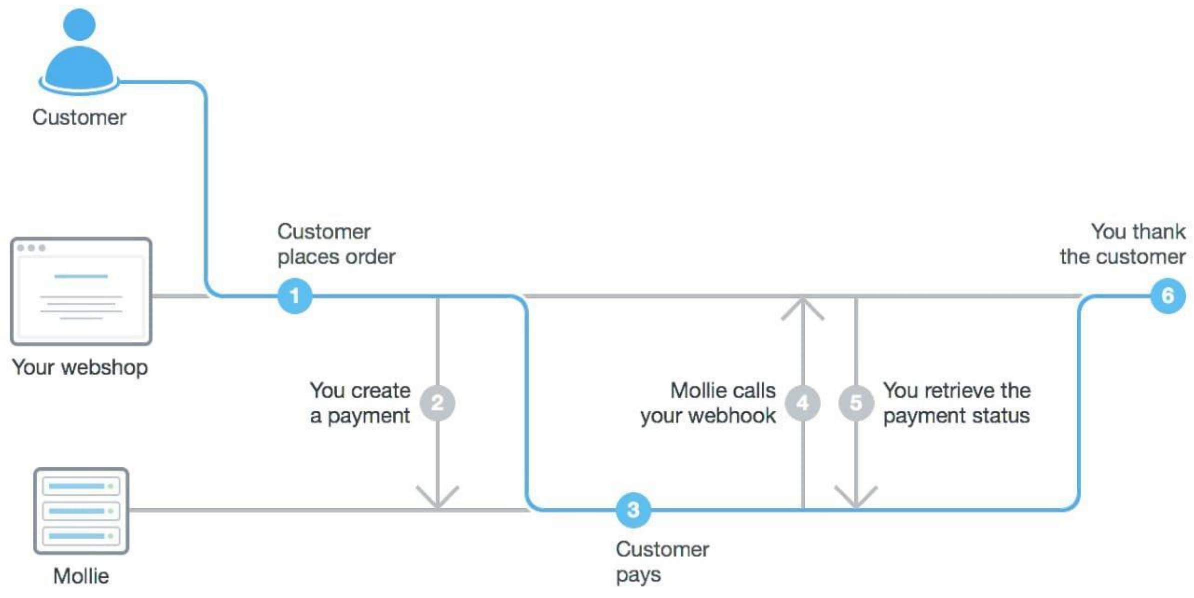
How Paymentus Works:

How to Pay:

Input the URL of the organization you want to pay into your web browser and navigate to the billing section to begin the payment process.

Unsure of the URL?:

It is often found on your paper billing statement. If not, look-up the webpage via your preferred search engine (Google, Bing, etc.).



Chapter–3

Tools and Technologies used

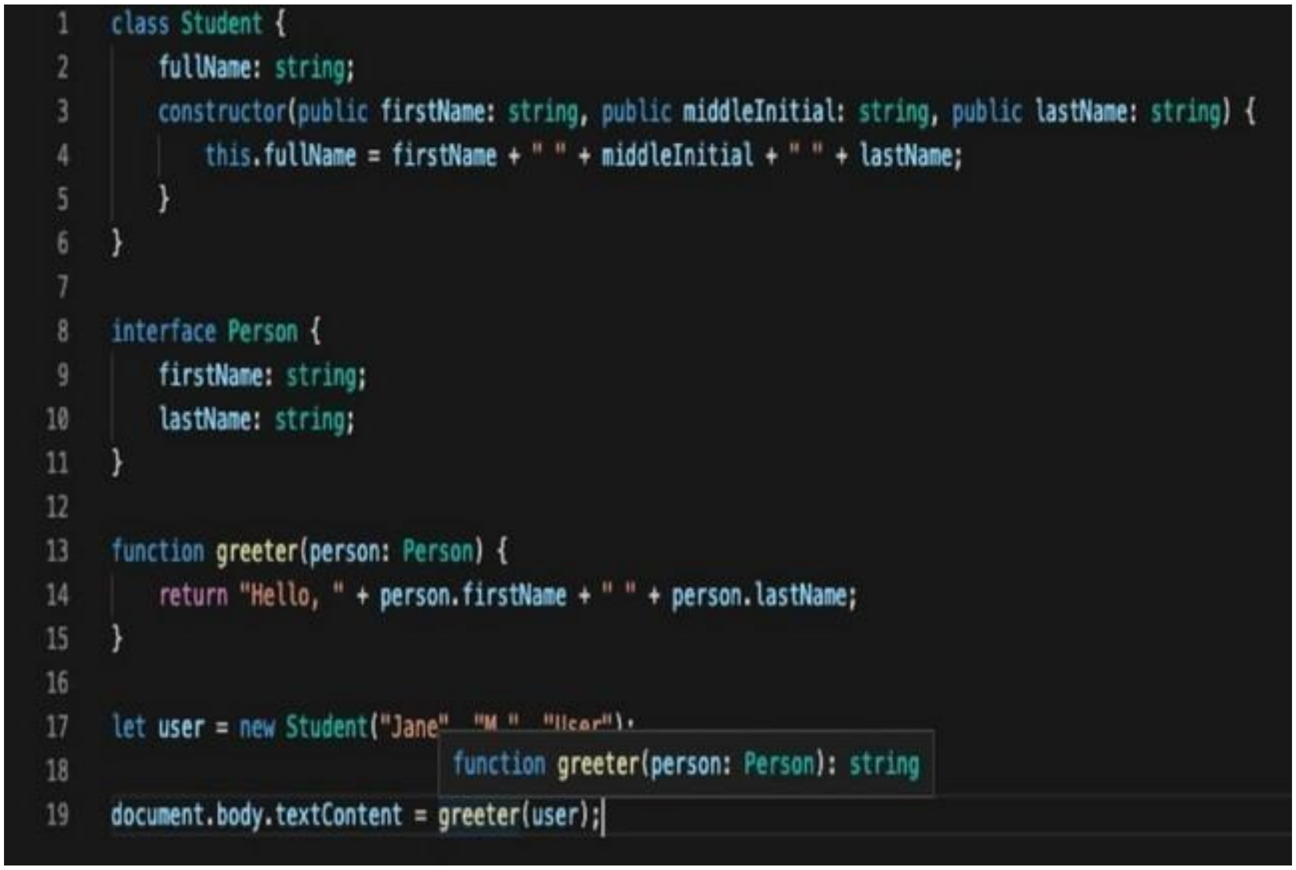
My Internship at Paymentus involved learning skills in languages and tools like **Angular**, **HTML**, **CSS**, **Typescript**, **GithHub, NodeJS** and **Javascript**. Some of the technologies we also used in our day to day work are **Kafka**, **Jira**, **Docker**, **Kubernetes**, **MongoDB** etc. Apart from that this internship has also taught me soft skills like working culture in a corporate firm, communication within a team. I can call it my most improved skill by joining daily meetings to discuss new strategies to improve and pipeline to implement them. Although these meetings were very short time wise (ie. 5-15 mins) but informatively, they were sufficient enough to get us hooked for the day. We were assigned seniors from the company that managed our work and guided us in right directions. They gave us feedback on every turn and helped us in very best way possible. The one thing we would like to add in this report about the seniors that managed us is that they not only taught us how to work efficiently and be more productive to the company but also how to manage work-life balance, how to overcome frustration of working in these covid times. Next I will be explaining in detail the technologies that we used.

JavaScript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

- Javascript is the most popular programming language in the world and that makes it a programmer's great choice. Once you learnt Javascript, it helps you developing great front-end as well as back-end softwares using different Javascript based frameworks like jQuery, Node.JS etc.

- Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript.



```
class Student {
    fullName: string;
    constructor(public firstName: string, public middleInitial: string, public lastName: string) {
        this.fullName = firstName + " " + middleInitial + " " + lastName;
    }
}

interface Person {
    firstName: string;
    lastName: string;
}

function greeter(person: Person) {
    return "Hello, " + person.firstName + " " + person.lastName;
}

let user = new Student("Jane", "M.", "User");
                        function greeter(person: Person): string
document.body.textContent = greeter(user);
```

- Javascript helps us create really beautiful and crazy fast websites. We can develop your website with a console like look and feel and give your users the best Graphical User Experience.

- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for one as a Javascript Programmer.

- Great thing about Javascript is that we can find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce our time to market.

When should we avoid JavaScript?

Despite the fact that JavaScript is a widely implemented and standardized language amongst web browsers you should always check to see if what you are trying to accomplish can be done in HTML, CSS, or (in some cases) a server side language first. This is because several older or low powered devices such as mobile phones have difficulties handling JavaScript. Also computers with higher security needs often disable JavaScript due to potential flaws. JavaScript is a very powerful tool, but remember to use it sparingly and look for alternative solutions.

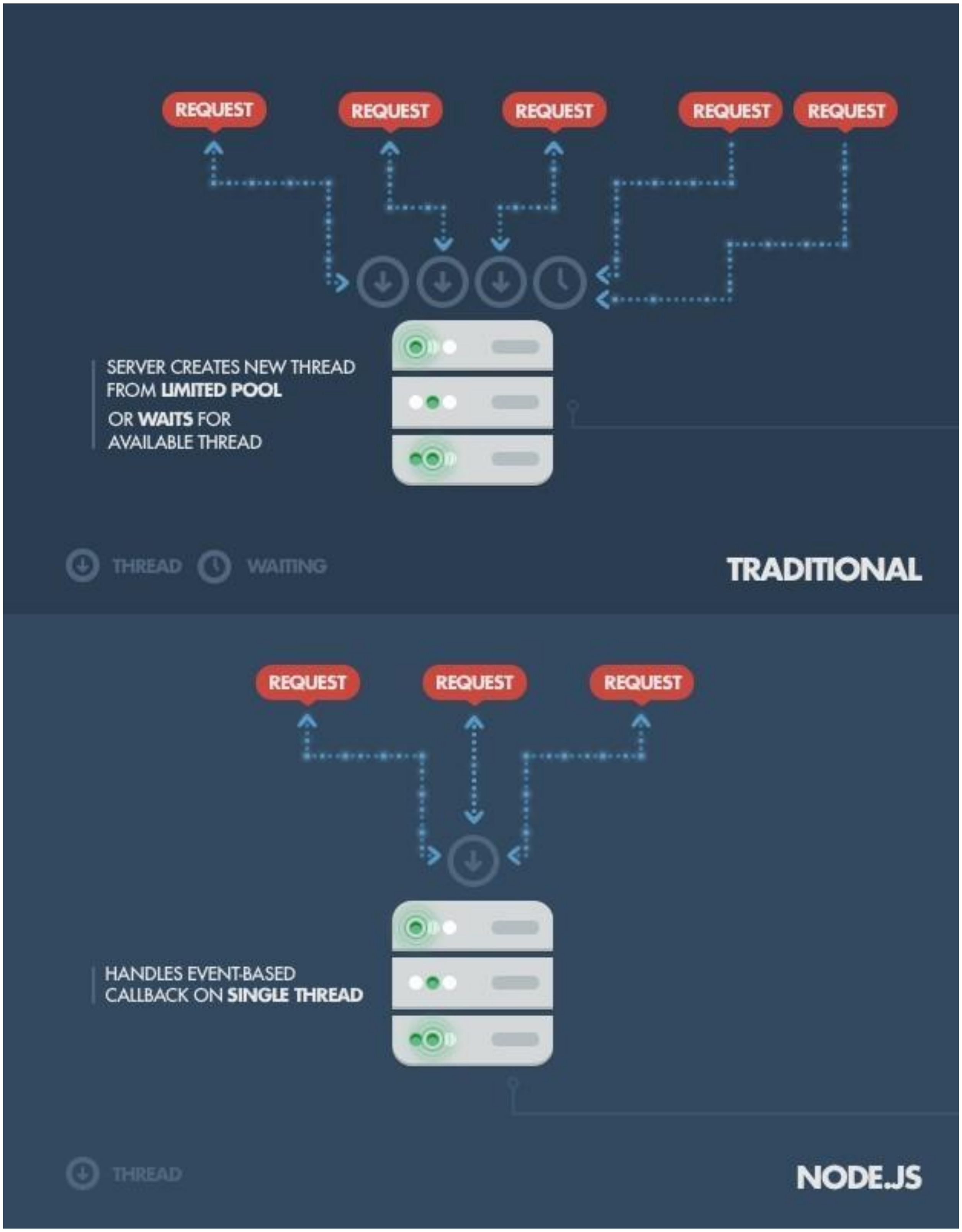
NODEJS

After over 20 years of stateless-web based on the stateless request-response paradigm, we finally have web applications with real-time, two-way connections.

In one sentence: Node.js shines in real-time web applications employing push technology over websockets. What is so revolutionary about that? Well, after over 20 years of stateless-web based on the stateless request-response paradigm, we finally have web applications with real-time, two-way connections, where both the client and server can initiate communication, allowing them to exchange data freely. This is in stark contrast to the typical web response paradigm, where the client always initiates communication. Additionally, it’s all based on the open web stack (HTML, CSS and JS) running over the standard port 80.

One might argue that we’ve had this for years in the form of Flash and Java Applets—but in reality, those were just sandboxed environments using the web as a transport protocol to be delivered to the client. Plus, they were run in isolation and often operated over non-standard ports, which may have required extra permissions and such.

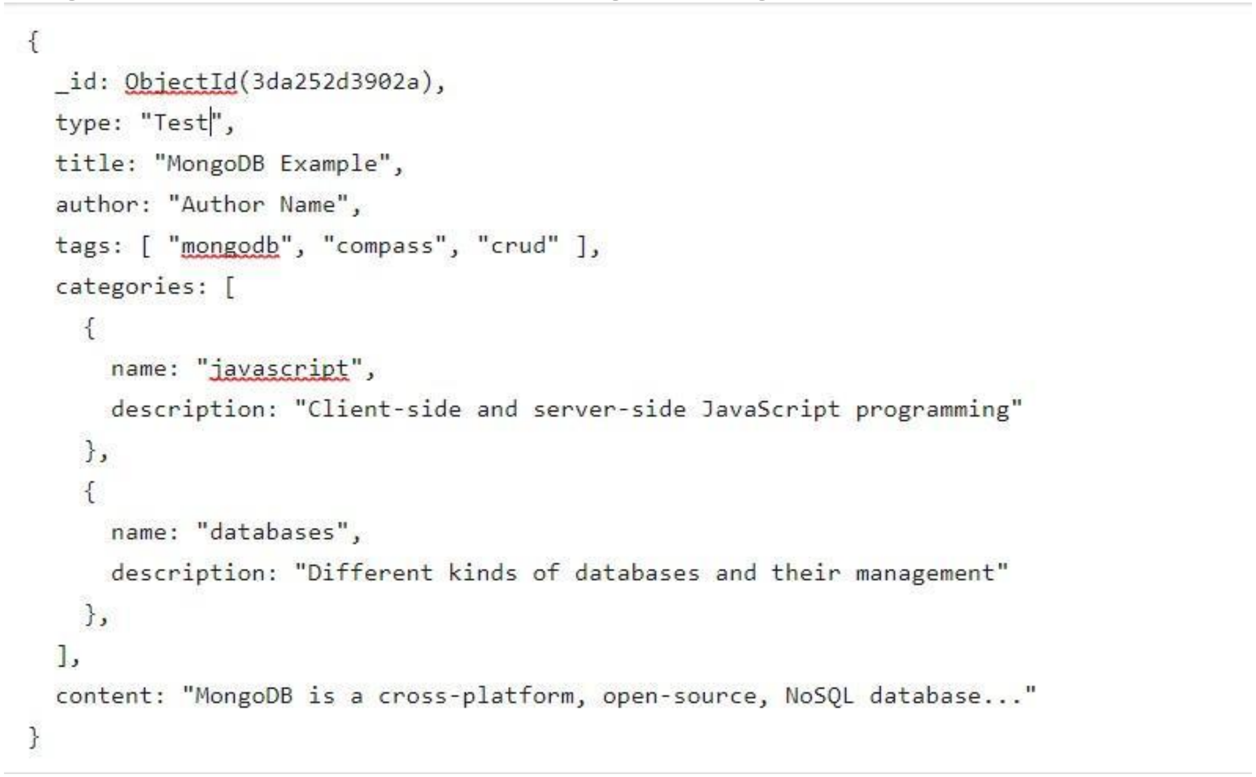
Node.js was never made to solve the compute scaling issue. It was made to tackle the I/O scaling issue, which it does extremely well. Being single-threaded, Node.js might be a bad choice for web servers serving as computational servers, since heavy computation will block the server's responsiveness. If your use case does not contain CPU escalated activities or get to any blocking resources, you can exploit the advantages of Node.js and make quick and adaptable system applications.


REQUEST
REQUEST
REQUEST
REQUEST
REQUEST
SERVER CREATES NEW THREAD FROM **LIMITED POOL**
OR **WAITS** FOR AVAILABLE THREAD
THREAD
WAITING
TRADITIONAL
REQUEST
REQUEST
REQUEST
HANDLES EVENT-BASED CALLBACK ON **SINGLE THREAD**
THREAD
NODE.JS

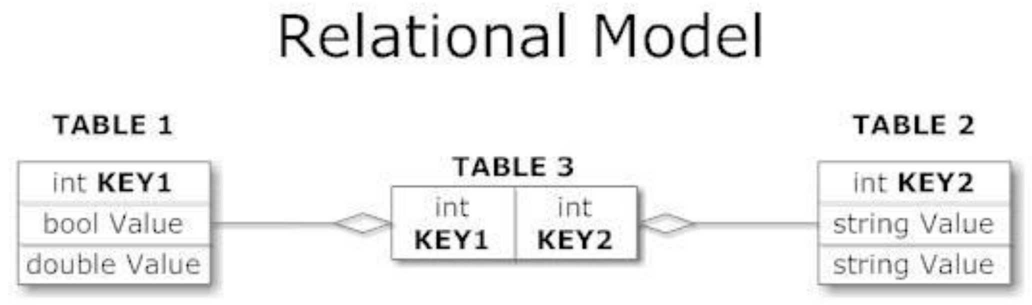
MongoDB

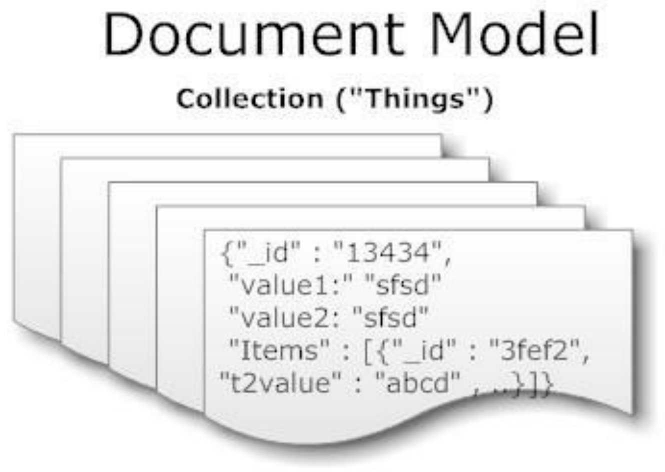
Mongo DB is a database which doesn’t contain SQL, it has no relations, that is it doesn’t contain tables. It contains what is known as a object. Data is stored in form of a object.

MongoDb schema i   s flexible and can be changed on the go unlike SQL .



```
{
  _id: ObjectId(3da252d3902a),
  type: "Test",
  title: "MongoDB Example",
  author: "Author Name",
  tags: [ "mongodb", "compass", "crud" ],
  categories: [
    {
      name: "javascript",
      description: "Client-side and server-side JavaScript programming"
    },
    {
      name: "databases",
      description: "Different kinds of databases and their management"
    },
  ],
  content: "MongoDB is a cross-platform, open-source, NoSQL database..."
}
```





```
{"_id" : "13434",
 "value1:" "sfsd"
 "value2: "sfsd"
 "Items" : [{"_id" : "3fef2",
"t2value" : "abcd" , ..}]}
```

Angular

One of the center contentions to pick angular over other single page application systems is its plainly characterized method of how things should be finished. It has an assessment, in a manner of

With all of its advantages, Node.js now plays a critical role in the technology stack of many high-profile companies who depend on its unique benefits.

Node.js is basically a server capable of executing JavaScript. At its core, Node.js is a server engine that you can alter and change, and it will just work after you set it up. It offers asynchronous and event-driven APIs so the requests to it are processed as a loop (event loop), and that's why Node.js is essentially a runtime. Being a part of the JavaScript ecosystem, which is awesome for application development, you can control it effortlessly alongside different JS tools, UIs, and connectors. In other words, it is an open source and cross-platform system for building web applications with just a few lines of code.

The chat application is the most typical real-time application where Node.js shows what it can do in terms of handling multiple users, intensive data, big traffic, and running across devices. Additionally, it's great to learn Node while making a chat app, as it covers almost all the programming of a typical Node.js application.

Because of HTML, server-side web applications are not a typical use case for Node.js. However, in the event that you combine Node.js and Express.js, you can make exemplary web applications on the server-side.

Node.js has some positive highlights in data streaming utilizing the fact that HTTP requests and responses are basically data streams. For example, processing data while it's simply being transferred, e.g. for sound/video encoding.

Another use case of Node.js is monitoring dashboards to gather on-going information about website visitors and visualization. User statistics and the ability to see what they are doing instantly, is definitely a great add-on for businesses.

Time will prove whether Node is that next big thing.

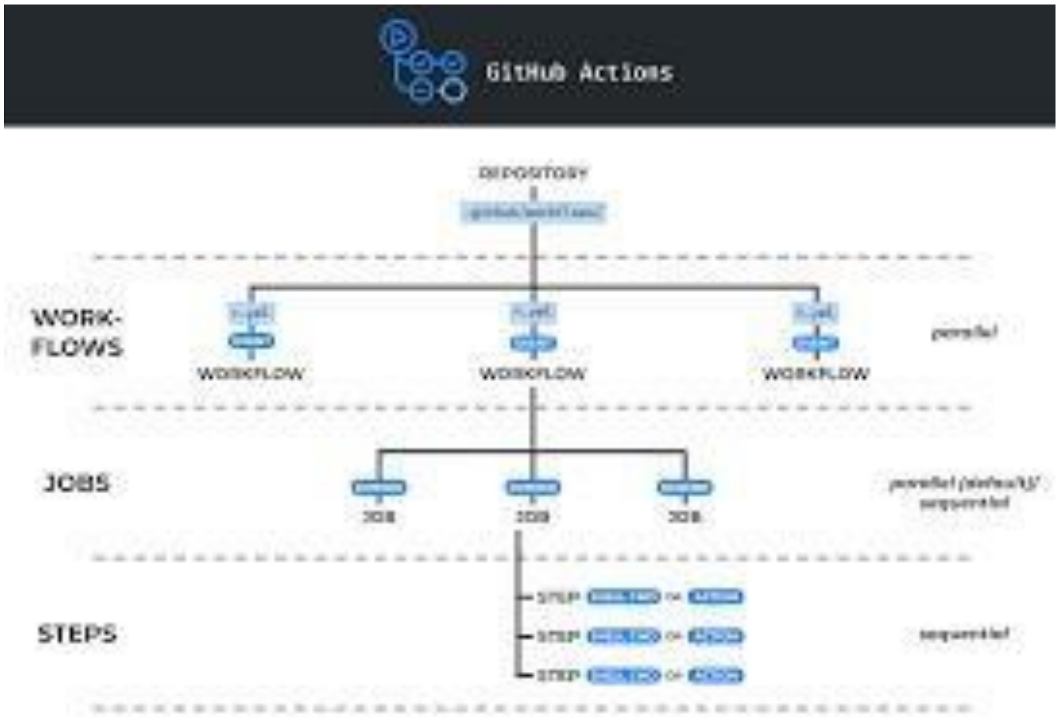
Git is utilized for dealing with the progressions to a task after some time. A task may be only a solitary record, a bunch of documents, or a large number of documents. Those records can be anything from plain content to pictures or recordings.

Since Git is centered around overseeing transforms, it is regularly utilized as a joint effort instrument permitting individuals to take a shot at a similar undertaking simultaneously. By following their individual changes, Git can unite everything to the last form. Envision that you're composing a blog entry that has numerous records related with it. You may have one principle content record which is the real post, an extra document for references, just as some different records that are graphs and different pictures.

Without Git, you may have these different records put away in an organizer on your PC, however it is extremely unlikely to tell where the entirety of the documents are at a given point in time. Envision you send your post to two companions to duplicate alter. How would you consolidate their progressions back? Which record is the first, which is altered?

With Git you can follow the entire repo at different focuses in time utilizing submits while likewise giving comments on why you chose to spare the undertaking by then. Afterward, you can per use this history of resolves to see an away from of your undertaking, and furthermore travel back in that history, if important. A software codebase works simply like that blog entry. At its most essential level, it is an assortment of records that are connected to each other.

At the point when an engineer is taking a 5 shot at a specific element, Git gives an approach to spare a depiction of the whole repo by means of a submit. This is generally done when gradual advancement has been made and an element is without bug. In making the submit, the designer can give comments clarifying what was changed and why it was changed. This message adds to the historical backdrop of the extend and can make it simple to decide when a specific element – or even a bug – was presented.



Typescript

The definition from the official site says: "a composed superset of JavaScript" yet it accept you comprehend what a "superset" is and what "composed" signifies. Rather to keep things straightforwardyoucanconsiderTypeScriptof"alayerontop"ofJavaScript.

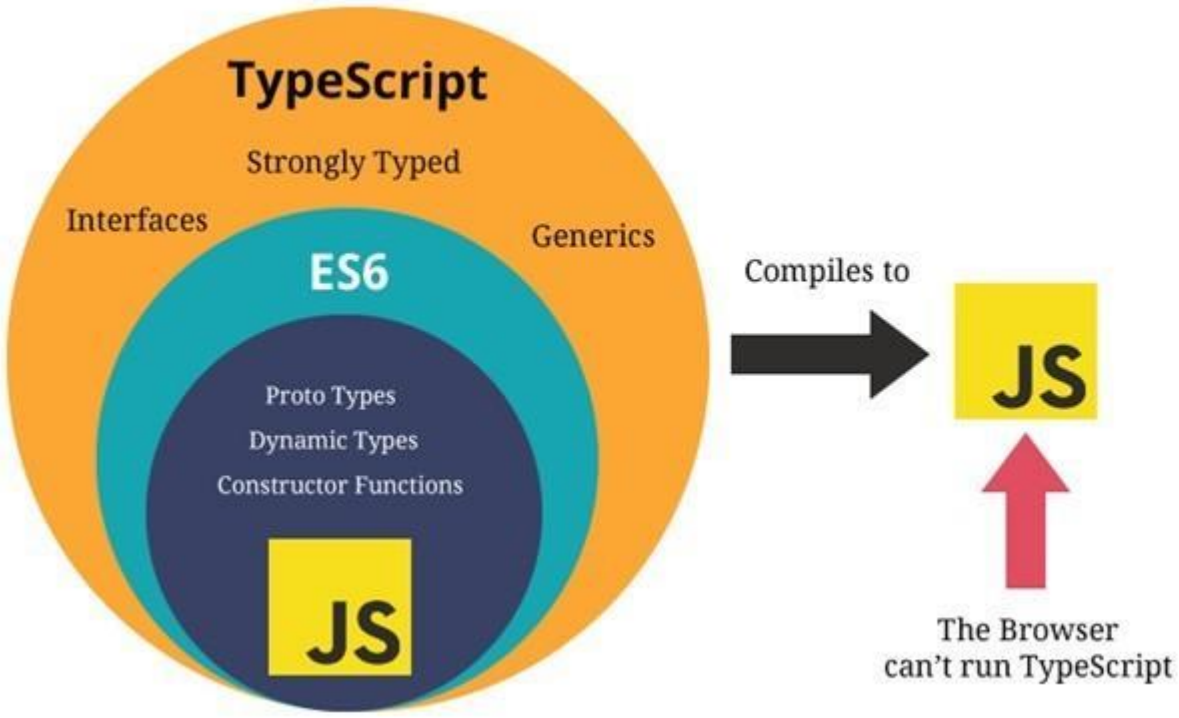
TypeScript is a layer since you can compose TypeScript code in your proofreader. After a gathering all thatTypeScriptstuffisgoneandyou'releftwithplain,straightforwardJavaScript.

Ontheoffchancethatthepossibilityofanassemblagestepconfoundsyourememberthat

JavaScript is as of now accumulated and afterward deciphered. There is a JavaScript motor that peruses and executes your code.

Be that as it may, JavaScript motors can't peruse TypeScript code so any TypeScript record ought to go under a "pre-interpretation" process, called aggregation. Simply after the principal assemblage step you're left with unadulterated JavaScript code, prepared to run in a program. TypeScript is an exceptional sort of JavaScript yet it needs an "interpreter" before running in a Program.

TypeScript begins from a similar punctuation and semantics that a large number of JavaScript designers know today. Utilize existing JavaScript code, join mainstream JavaScript libraries, and call TypeScript code from JavaScript.TypeScript arranges to spotless, basic JavaScript code
which runs on any program, in Node.js, or in any JavaScript motor that underpins ECMAScript 3 (or more current).


TypeScript
Strongly Typed
Interfaces
Generics
ES6
Proto Types
Dynamic Types
Constructor Functions
JS
Compiles to
JS
The Browser
can't run TypeScript

Angular Material Design

MaterialDesign

ut ilized to make computerized encounters. It's a lot of standards and rules across stages and gadgets for intuitiveness, movement and segments that streamlinetheplanworkprocessforgroupsstructuringtheiritem.

The Material parts permit you to make proficient UIs with ground-breaking measured quality, themingandcustomizationhighlights.

Angular Material Is The execution of Material Design standards and rules for Angular. It contains differentUIsegments,forexample,

● structureCont rols(in put,se lect,ch eckb.ox,datep.icker and s lidersandsoon.),
● routedesigns(menus,sidenavandtoolbar)
● format segments (matrices, cards, tabs and records)

● catches
● pointers (progress bars and spinners)
● popups and modals

POSTMAN

Postman is a scalable API testing tool that quickly integrates into CI/CD pipeline. It started in 2012 as a side project by Abhinav Asthana to simplify API workflow in testing and development. API stands for Application Programming Interface which allows software applications to communicate with each other via API calls.
Here are some major reasons on why we should use Postman over it's competitions.

With over 4 million users nowadays, Postman Software has become a tool of choice for the following reasons:

1. Accessibility - To use Postman tool, one would just need to log-in to their own accounts making it easy to access files anytime, anywhere as long as a Postman application is installed on the computer.
2. Use of Collections - Postman lets users create collections for their Postman API calls. Each collection can create subfolders and multiple requests. This helps in organizing your test suites.
3. Collaboration - Collections and environments can be imported or exported making it easy to share files. A direct link can also be used to share collections.
4. Creating Environments - Having multiple environments aids in less repetition of tests as one can use the same collection but for a different environment. This is where parameterization will take place which we will discuss in further lessons.
5. Creation of Tests - Test checkpoints such as verifying for successful HTTP response status can be added to each Postman API calls which help ensure test coverage.
6. Automation Testing - Through the use of the Collection Runner or Newman, tests can be run in multiple iterations saving time for repetitive tests.
7. Whenever we make a big project after some time we lose small details and minor errors leads to

unresovable issues, so this is debugger comes in to play which can help you check what data youhaveretrievedandmakes processofdebuggingeasier.

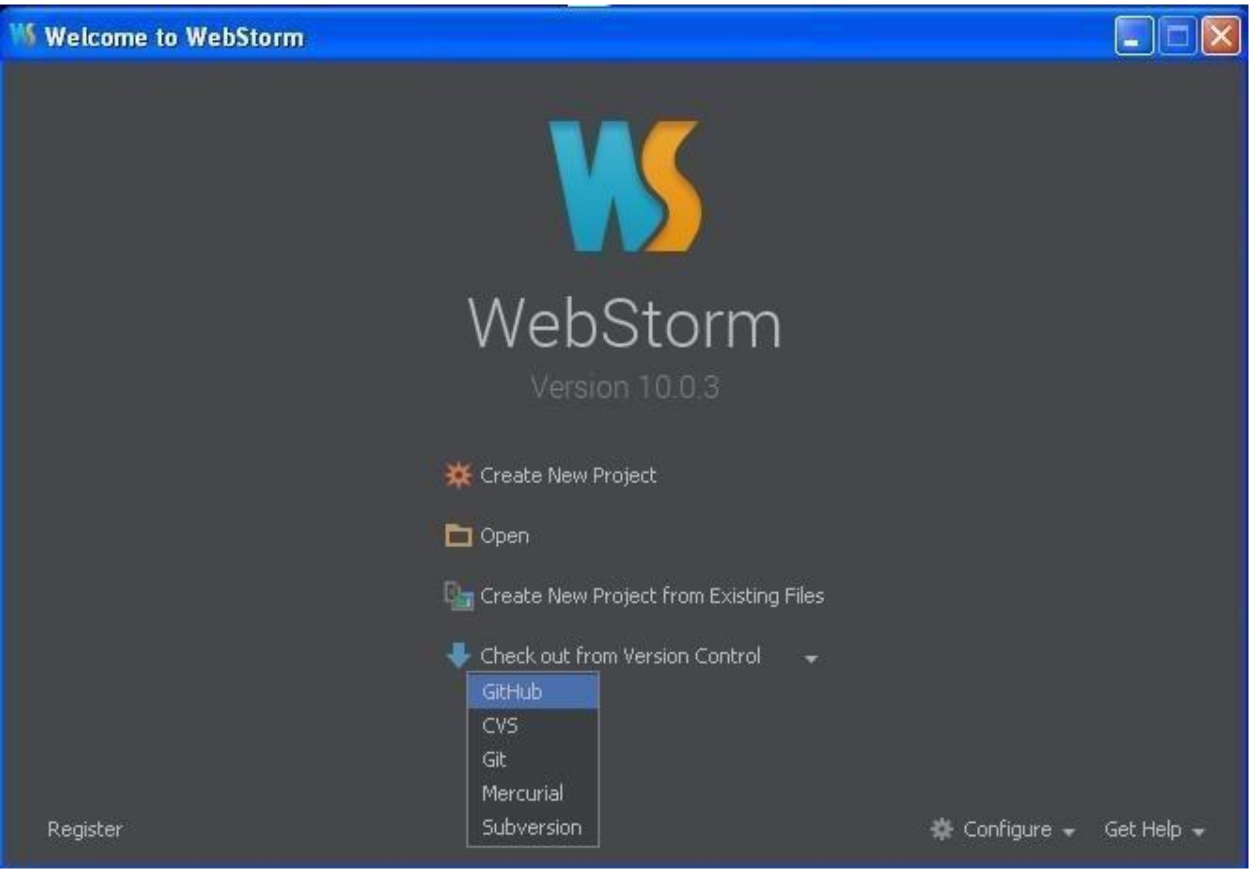
8. Inanydevelopmentwork,itisbasicpracticethatwecancontinuouslyintegratenewupdatesto ourbaselineproduct.Postmanprovidesthisabilityandmakesitveryeasytodoso.

HowtousePostmantoexecuteAPIs

“Ihaveattachedascreenshottoshowhowthepostmanworkspacelookslike.Ihavementionedsome pointsarethatworksasastepbystepguidetoexploretheenvironmentanddifferentfeaturesof Postman.”

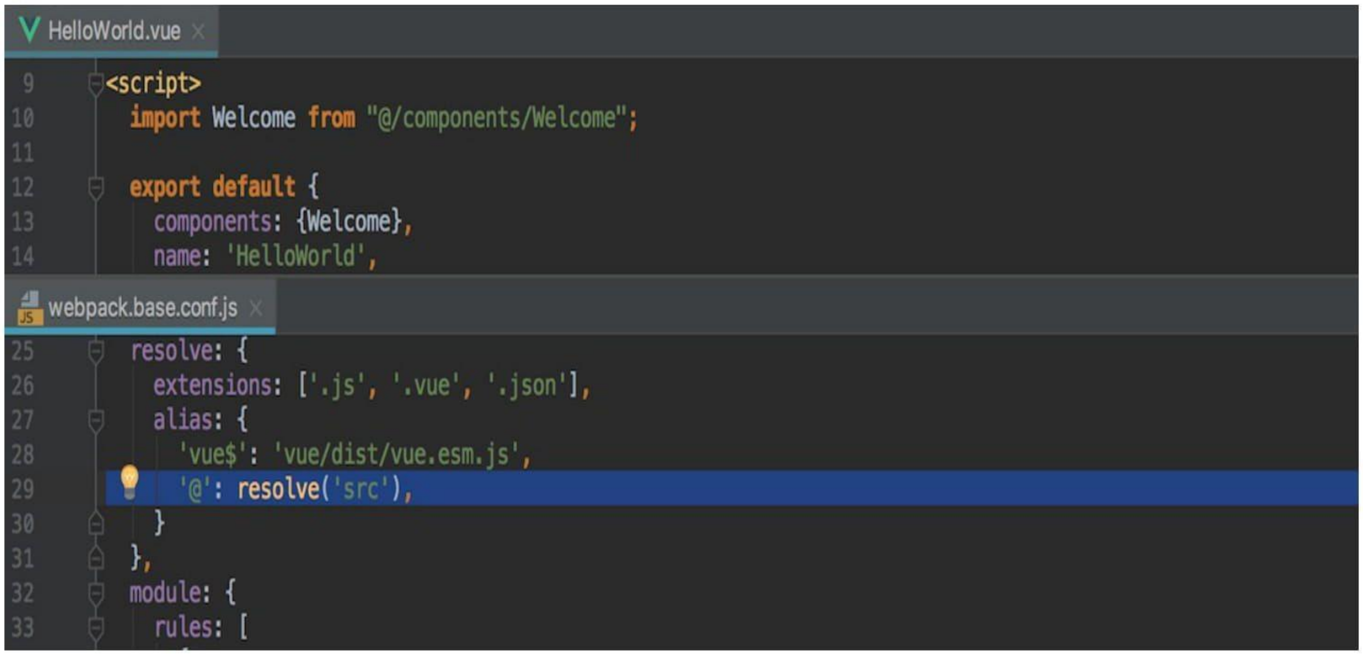
1. New - This is where you will create a new request, collection or environment.
2. Import - This is used to import a collection or environment. There are options such as import from file, folder, link or paste raw text.
3. Runner - Automation tests can be executed through the Collection Runner. This will be discussed further in the next lesson.
4. Open New - Open a new tab, Postman Window or Runner Window by clicking this button.
5. My Workspace - You can create a new workspace individually or as a team.
6. Invite - Collaborate on a workspace by inviting team members.
7. History - Past requests that you have sent will be displayed in History. This makes it easy to track actions that you have done.
8. Collections - Organize your test suite by creating collections. Each collection may have subfolders and multiple requests. A request or folder can also be duplicated as well.
9. Request tab - This displays the title of the request you are working on. By default, "Untitled Request" would be displayed for requests without titles.
10. HTTP Request - Clicking this would display a dropdown list of different requests such as GET, POST, COPY, DELETE, etc. In Postman API testing, the most commonly used requests are GET and POST.
11. Request URL - Also known as an endpoint, this is where you will identify the link to where the API will communicate with.
12. Save - If there are changes to a request, clicking save is a must so that new changes will not be lost or overwritten.
13. Params - This is where you will write parameters needed for a request such as key values.
14. Authorization - In order to access APIs, proper authorization is needed. It may be in the form of a username and password, bearer token, etc.
15. Headers - You can set headers such as content type JSON depending on the needs of the organization.
16. Body - This is where one can customize details in a request commonly used in POST request.
17. Pre-request Script - These are scripts that will be executed before the request. Usually, pre-request scripts for the setting environment are used to ensure that tests will be run in the correct environment.

JETBRAINSWEBSTORM


Welcome to WebStorm
WebStorm
Version 10.0.3
Create New Project
Open
Create New Project from Existing Files
Check out from Version Control
GitHub
CVS
Git
Mercurial
Subversion
Register
Configure
Get Help

Webstorm : As the name says it is a development environment that is mainly focused on web related technologies such as Js and all of it's family options. Just like other development environments, it also offers easy project setup, a very helpful
debugger, easy to configure environment and paths. It also provies basic code automations and with some plugins to help with code suggestion we have a lot easier environment to work with that can reduce typos to a great extent.
It also helps in refactoring and on the fly error prevention and much more. It also has a very active community that supports you deal with common issues and present a great learning opportunity overall. With right use, you can achieve great comfort in using webstorm and will be preferred in works over others.

Initially it was founded by three russian devs in Prague in early 2000. It was designed for JAVA only in its intial releases and later on after some releases they started including other languages. so this way a very basic debugger and tester evolved into very helpful software for other developers.



```
<script>
  import Welcome from "@/components/Welcome";

  export default {
    components: {Welcome},
    name: 'HelloWorld',
```

```
  resolve: {
    extensions: ['.js', '.vue', '.json'],
    alias: {
      'vue$': 'vue/dist/vue.esm.js',
      '@': resolve('src'),
    }
  },
  module: {
    rules: [
```

JIRA

What is JIRA?

JIRA is a tool developed by Australian Company Atlassian. This software is used for bug tracking, issue tracking, and project management. The JIRA full form is actually inherited from the Japanese word "Gojira" which means "Godzilla". The basic use of this tool is to track issues and bugs related to your software and mobile apps.

It is also used for project management. The JIRA dashboard consists of many useful functions and features which make handling of issues easy. Some of the key features are listed below.

Jira software can be used for the following purposes:

- Requirements and Test case management
- In Agile Methodology
- Project Management
- Software Development
- Product Management
- Task Management
- Bug Tracking

Here is a step by step process on how to use Jira software:

- **Step 1)** Open Jira software and navigate to the Jira Home icon
- **Step 2)** Select Create project option
- **Step 3)** Choose a template from the library

- **Step4)**You shoul setup the rows as per the direction from managment.
- **Step5)**Start with init command of the issue.
- **Step6)**Send invitation to your team mates and start working!

InsideJIRAeeverythingcanbeconfigured,theyconsistofthefollowing

- “ScreenNotification”
- “Permission”
- “FieldConfiguration”
- “customFields”
- “Issuetypes”
- “Workflows”
- “EntityManagement”

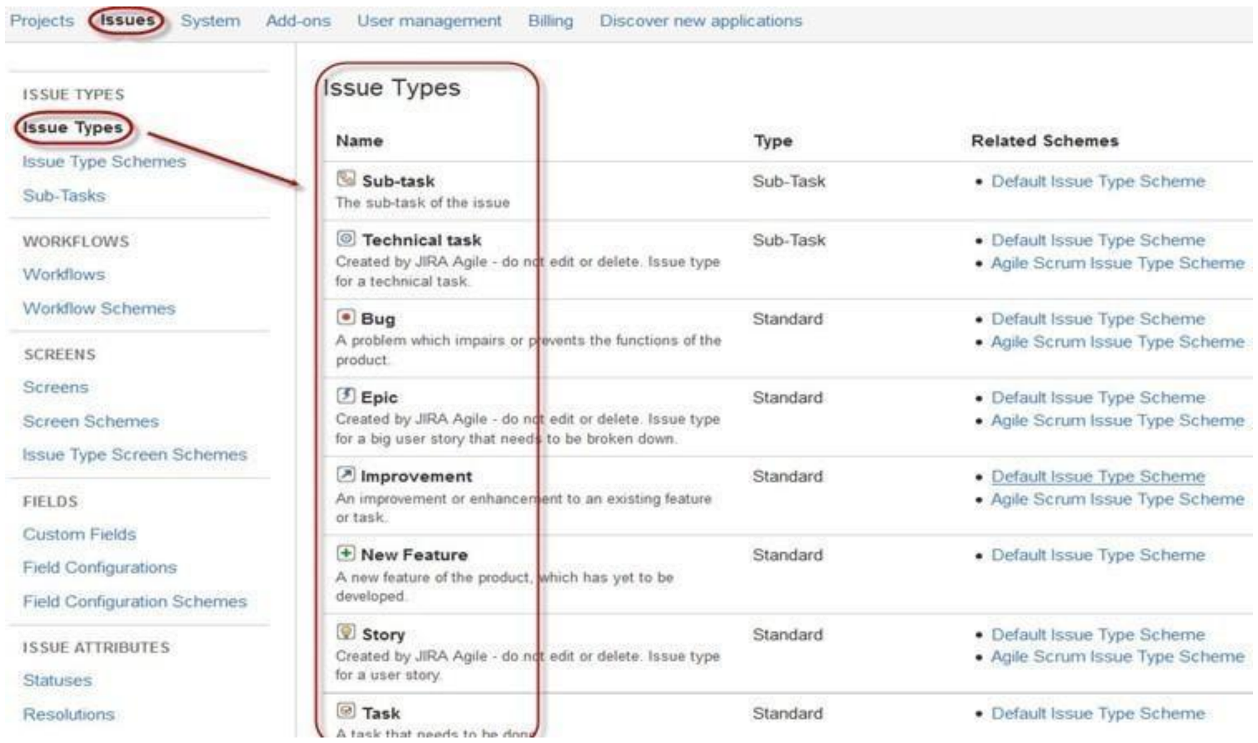
So in this tutorial, we shall learn to implement the actual software in detail as follows:

Lets see what all kinds of issues/tickets JIRA has to offer:

IssueTypedisplaysalltypesofitemsthatcanbecreated and tracked via Jira testing tool. JIRA Issues are classified under various forms like new feature, sub-task,bug,etc.asshowninthescreenshot.

InsideJIRAeeverythingcanbeconfigured,theyconsistofthefollowing:

- “Screen”
- “Notification”
- ‘Permission”
- “FieldConfiguration”
- “customFields”
- “Issuetypes”
- “Workflows”
- ‘EntityManagement’



Issue Types

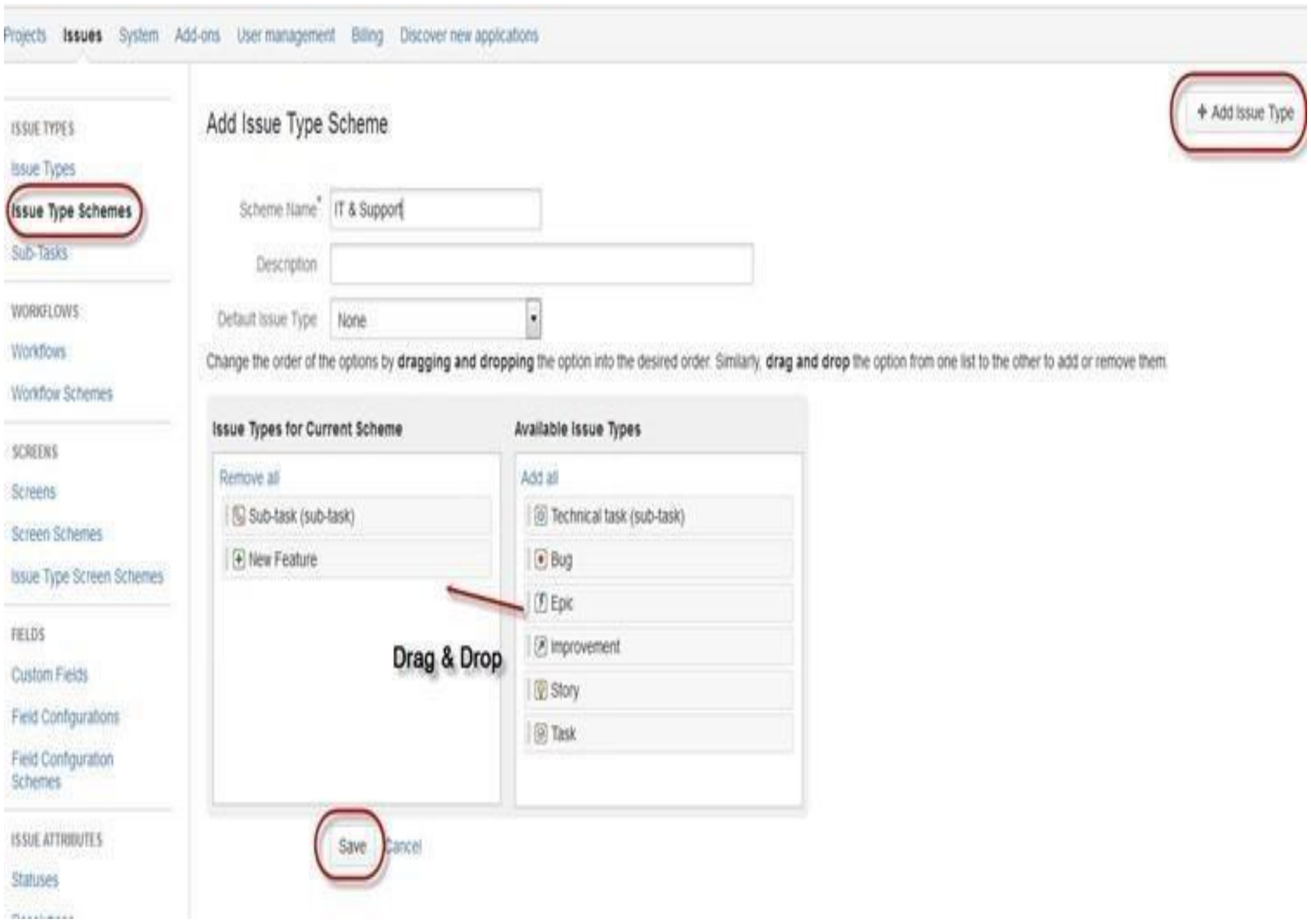
Name	Type	Related Schemes
Sub-task The sub-task of the issue	Sub-Task	• Default Issue Type Scheme
Technical task Created by JIRA Agile - do not edit or delete. Issue type for a technical task.	Sub-Task	• Default Issue Type Scheme • Agile Scrum Issue Type Scheme
Bug A problem which impairs or prevents the functions of the product.	Standard	• Default Issue Type Scheme • Agile Scrum Issue Type Scheme
Epic Created by JIRA Agile - do not edit or delete. Issue type for a big user story that needs to be broken down.	Standard	• Default Issue Type Scheme • Agile Scrum Issue Type Scheme
Improvement An improvement or enhancement to an existing feature or task.	Standard	• Default Issue Type Scheme • Agile Scrum Issue Type Scheme
New Feature A new feature of the product, which has yet to be developed.	Standard	• Default Issue Type Scheme
Story Created by JIRA Agile - do not edit or delete. Issue type for a user story.	Standard	• Default Issue Type Scheme • Agile Scrum Issue Type Scheme
Task A task that needs to be don	Standard	• Default Issue Type Scheme

NextwewillseeJIRAISSUEinsomedetailsinupcomingpage.

Some type of Issues

hellothere are you whaat oding typeofissuesthstyoucancreateandtrackdown.Theyareclassified undersubcategoriessuchassubbtask,busg,etc.Lookinthescreensjotfordetails.

Apartfromthesetwoissuetypeschemes,youcanalsoaddschemesmanuallyasperrequirement,for examplewehavecreatedIT&Supportscheme,forthesewewilldraganddroptheissuetypesfromthe AvailableIssuetypetoIssuetypeforcurrentschemeasshowninthescreenshotbelow.



Add Issue Type Scheme

Scheme Name* IT & Support

Description

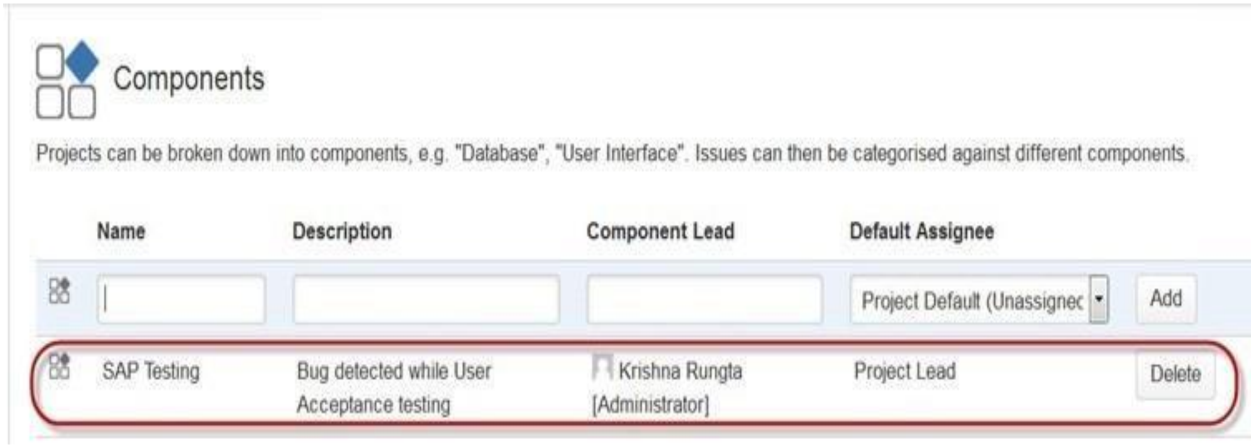
Default Issue Type None

Change the order of the options by **dragging and dropping** the option into the desired order. Similarly, **drag and drop** the option from one list to the other to add or remove them.

Issue Types for Current Scheme	Available Issue Types
Remove all	Add all
Sub-task (sub-task)	Technical task (sub-task)
New Feature	Bug
	Epic
	Improvement
	Story
	Task

Apart from these two basic schemes we can also do manual sceme addition that comes in very handy.

You can also decide which issues should come under your custom scheme.



Components

Projects can be broken down into components, e.g. "Database", "User Interface". Issues can then be categorised against different components.

Name	Description	Component Lead	Default Assignee	
			Project Default (Unassignec	Add
SAP Testing	Bug detected while User Acceptance testing	Krishna Rungta [Administrator]	Project Lead	Delete

, Also we’ve removed th text which was wrong in the previous check. component lead and default assignee. Subsections of a project are jira comeponents. They can be used to group in smaller subsections and taken easily on by one and also helps in gaining easy insight into the issues and help in deadlines.

As seen from the screen shot, to add a new entry you have to fill it up accoridngly.

RESULTS AND CONCLUSION

This internship was indeed a pool of knowledge, not only have I gained knowledge in Full Stack Development but I have also learned about how development of any project takes place, how team works, how the work of each employee is tracked, how work is distributed between different team mates, what are the different stages of development, what are the technical problems that one faces in the development of any project, what all things are required before the development of any project, what the code base should be like and what norms need to be followed in the development. The card scanner is able to scan various types of debit / credit cards. It easily scans non embossed cards but takes some time to scan the embossed ones. This can be improved by further improvements in the scanning algorithm used

REFERENCES

- Angular.io
- Github.com
- Paxcom.net
- Paymentus.com
- Wikipedia.com
- Material.angular.io
- Npmjs.com
- restfulapi.net