

INTERNSHIP REPORT

M Rachana Shenoy

USN – 4MT17IS024

INFORMATION SCIENCE AND ENGINEERING
MANGALORE INSTITUTE OF TECHNOLOGY AND ENGINEERING, MOODABIDIRI

ABSTRACT

The Internship was done at PACE Wisdom Solution which is a Bangalore- based mobile application development organization. This internship has helped in knowing how web actually works. The internship started with the basics of web and then proceeded by the assignment of a project. The project is about developing an application which provides various services to temples, this was achieved by building APIs using LOOPBACK with MONGODB as a database and javascript as the language. The main function/features of the API is to provide user verification, choose appropriate “plans” which is basically a service and managing and retrieval of valid data from the database in an authorized environment. As a student it was a great experience to work on real-time applications with the professionals as our guides which has helped us to gain knowledge on the web domain. This project is completed with my limited knowledge which I would like to improve in the future and kindly pardon my mistakes if there were any from my side. Thank you for providing me with this opportunity.

ACKNOWLEDGEMENT

I deem it a privilege to place on record the deep sense of gratitude to my supervisor **Mr Manjunath H**, Head of the Department, Information Science and Engineering, who always stood behind me and supported in each and every step of the internship work. I am grateful for his support and encouragement.

I am grateful to **Mr. Mohammed Azzan Patni**, the Co-founder and Director at Pace Wisdom Solutions Pvt. Ltd, for being a very supportive and encouraging mentor throughout my internship.

I would like to thank our Dean - Placement and Training **Mr. Narendra U. P.** for his constant support and encouragement to accomplish the internship work.

I would like to thank our Principal **Dr. G.L. Easwara Prasad**, for encouraging me and giving me an opportunity to accomplish the internship work.

I also thank our management who helped me directly and indirectly in the completion of this work. My special thanks to faculty members and others for their constant help and support.

Above all, I extend my sincere gratitude to my parents and friends for their constant encouragement with moral support.

M Rachana Shenoy
4MT17IS024

CONTENTS

Abstract	i
Acknowledgement	ii
Contents	iii
List of Figures	iv

Chapter Number	Chapter Name	Page Number
1	INTRODUCTION	1
2	ABOUT THE COMPANY	2
	2.1 SUMMARY OF THE COMPANY	2
	2.2 SERVICES PROVIDED	3
	2.3 CLIENTS	3
3	TECHNOLOGIES COVERED	4
4	TASKS PERFORMED	6
	4.1 WEEK 1	6
	4.2 WEEK 2	7
	4.3 WEEK 3	8
	4.4 WEEK 4	09
5	OUTCOMES OF THE INTERNSHIP	10
6	CONCLUSION	15
	REFERENCES	16

LIST OF FIGURES

Figure Number	Figure Name	Page Number
5.4.2.1	Loopback Overview	11
5.4.4.1	Login user API response on loopback	13
5.4.4.2	User API response on loopback	13
5.4.4.3	Home page of ITemplate	14
5.4.4.4	Email sent for registering users	14

CHAPTER 1

INTRODUCTION

The Internship was conducted in association with Bangalore-based mobile application development organization, PACE Wisdom Solutions. The internship focused on a variety of topics related to web including database management using NoSQL, back-end development and RESTful services using APIs. JavaScript, which is currently one of the most popular programming languages in the world, was used as a medium for application development. Various features of Web development were explored throughout the internship. In addition, cloud deployment using the Heroku platform was also explored.

1.1 ORGANIZATION

The rest of the chapters aim to provide a retrospect into the process of learning that was adapted during the internship. Chapter II gives an elaborate view about the company, the services provided and the various clients associated with the organization. Chapter III focuses on the technologies covered during the entire internship. Further, in Chapter IV, a clear explanation of the tasks performed at the Internship is presented. Finally, Chapter V concludes the report.

CHAPTER 2

ABOUT THE COMPANY

The current chapter gives an elaborate view about the company, PACE Wisdom Solutions. The objectives of the company, services provided by the organization and the various clients of the organization are presented in this chapter.

2.1 SUMMARY OF THE COMPANY

PACE Wisdom Solutions is a Bangalore-based mobile application development company. Since 2012, the company has been building and designing mobile apps for sectors of all kinds. The company started off as a small team and now employs more than 60 employees. The organization is powered by a comprehensive team of mobile-application developers that consists of experts in strategizing, planning, designing, coding and testing. The company has served over 80+ clients spread across 10 countries worldwide. The company has grown over the years and currently serves many clients in domains like Mobile Application Development, Internet of Things, Machine Learning, Web Application Development, etc.

2.2 SERVICES PROVIDED

The company provides Enterprise Application Development and designs them in all shapes and sizes, right from conception to deployment. The applications developed are thoughtfully engineered for the ever-evolving tech world. The company works close with clients and involve them throughout the development process. The company follows a strategy, design, development and testing process that help develop applications with ease. The various domains covered by the company are:

- **Technology Consulting:** The Company provides real world implementation plans for client ideas. It follows an elaborate research process to create comprehensive implementations to client-ideas.

- **Problem Solving:** The Company collaborates with clients and helps develop solutions to the problems faced throughout the development and after.
- **Internet of Things:** The Company provides services to build comprehensive IoT applications for its clients.
- **Mobile Application Development:** The Company develops robust applications for different mobile operating systems such as iOS and Android.
- **Web Application Development:** The Company helps build robust and powerful Web applications for its clients.
- **Machine Learning Solutions:** The Company works with various Machine Learning concepts and provides powerful Machine Learning solutions to the clients.

2.3 CLIENTS

The Company has served multiple clients across the world and helped them build comprehensive enterprise applications. Some of the clients of the Company are Samsung, Affine, mscripts, Philips, Medi Assist, IQVia, Store King, Assetz, Zinnov, Monsanto and many more.

CHAPTER 3

TECHNOLOGIES COVERED

The Internship focused on a variety of Web-based technologies. The various technologies covered throughout the Internship are explored in this chapter.

3.1 JAVASCRIPT

JavaScript (JS) is a lightweight interpreted or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. JavaScript is used to develop comprehensive client-side logic and since 2009, JavaScript has been used to build powerful server- side scripts.

3.2 NODE.JS

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Typically, JavaScript is used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript to write Command Line tools and for server-side scripting-running scripts server- side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server side and client-side scripts.

3.3 LOOPBACK.JS

Loopback.js, or simply Loopback, is a web application framework for Node.js, released as free and an open source software developed by IBM. It is designed for building web applications and APIs. It is built on express js and is relatively easier to use and understand.

3.4 MONGODB

MongoDB is a free and open source platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata. MongoDB is developed by MongoDB Inc., and is published under a combination of the Server-Side Public License and the Apache License. MongoDB accounts of unstructured data which is not the case in SQL-based database management systems.

3.5 RESTful APIs

A RESTful API, also referred to as a RESTful web service, is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development. REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST leverages less bandwidth, making it more suitable for internet usage. An API for a website is code that allows two software programs to communicate with each another. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application. A RESTful API breaks down a transaction to create a series of small modules. Each module addresses a particular underlying part of the transaction. This modularity provides developers with a lot of flexibility, but it can be challenging for developers to design from scratch.

Chapter 4

TASKS PERFORMED

The Internship had weekly catch-up sessions with trainers where the interns had to report progress to their respective trainers. Each week, various technologies were slated to be completed and progress was recorded. A comprehensive report of commencement of the Internship is provided in this chapter.

4.1 WEEK 1

The first week of the Internship focused on introduction to the organization and the various services provided by the organization. Further, the plan for the entire internship was clarified and a flow of activities was slated. The first week further focused on introduction to the programming language to be used for development and also an introduction to the various aspects of backend development.

4.1.1 INTRODUCTION TO JAVASCRIPT

The Internship began with a brief introduction into JavaScript. The various syntaxes, Development methodologies and the overall flow of JavaScript development were explored. The asynchronous flow of JavaScript programming was properly explored. Mini projects were done to better understand the language.

4.1.2 INTRODUCTION TO NODE.JS

The Internship further focused on development of robust backend using Node.JS. The various concepts of Node.JS were covered throughout this process. The first task was to create a simple Node.JS server that provided a simple text as a response. Further, basic concepts of Node.JS such as asynchronous flow and server hosting were covered. The following script is used to start a server in Node.JS:

```
portNumber=3000
app.listen (portNumber, () =>{
console.log(' Server Ready! and listening to post ${portNumbe}!'); }
```

4.2 WEEK 2

The second week of the Internship focused on exploring more concepts of Node.JS and further focus was shifted towards Express.JS. Further, the various concepts of NoSQL databases were explored with the introduction to MongoDB were covered.

4.2.1 INTRODUCTION TO LOOPBACK.JS

The Internship further proceeded to provide a deeper understanding of Node.JS. Various concepts like promises, callbacks and model creation. The concepts of installation of packages to use their services in the application were also explored. Further, an introduction to the Node.JS framework, Loopback js was provided. Loopback js makes the development process easier by initializing the necessary requirements to build a web server. Routing and redirection were covered in the same week. The following code snippet is used to install the loopback package in our system.

```
Npm install -i loopback/cli
```

4.2.2 INTRODUCTION TO MONGODB

The Internship further went on to focus on database management system, MongoDB which is based on NoSQL philosophy for storing data. The installation process, initialization and integration with Node.JS were done in this week. An elaborate view of the document-based data storage of MongoDB was analyzed and understood. Further, basic operations like creation of data models using the MongoDB schema were explored.

At its core, MongoDB presents itself as an open source designed for improvement and scaling ease and a document-oriented database. Each record is actually a document in MongoDB. Documents are stored in a JSON-like format, Binary JSON in MongoDB. BSON documents are the objects that contain an ordered list of saved elements. Each element is composed of a field name and a specific type of value[1].

4.3 WEEK 3

The third week of the Internship focused on the industry level projects that were to be assigned

to the interns. Peers were assigned a head for their respective project, who were very experienced in the respective fields. Further, an introduction to project was given.

4.3.1 INTERFACING LOOPBACK.JS AND MONGODB

The Internship further proceeded to provide a deeper understanding of interfacing between Loopback.JS and MongoDB. Loopback has a mongodb connector to bind the two. The connector is used to specify which type of datasource are we using for the application. A web application with functionalities like creating, retrieving, updating and deleting data was constructed from the scratch. The model file included the methods for creating the basic operations.

The connector helps in recognising the database and in loopback there are many such connectors including in memory space .Different models can have different datasources but it is advisable not to include many as it may lead to a confusion.The following code is the datasource which is mongodb and how its connected to loopback.

```
“db”:{  
  “host”:"localhost",  
  “port”:27017,  
  “database”:"db2",  
  “name”:"db",  
  “newURLParser”:true,  
  “connector”:"mongodb"  
}
```

4.3.2 INTRODUCTION TO RESTful APIs

An introduction was provided into the various concepts of creating and using RESTful APIs. The sessions began with an elaborate introduction into the Service oriented architecture paradigm where multiple applications could render and obtain services from each other through the internet. A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data. A RESTful API, also referred to as a RESTful web service, is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development. A RESTful API breaks down a transaction to create a series of small modules. Each module addresses a particular underlying part of the transaction [3]

4.4 WEEK 4

The fourth week focused on implementation of RESTful APIs using loopback. The various concepts of building RESTful services using loopback were discussed. The REST APIs were connected to the model file and the methods in the model file were used to communicate with the database. A persisted model is REST api.

4.4.1 DEEPER VIEW INTO LOOPBACK WITH MONGODB

The Internship further proceeded to provide a deeper view into persisted models in Loopback. The following code snippet represents various persisted models is given below:

```
"User": {
  "dataSource": "db",
  "acls": [{
    "principalType": "ROLE",
    "principalId": "$everyone",
    "permission": "ALLOW",
    "property": "mail"
  }],
  "public": true,
  "options": {
    "emailVerificationRequired": true
  }
},
"AccessToken": {
  "dataSource": "db",
  "public": false
},
"ACL": {
  "dataSource": "db",
  "public": false
},
"RoleMapping": {
  "dataSource": "db",
  "public": false,
  "options": {
    "strictObjectIDCoercion": true
  }
}
```

Chapter 5

OUTCOME OF THE INTERNSHIP

The Internship aimed at providing clarity regarding back-end development for robust Web applications. It provided clear view into the development process in loopback and MongoDB. Various mini-projects were developed to enhance the skills acquired throughout the internship. Furthermore, RESTful services were built using loopback. The Internship provided knowledge that was helpful in developing many projects that followed. Description about the projects and the outcomes of the Internship are presented in this chapter.

5.1 UNDERSTANDING OF BACKEND DEVELOPMENT USING NODE.JS

The Internship was beneficial in terms of presenting the various concepts of web backend development using Node.JS and loopback. Since Node.JS is a trending language in the current industry, it is truly advantageous to learn and have hands-on experience with the language. Various higher-level concepts like rendering, promises, asynchronous programming, etc., were covered throughout the Internship.

5.2 UNDERSTANDING OF MONGODB

The data generated every day in the real world is unstructured. NoSQL databases allow storage and processing of data in a very easy way. MongoDB is one of the most popular and in- demand NoSQL database management systems and is on a fast track in becoming one of the most used databases. The knowledge about MongoDB acquired through the Internship could be beneficial while stepping into the industry.

5.3 UNDERSTANDING OF RESTful APIs USING LOOPBACK

The Internship helped elaborately understand RESTful APIs and the Service Oriented Architecture. Hands-on experience with creation of RESTful APIs using loopback was helpful in understanding the process for the same.

5.4 PROJECTS DEVELOPED

The acquired knowledge through the Internship successfully helped us build a project entitled ‘I Temple’.

5.4.1 I Temple

The project ‘ITemple’ is a web-based application that allows a user(temple) to opt for various services provided . The database is provided as a service to the users through restful APIs. These restful APIs are connected to the front end through which the both users and professionals get a beautiful UI for interaction. The application has three types of roles one being the super admin role and the other two being temple admin and temple staff. The admin also has the privilege to register a staff and manage the staff under them. The role of super admin is to manage both admin and staff ,is the one with highest authority. The api has 3 models to support these roles they are plans ,users,subscription.The entire application was built using node.js, loopback, mongo dB and gatsby,strapi as the front end.

5.4.2 Loopback overview

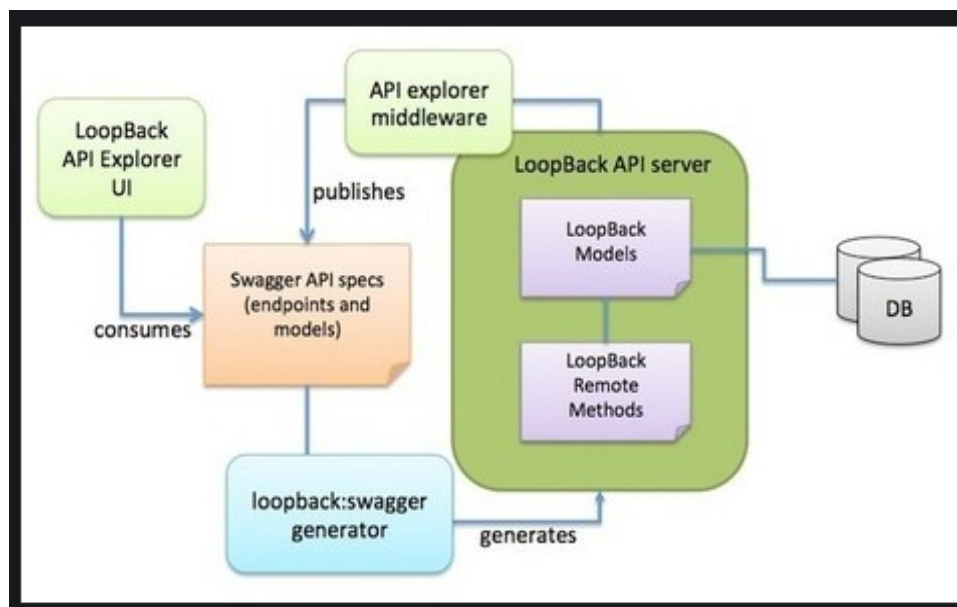


Fig 5.4.2.1 loopback overview

5.4.3 API ENDPOINT

1. User:

INPUTS:

1. UserName
2. Email
3. Password
4. EmpId

JWT token: Generated in response body payload as “token”

Admin Signup API - /api/users
Admin Login API - /api/users/login
Admin reset API - /api/users/resetpassword
Admin Update API - /api/users/update
Admin Delete API - /api/users/delete
Get All Admins API - /api/users/getall
Get One Admin API - /api/users/getbyId

2. Plans:

INPUTS:

1. PlanName
2. Price
3. Validity

JWT token: Generated in response body payload as “token”

Plan add API - /api/plan
Plan Update API - /api/plan/update
Plan Delete API - /api/plan/delete
All plans API - /api/plans/getall
Get One plan API - /api/plan/getbyId

3. Subscription:

INPUTS:

1. Planid
2. Empid
3. Expiry date

JWT token: Generated in response body payload as “token”

Subscription add API - /api/subscription
Subscription Update API - /api/Subscription/update
Subscription Delete API - /api/Subscription/delete
Get All Subscription API - /api/Subscription/getall
Get One Subscription API - /api/Subscription/getbyId

5.4.4 API RESPONSES:

LoopBack API Explorer

Token Not Set

accessToken

Set Access Token

Response Content Type

application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
credentials	<div><pre>{ "username": "rshen", "password": "tmp000"} </pre></div>		body	Inline Model {}
include		Related objects to include in the response. See the description of return value for more details.	query	string

Parameter content type:

application/json

Try it out!

Hide Response

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "username": "rshen",
  "password": "tmp000"}' 'http://localhost:3000/api/Users/login'
```

Request URL

http://localhost:3000/api/Users/login

Fig 5.4.4.1 Login user API response on loopback explorer

LoopBack API Explorer

Token Set

9l8HdiwHaDbVaFwBB8fV2

Set Access Token

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "username": "rshen",
  "password": "tmp000"}' 'http://localhost:3000/api/Users/login'
```

Request URL

http://localhost:3000/api/Users/login

Response Body

```
{
  "id": "9l8HdiwHaDbVaFwBB8fV2Nn3jn2unTrHx26ymeSGPsIgVn7fg1X7kXXm8FjVDy1",
  "ttl": 1209600,
  "created": "2020-07-03T18:14:16.813Z",
  "userId": "5eff748ab4b22b2a7c31a46e"
}
```

Response Code

200

Response Headers

```
{
  "access-control-allow-credentials": "true",
  "access-control-allow-origin": "http://localhost:3000",
  "connection": "keep-alive",
  "content-length": "160",
  "content-type": "application/json; charset=utf-8",
  "date": "Sat, 03 Jul 2020 18:14:17 GMT"
}
```

Fig 5.4.4.2 Token set for user API response on loopback explorer

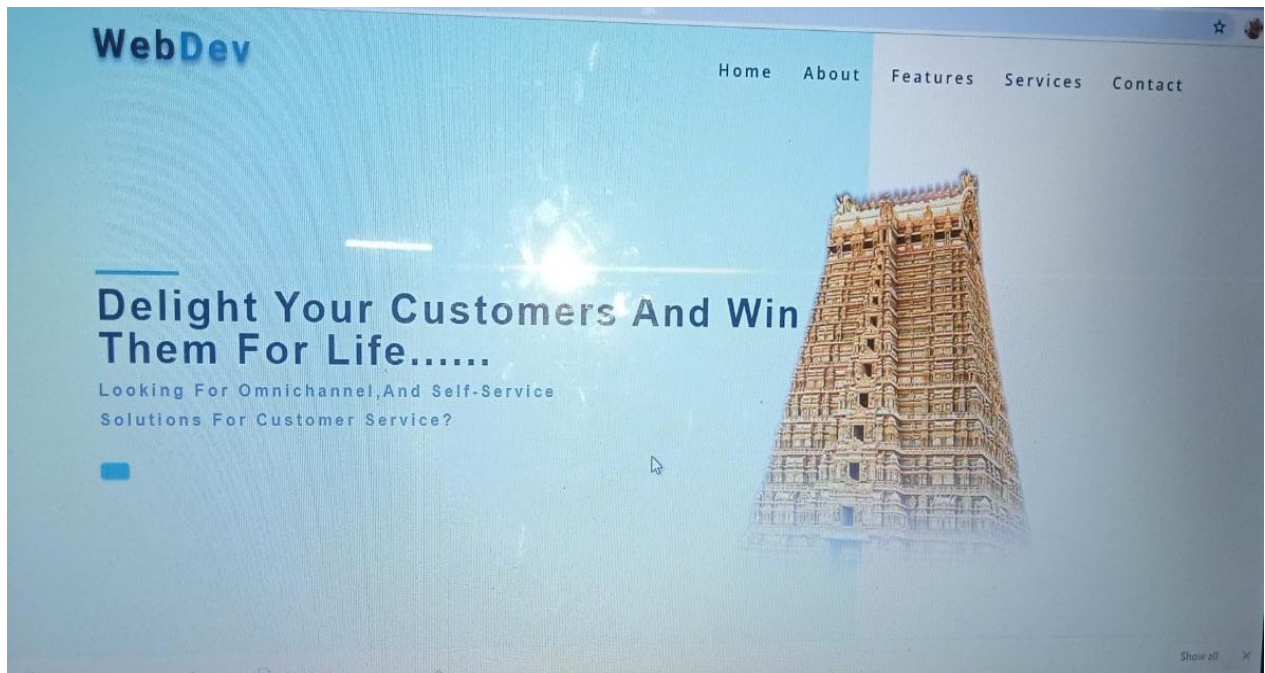


Fig 5.4.4.3 Home page of ITemple

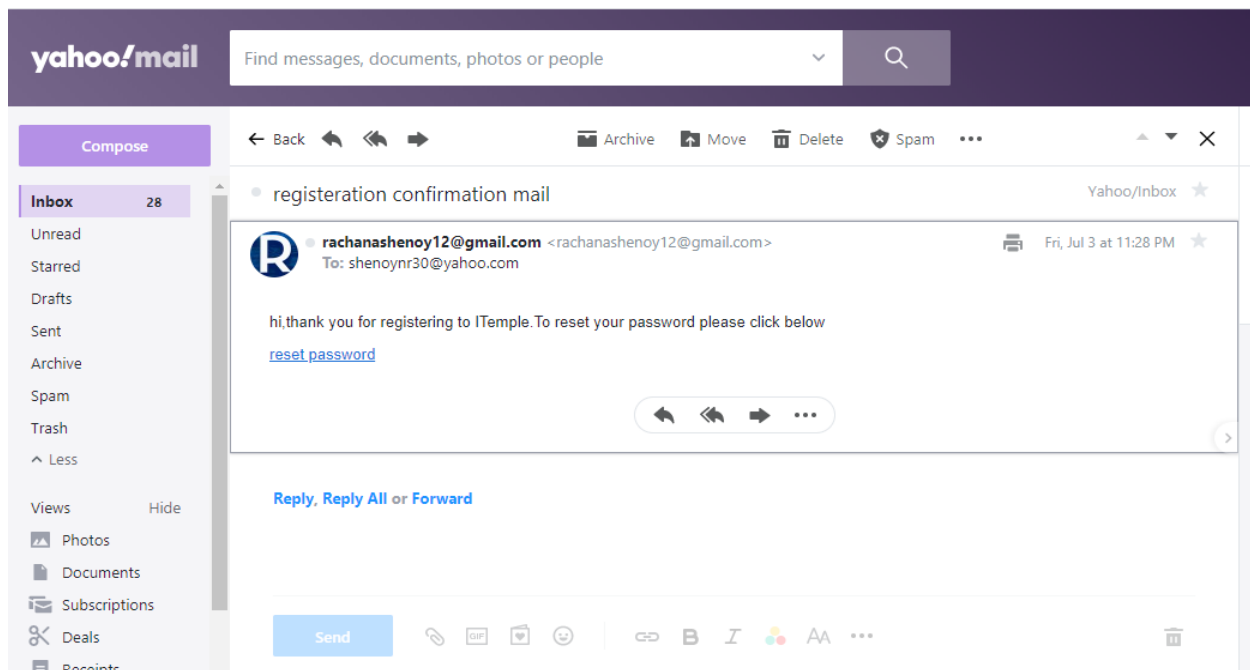


Fig 5.4.4.4 Mail sent after registration of users

CHAPTER 6

CONCLUSION

The Internship was successful in providing a clear view into various concepts and methodologies used while developing Web Applications using Node.JS as the programming language. Further it provided a clear distinction between traditional backend environments and service-oriented structure using REST APIs. Hands-on experience on REST API creation using loopback was obtained through the Internship. It also provided a view into the MongoDB databases and the NoSQL paradigm.

REFERENCES

- [1] <https://dzone.com/articles/nosql-concept-and-mongodb>
- [2] <https://loopback.io/doc/en/lb3/User-management-example.html>
- [3] <https://searchapparchitecture.techtarget.com/definition/RESTful-API>
- [4] <https://loopback.io/doc/en/lb3/Managing-users.html>