Daily Expense Manager

Using MEAN Stack

By

Rajat Singhal(14BCE094)

Sarath Kaul(14BCE104)



DEPARTMENT OF COMPUTER ENGINEERING

Ahmedabad, 382481

Daily Expense Manager Using MEAN Stack

Minor Project

Submitted in fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Engineering

By

Rajat Singhal(14BCE094)

Sarath Kaul(14BCE104)

Guided By

Prof. Vishal Parikh

[DEPARTMENT OF COMPUTER ENGINEERING]



DEPARTMENT OF COMPUTER ENGINEERING

Ahmedabad, 382481

CERTIFICATE

This is to certify that the Minor Project entitled “Daily Expense Manager Using MEAN Stack” submitted by Rajat Singhal(14BCE094) and Sarath Kaul(14BCE104), towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Engineering of Nirma University is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Prof. Vishal Parikh Dr. Sanjay Garg

Dept. of Computer Engg.,

Department of Computer Engg., Institute of Technology,

Institute of Technology, Nirma University,

Nirma University, Ahmedabad

Ahmedabad

ABSTRACT

First of all what is MEAN Stack as it is included in our title. MEAN is an acronym for MongoDB, ExpressJS, AngularJS and NodeJS. It is one of the most recent full stack which is being popularly used for developing we applications and websites. For this particular full stack everything from front to backend is written in JavaScript. We have tried to develop an application Daily-Expense-Manager using MEAN Stack which can keep track of daily expenses of a person. Moreover it provide features of editing, deleting transactions, along with maintaining different accounts of a particular user.

ACKNOWLEDGEMENT

The completion of our Minor Project topic “Daily Expense Manager Using MEAN Stack” wouldn’t be possible without the immense support and guidance of our guide Prof. Vishal Parikh. He suggested several features to add in our project and helped us to implement the same.

We would also like to thank all our friends who helped us to complete the project.

Rajat Singhal(14BCE094)

Contents

Certificate 3

Abstract 4

Acknowledgement 5

1. Introduction 7
2. Technical Details
   1. Tools Used 8

2.2 Technologies 8-10

1. Features 11
2. Requirements to Run Web-App 12
3. Implementation 13-16
4. Database Creation 17
5. Conclusion 18

**1.0 Introduction::**

In this fast growing world managing money has become one of the most important tasks. Everybody tries to keep track of the money they are spending on daily basis. Our web application tries to address the above mentioned concern. In our application we have to tried incorporate basic features of a daily-expense-manager. Some of the features include creating and logging in account which incorporates all the necessary security features. Than a user can add, edit and delete a transaction. Moreover user can create different accounts and can manage transactions according to account.



Fig.1 MEAN Stack

**2.0 Technical Details::**

2.1 Tools Used:

1. Node.js
2. Atom
3. GIT
4. MongoDB

2.2 Technologies:

1. NodeJS – Node.js is a Javascript runtime built which was developed on Chrome. The package npm which is developed for node.js is considered to be one of the largest open source libraries in world. It is lightweight and efficient due to event-driven nature of nodejs. It tends to follow the non-blocking I/O model. NodeJS also contain libraries which helps it to run the javascript applications outside the browser. The main use of nodejs is to develop server side web applications.



Fig.2 NodeJS

1. AngularJS: AngularJs is a framework which was developed to build web-apps. It allows us to use HTML as the base to develop the application and we can further extend its feature to make our components more attractive.

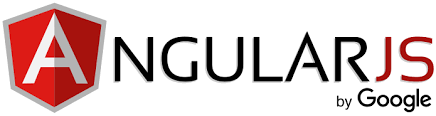


Fig.3 AngularJS

1. MongoDB: It is a NoSQL database which is a cross platform database. It provides high performance, availability and can be easily scalable.



Fig.4 MongoDB

1. ExpressJS: It is a nodejs web app framework which provides various features for mobile and web applications.



Fig.5 ExpressJS

**3.0 Features:**

(a)Register

User can register if he is not having a account.

(b)Login

User can login to the account he/she has previously made.

(c)Profile

All the basic information of user is depicted on this page.

(d)Editing Previous Transactions

User can see all the recorded transactions and can edit or delete the added transaction.

(e)Adding new Transaction

New transactions can be added providing different required data.

(f)Account Creation

Different accounts can be created.

**4.0 Requirements:**

1. Web Browser
2. MongoDB
3. AngularJS
4. ExpressJS
5. NodeJS
6. Internet Connection

**5.0 Implementation:**

* Login/SignUp Page

We have first implemented the login/ signup page to keep track of the users which are using our web-app.

We have used different modules like login and register to implement the same.

We have defined authenticate service to authenticate the user at the time of login and if any wrong info is provided than appropriate flash messages are generated.

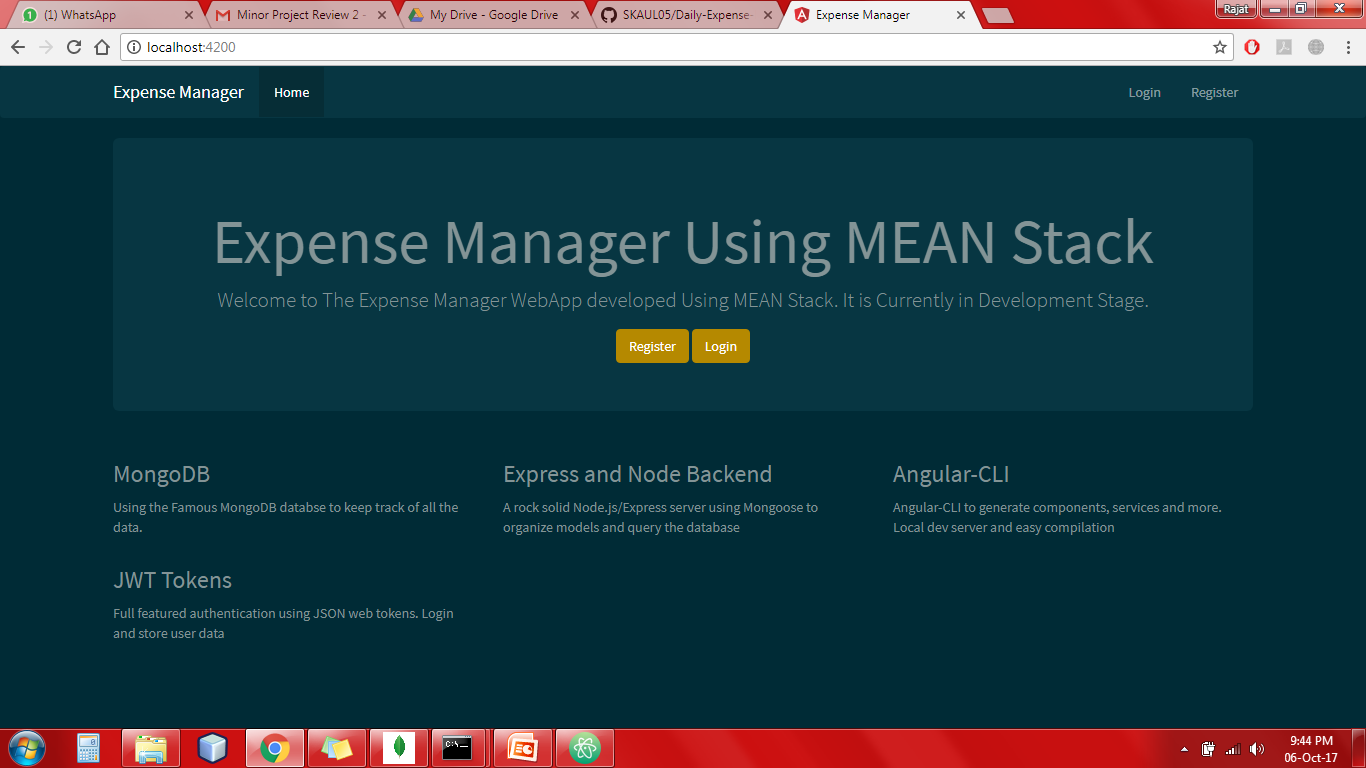


Fig.6 Home Page

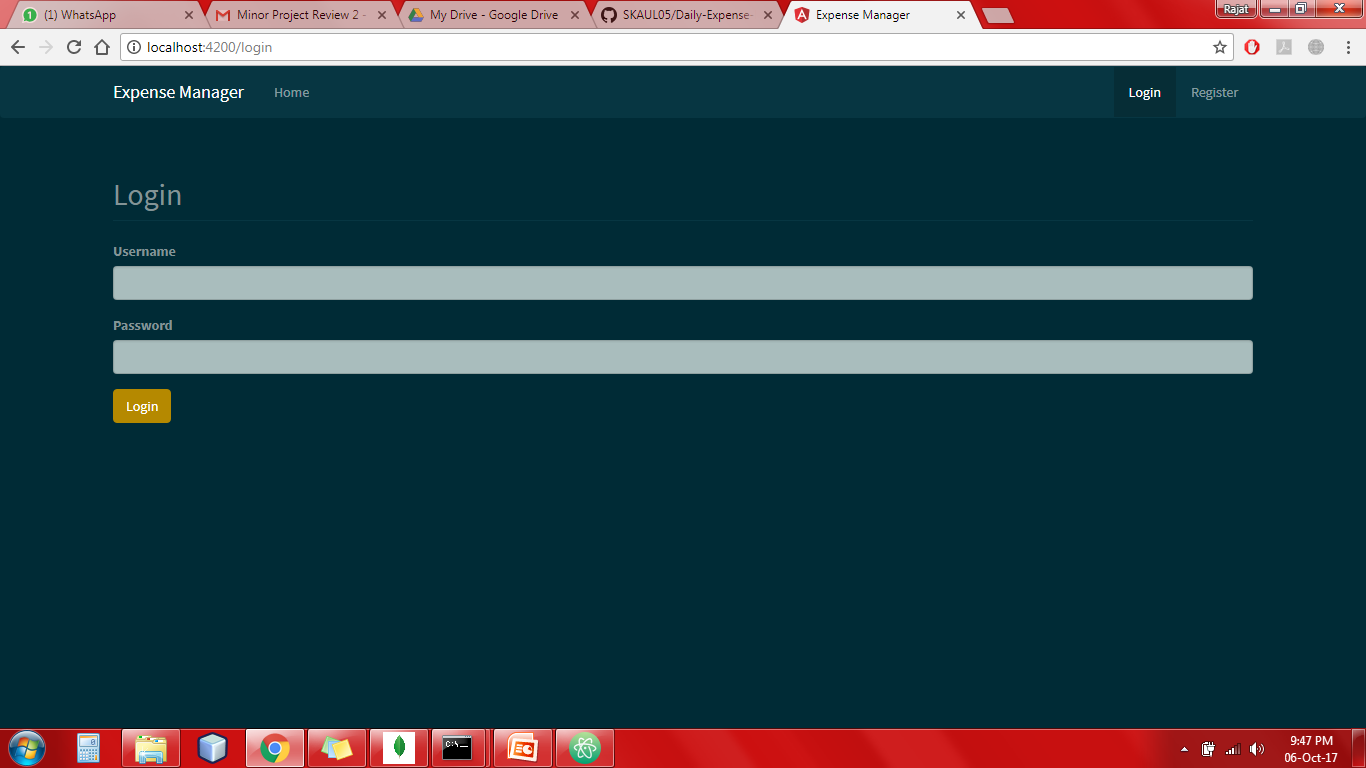


Fig.7 Login Page

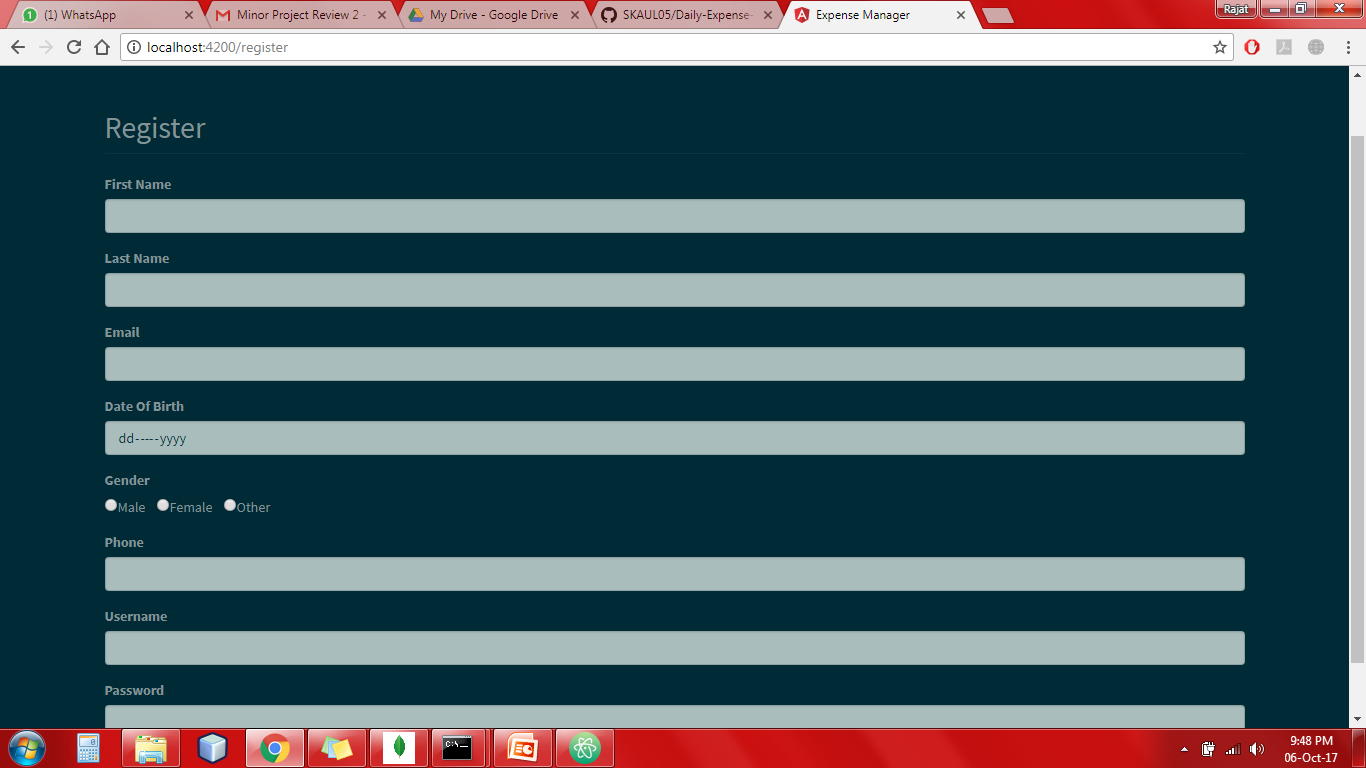


Fig.8 Sign-Up Page

* After successfully logging-in the account user is moved to dashboard where there are several option which are available to the user.

Profile: where all the personal details of user is displayed.

Transactions: To add transactions to particular account here user can add, modify and delete the transaction.

Accounts: To add different Accounts which user wants to keep track.

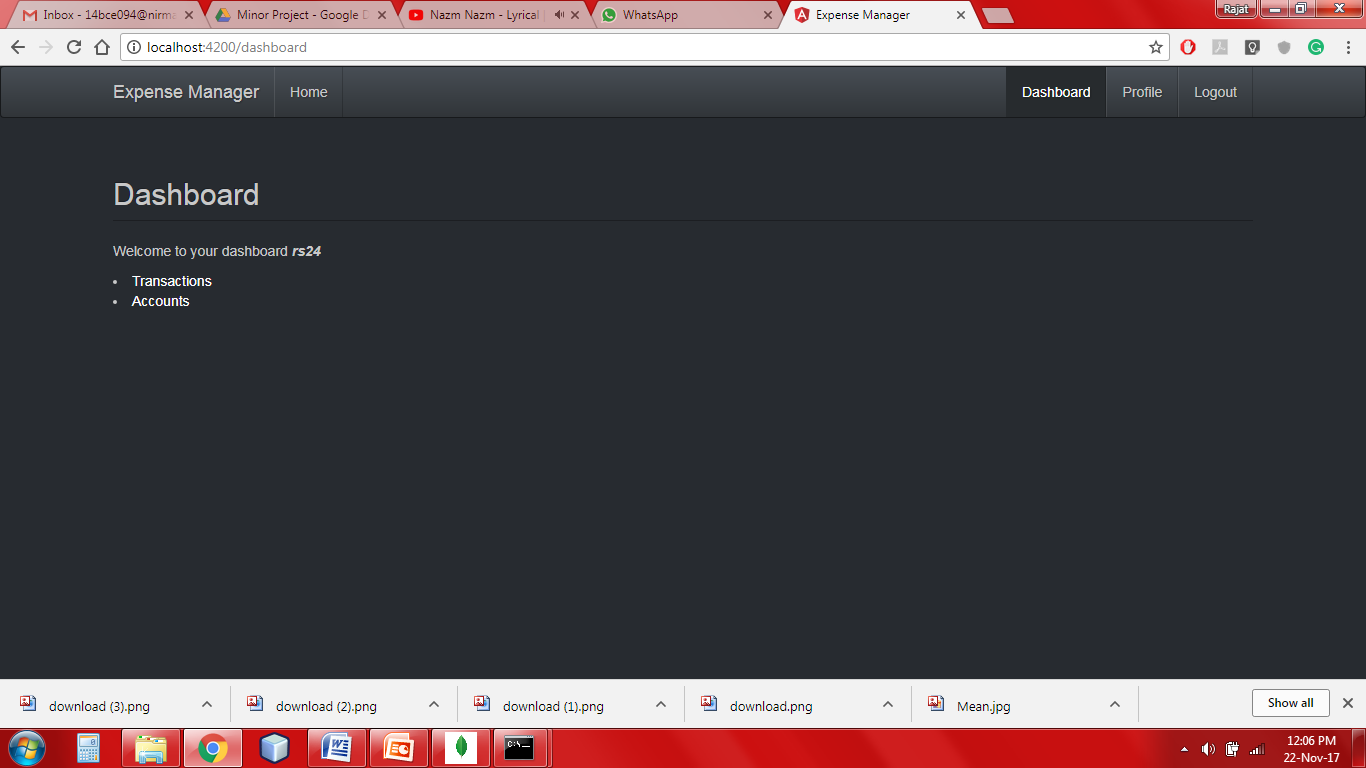


Fig. 9 Dashboard

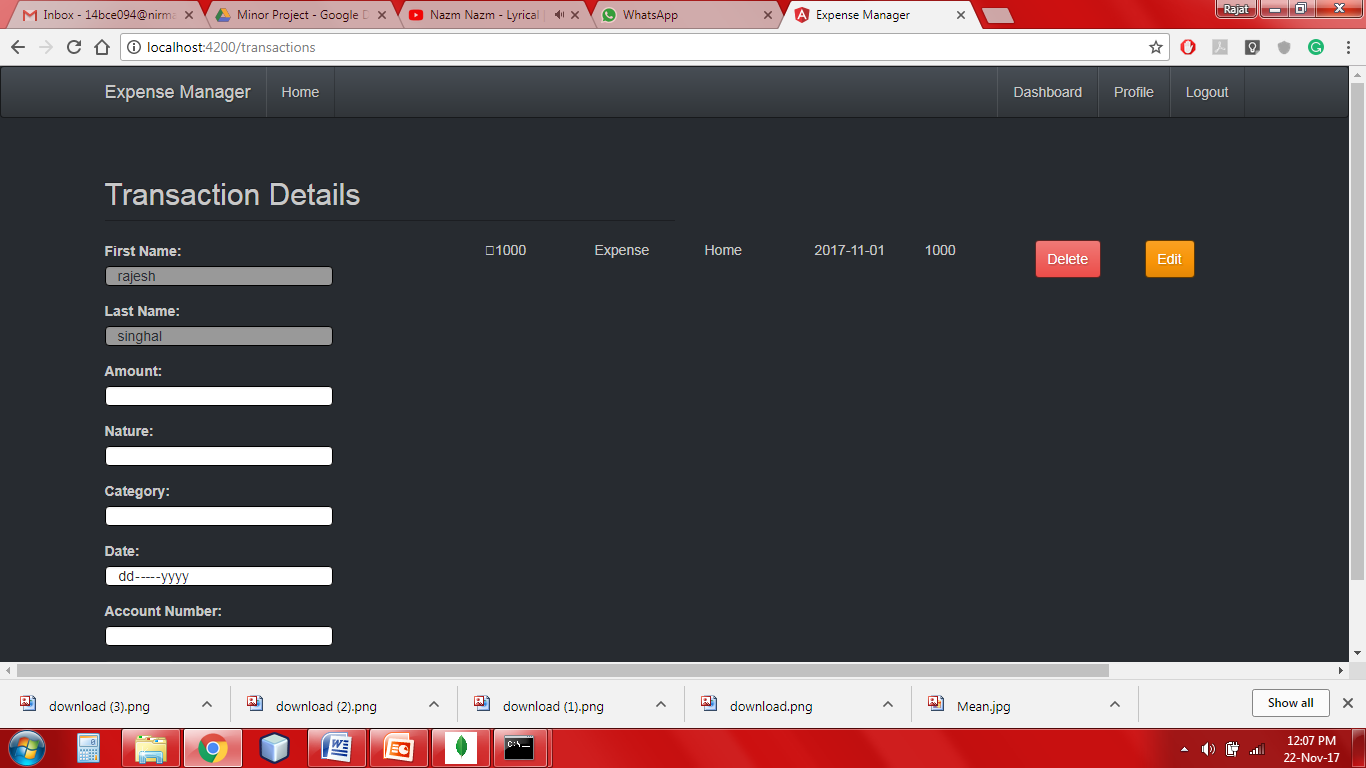


Fig.10 Transaction Page

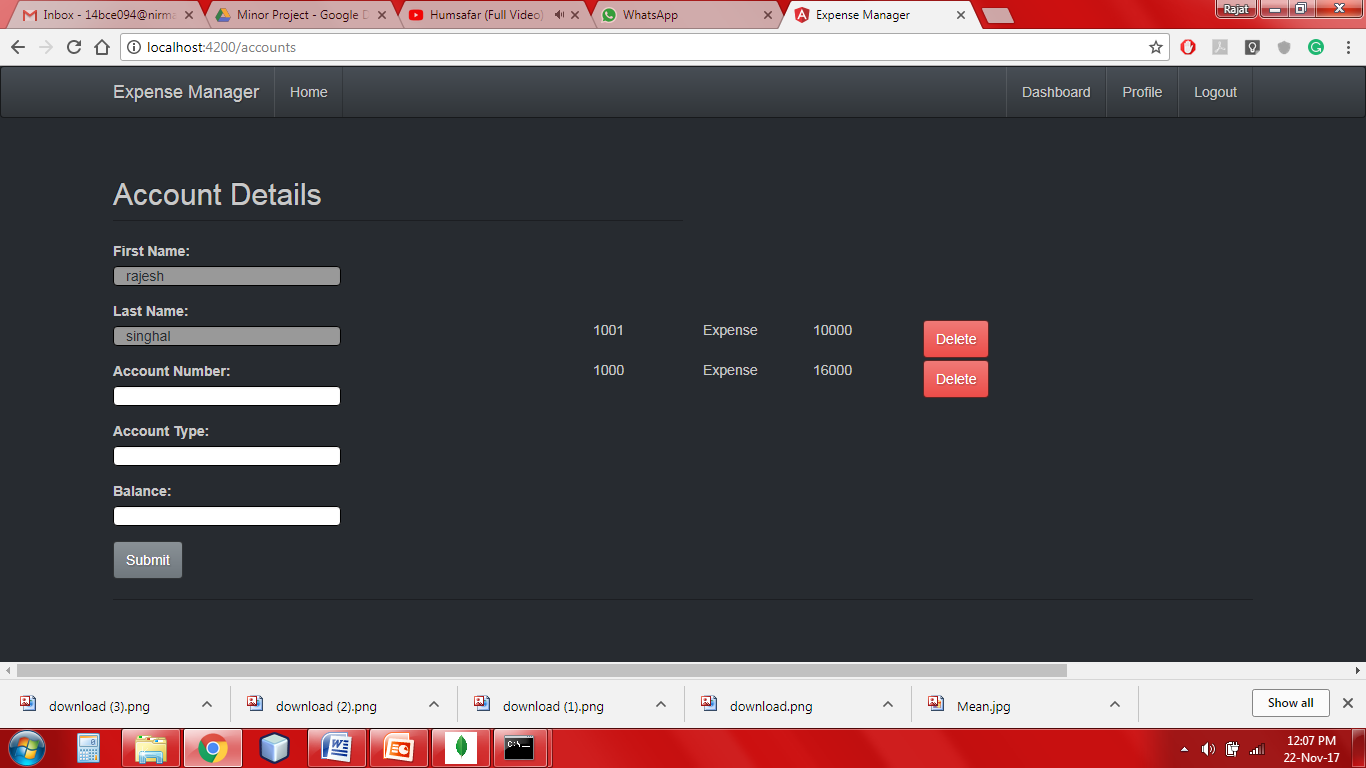


Fig.11 Accounts Page

**6.0 Database Creation:**

* We have generated our database using mongodb.
* There is an unique e-mail Id for every customer which distinguish a user from other users.
* Corresponding to a user different fields regarding personal information is added.
* Apart from personal information there is transactions associated to every customer which is fetched using the e-mail id which is unique.
* We have kept the username fixed while adding a new transaction as t would not be changing.
* The username is prefetched in the fields and cannot be modified.
* After adding a transaction we can modify it all the fields will be prefetched and we can do the desired modification.

**7.0 Conclusion**

After completing this project I was having good exposure towards MEAN Stack Applications. Doing this project helped me to learn something which is not embedded in our syllabus. In future it will be much easier for me to develop web-app.