The Rent-it Application

Gowthamy J, B.V.Sai Mohan, R.Pranay Sharma, Shashank Shekhar, Aastha Sharma

SRM Institute Of Science And Technology Ramapuram,Chennai

Abstract:

The online rent-it application has an abundance of items ranging from luxury to budget for renting. It offers online rental service for corporate houses, local customers and anybody with a requirement to rent. The application offers the best of rates depending upon the facilities availed and offer both inter and intra city facilities. The online rental system provides complete functionality of listing and booking items online. The online rental database system provides the best rent services online. So user can rent from anywhere anytime as per their requirement. No need to go visit rental office to rent an item. That way this system can save customers time as well as employee's time and provide easy and convenient service to customers.

Keywords— ***E-renting, Internet, E-strategies, Online Trading, Web Application, Ajax***

1. INTRODUCTION

Companies in the 21st century are geared towards a world in which those who are best at adapting to our current times are those that can survive in an exponentially connected world. Technology is rapidly changing our world, and companies are competing in a more digitally- driven manner. Over the next few decades, it is expected that there will be a 10% growth in Internet-based companies. These will become one of the biggest sectors of the economy. There are many trading platforms nowadays but there is no good platform designed for direct renting, primarily for common peoples directly to other peoples within their university or city. Such a need arises in a social network where items should be rented or exchanged easily with a community. The famous websites such as Amazon, eBay or Olx are too global in nature and does not support the renting of the goods and services.

An online web application called "Rent-it" has been developed. It is a state-of-the-art platform for direct consumer-to-consumer renting over the Internet. The platform is targeted for direct consumer-to-consumer renting among common peoples. The items for renting include books, household items, electronics, housing rental, vehicles, sports equipment and tutoring services. The web application design needs to be modern, fast, and very simple to use. It is developed using Node.js, Express.js, HTML, CSS, Jason, Ajax and PostgreSQL. The main contribution of this paper is on the design intelligence of the Intelligence web application. The objective is to help the user to decide on the renting price of the item. In addition, the web application can also have features of a recommender system [1].

That is, the renting system would also have the intelligence of recommending items or products to a potential renters given his previous rented patterns. The aim of this paper is to show the development of an intelligent, simple-to-use and user-friendly renting platform targeted for direct consumer-to consumer renting among common peoples. This would facilitate easy renting of items among peoples.

2.LITERATURE SURVEY

Node.js is an open-source, cross-platform. Java Script run-time environment for executing Java Script code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML, to be run client-side by a JavaScript engine in the user's web browser. Node.js runs on JavaScript to be used for server-side scripting, and runs scripts server-side to produce dynamic web page content. Then the page is sent to the user's web browser. Consequently, Node.js has become one of the foundational elements of the "JavaScript everywhere" paradigm, allowing web application development to unify around a single programming language, rather than rely on a different language for writing server side scripts.

Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.[1] It is in fact the standard server framework for Node.

In the earlier computing model that is client–server, the processing load for the application was shared between codes. The server and code installed on each client locally. In other words, an application had its own pre-compiled client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server-side code of the application would be required. An upgrade to the client-side code installed on each users workstation, adding to the support cost and decreasing productivity. In addition, both the client and server components of the application were usually tightly bound to a particular computer architecture and operating system and porting them to others was often prohibitively expensive for all but the largest applications. (Nowadays, native apps for mobile devices are also hobbled by some or all of the foregoing issues.)

3. METHODOLOGY

Aiming at the functional requirements of the Renting System, This paper discusses the advantages of using AngularJS to build the front-end framework, the advantages of using NodeJS to construct the back-end Web server, and the performance advantages of storing data based postgre SQL. This paper focuses on the storage solutions of using SQL to store large data and the statistical analysis solutions based on MapReduce. This paper argues on how to build Web services that meet the requirements of large data visualization based on NodeJs. and safety supervision.

4. KEY TECHNOLOGIES

A. NodeJS

Node.js written in C ++ language, is a JavaScript operating environment. Node.js is a JavaScript runtime environment. Node.js uses the Google Chrome V8 engine for good

performance, and also provides a lot of system-level APIs, such as file operations, web programming, and so on. The JavaScript code on the browser side is subject to various security restrictions at run time, and the operation of the client system is limited. Node.js uses event-driven, asynchronous programming, and designed for network services. Node.js design ideas take the event-driven as the core, it provides the vast majority of APIs that are event-based, asynchronous style.

Take the Net module as an example, where the net Socket object has the following events: connect, data, end, timeout, drain, error, close, etc. The developer using Node.js needs to register the corresponding callback function according to its business logic. These callback functions are executed asynchronously, which means that although these functions appear to be registered sequentially in the code structure, they do not depend on the order in which they appear, but rather wait for the corresponding event to fire. The important advantage of Event-driven and asynchronous programming is that make full use of the system resources. The implementation of the code without waiting for a certain operation to complete, and the limited resources can be used for other tasks. This design is very suitable for back-end network service programming, which is the goal of Node.js. In server development, concurrency request processing is a big problem, and blocking functions can lead to the waste of resource and the delay of time. Through event registration, asynchronous function, developers can improve the utilization of resources, and performance will also improve. From the supported module provided by Node.js, we can see that many of the functions, including file operations, are executed asynchronously, which is different from traditional languages.

In order to facilitate server development, Node.js' network modules are particularly large, Including HTTP, DNS, NET, UDP, HTTPS, TLS, etc., developers can build a Web server on this basis.

B. AngularJS

AngularJS is an open source JavaScript library maintained by Google to help a single page application run. Its goal is to enhance browser-based applications with MVC mode (MVC) capabilities, making development and testing easier.

C. Postgre SQL

It is the world's most advanced open source database. PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. PostgreSQL runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. This tutorial will give you quick start with PostgreSQL and make you comfortable with PostgreSQL programming. PostgreSQL, originally called Postgres, was created at UCB by a computer science professor named Michael Stonebraker. Stonebraker started Postgres in 1986 as a follow-up project to its predecessor, Ingres, now owned by Computer Associates.

1977-1985 − A project called INGRES was developed.

From there, over the years many changes have been made to the project and it was re-released as PostgreSQL in 1996.

Establishment of the PostgreSQL Global Development Team

D. Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library. Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

5. PLATFORM IMPLEMENTATION

This section contains a brief technical overview of the Node.js platform.

A. Architecture

The Node.js platform's architecture is notable for its eventbased execution model. It uses a single thread—the event thread—for executing application code. In this aspect Node.js is similar to the Java Disruptor [10] architecture. Running in a single thread means that there is no application level. The 7th International Conference for Internet Technology and concurrency and all incoming requests are handled by the same thread. Without application level concurrency, error-prone and often performance limiting synchronization and locking of shared resources is unnecessary.

On the one hand, this approach simplifies application development. On the other hand, time consuming calculations and blocking I/O calls executed in the event thread prevent the applications from handling other requests. Node.js' answer to this problem is the use of asynchronous I/O. A common pattern in Node.js is for the otherwise possibly blocking I/O functions, e.g., database requests or file system operations requiring disk access, to take a callback function as an additional argument and to return immediately. The runtime then executes the actual I/O operation in background utilizing threads from an internal tread pool and, if available, efficient non-blocking system calls such as epoll. When an operation completes, the corresponding callback function provided by the application is called on the event thread. The Node.js' execution model contrasts to the thread-based execution model where each client connection is handled by a separate thread. The threaded model is employed by conventional web application platforms. In figure 5.1 , architecture of the proposed system has explained.

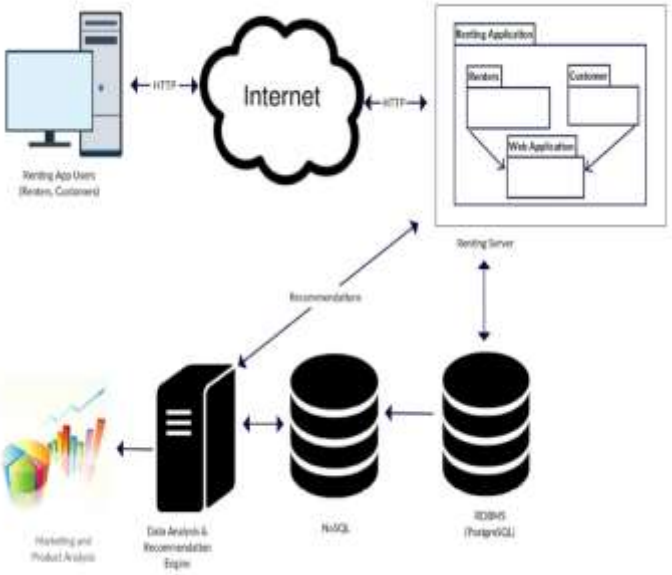

Fig 5.1: Architecture diagram

In principle, the threaded and event-based models for server applications are equivalent in their resource usage and performance, given efficient implementations [11], [12]. While the one-thread-per-request model may be a more natural way for the developers to think about the flow of their applications, the scalability of currently available thread-based mainstream server platforms is less than optimal and they are incapable of handling tens of thousands client connections efficiently without consuming extensive amounts of system resources.

The approach taken by Node.js allows it to use system resources more efficiently compared to platforms with threaded execution models. In case of server applications, more efficient resource utilization usually translates to higher resiliency in case of, for example, denial of service attacks.

B. Language

The platform's native language is JavaScript (ECMAScript) [13]. Historically, JavaScript is known as a basic client side scripting language for web pages. Thanks to the popularity of the language in building dynamic web pages and tight competition in browser market, the language has received many enhancements and the execution engines have become more and more efficient over the years. JavaScript can now be considered a full-featured general-purpose programming language.

JavaScript has some qualities that make is especially well suited for the Node.js platform and have helped the platform to gain in popularity. First of all, probably all web developers are already somewhat familiar with this language. Using the same language on the client side and also on the server side is convenient for developers as there is less to learn to become productive. Secondly, JavaScript is well suited to the callback oriented programming as it supports anonymous functions which capture the referenced values from the context of their definition (closures). Thanks to closures and the overall conciseness of the language, it is generally possible to achieve desired functionality with short and elegant programs. Unfortunately, JavaScript, especially on the server side, also has its weaknesses. We will discuss some of these in Section III. The third circumstance that made JavaScript a reasonable choice for the language is the existence of an efficient execution engine which will be described next. C. Runtime

Node.js provides a runtime environment for the platform, comparable to what the Java Virtual Machine is to the Java platform. Node.js is implemented in C++ and it is based on the Google's V8 JavaScript engine [14] originally developed for the Google Chrome browser. V8 is fast and it employs advanced techniques such as just in time compilation to native code and efficient memory management. The runtime manages events that are processed on the event thread. Internally it also contains a thread pool to give the applications an impression of non-blocking I/O.

D. Libraries and Modules

A programming language without any libraries is useless for practical applications. Therefore Node.js comes with an API covering low level networking, basic HTTP server functionality, file system operations, compression and many other common tasks. Parts of the API have been frozen, but as the platform is still young, other parts are changing, sometimes without backwards compatibility.

The available functionality of Node.js libraries can be extended with add-on modules (packages). The modules are distributed via public or private package registries. The packages are structured according to the CommonJS package format and can be installed with the npm package manager. The main public package registry [15] contains already more than 12 thousand packages contributed by the community.

To conclude the overview of the Node.js platform, we can say that the unique combination of the JavaScript language with an efficient runtime and the event-based architecture make Node.js an attractive platform for application developers. Unfortunately, due to the design decisions, it also has some inherent weaknesses which we will discuss in the next section.

6. PROPOSED MODEL

The model is built in hottest open source web platforms currently available. It's built on Google Chrome's V8 JavaScript runtime engine and it allows you to write all kinds of network applications and servers in just a few lines of code. Node.js uses an asynchronous programming model built on non-blocking I/O and a single-threaded event loop. What this means, basically, is that you don't need to be concerned with awful race conditions or synchronization issues that arise when programming for a concurrent multi-user environment. This workshop is aimed mainly at web development instructors that would like to consider teaching a single programming language, the basic template of registration form has explained in figure 6.1, where the required details of the user is going to be taken. Figure 6.2depicts a model in which different rental services are planned to be offered to customers for fair usuage.

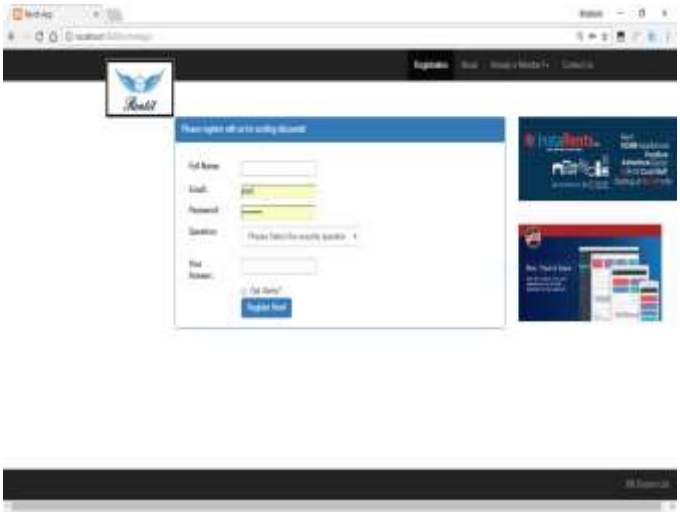

Fig 6.1. Registration layout

JavaScript, for both client-side and server-side coding. Participants will learn how to employ Node.js on Windows, Mac OS or Linux in order to write scalable web servers and applications. Additionally, the Express web framework will be introduced in order to demonstrate how to quickly program traditional webapps and single-page applications (SPA) with the aid of jQuery, AJAX and RESTful web services. The resulting programs will be usable from any modern web browser, including those found in desktop and laptop computers, and mobile devices such as tablets and smart-phones. Participants should have prior working knowledge of client-side (running on a browser) JavaScript and HTML.

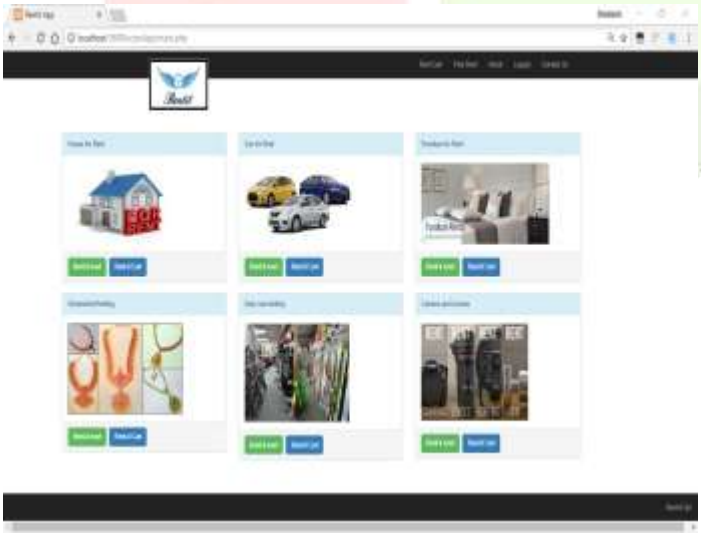

Fig 6.2 proposed page layout

ACKNOWLEDGEMENT

The content which represented in this is based on the base papers which we collected.

REFERENCES

[1] Arto Ojala, "Software-as-a-Service Revenue Models". IEEE IT Professional, vol. 1 5,3 , pp. 54 – 59, 18 July 201 2.

[2] Deng Zhonghua, Hai Erfeng, "Analysis of Sa as-Based E-Commerce Platform". IEEE 20 I 0 International Conference on E-Business and E-Government,.

[3] Wegner. Why Node.JS? Security. [Online]. Available: http: //www.wegnerdesign.com/blog/why-node-js-security/
J. Holowaychuk. Connect – a middleware layer for Node.js. [Online].

[4] Available: https://github.com/senchalabs/connectI.Yugashini, S. Vidhyasri, K.Gayathri Devi, Design And Implementation Of Automated Door Accsessing System With Face Recognition, International Journal of Science and Modern Engineering(IJISME), Volume-1, Issue-12, November 2013.

[5] R. Hansen. Slowloris HTTP DoS. [Online]. Available: http://ha.ckers.org/slowloris/

[6] The slowhttptest project. SlowHTTPTest – an application layer DoS attack simulator. [Online]. Available: http://code.google.com/p/ slowhttptest/

[7] C. Robbins. Node.js module forever. [Online]. Available: https: //github.com/nodejitsu/forever/.

[8] A White, JavaScript Programmer's Reference. John Wiley & Sons, 2010.

[9] Google, Inc. V8 JavaScript Engine. [Online]. Available: http://code.google.com/p/v8/

[10] I. Z. Schlueter. The Node Package Manager and Registry. [Online]. Available: https://npmjs.