



A Project Report Based On

CSE 300 (SOFTWARE DEVELOPMENT PROJECT)

A Productive Browser Extension to utilize browsing time more effectively and efficiently

Submitted By:

SADIA ISLAM SAMI

Roll: 16102018

Registration: 4870

ASHIKUZZAMAN

Roll: 16102020

Registration: 4872

Submitted To:

DR.MD.MIJANUR RAHMAN

Professor

Dept. of Computer Science and Engineering

Jatiya Kabi Kazi Nazrul Islam University

Dept. of Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University, Trishal, Mymensingh, Bangladesh.

June 2019

Certification of Project work

The project field title “A Productive Browser Extension to utilize browsing time more effectively and efficiently” by (**SADIA ISLAM SAMI**, Roll: **16102018**, Registration: **4870**) and (**ASHIKUZZAMAN**, Roll: **16102020**, Registration: **4872**) Session: **2015-16** has been accepted as satisfactory in partial fulfillment of the requirement for the **B.SC** degree in **Computer Science and Engineering**.

Board of the Examiner

1st Examiner

(Signature)

2nd Examiner

(Signature)

External Examiner

(Signature)

Declaration

We hereby declare that we are the sole author of this project. We authorize the **Jatiya Kabi Kazi Nazrul Islam University** to lend this project to other institutions or individual for the purpose of scholarly research. We further authorize **Jatiya Kabi Kazi Nazrul Islam University** to reproduce this project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature of the Candidate

26-07-2019

SADIA ISLAM SAMI

Signature of the Candidate

26-07-2019

ASHIKUZZAMAN

Acknowledgement

All praises are to Almighty Allah for giving us such capabilities for the successful completion of this Project. We would like to express our heartily gratitude to our project supervisor, **Prof. Dr. Mijanur Rahman**, Professor, Department of Computer Science & Engineering, Jatiya Kabi Kazi Nazrul Islam University, Trishal, Mymensingh, for his guidance and supervision during the entire project work. We wish to express thanks to **Prof. Dr. A.H.M. Kamal**, Head, Department of Computer Science & Engineering, Jatiya Kabi Kazi Nazrul Islam University for providing us lab opportunities with lab materials related to our project work. We also thank all of our friends, group member & lab assistant and our family for helping to complete the project of “**Productive Browser Extension**”. Finally, we express our thanks to the Department of Computer Science and Engineering for giving us the opportunity to study here and supporting us greatly through our graduate study.

Authors

Abstract

“**Productive Browser Extension**” is a web based Extension application which gives a platform to maintaining time of internet browsing by many feathers and effective processes. Here management of browsing time is presented. Each participant in the process plays a different role, and cooperates with other participants. We use **Node.js** for backend because it is an open source server environment running on various platforms (Windows, Linux, UNIX, Mac OS X, etc.) which uses **JavaScript** on the serve. A database management system is provided by Firebase is a backend platform for building Web, Android and IOS applications. It offers real time database, different APIs, multiple authentication types and hosting platform by which **HTML** content to end-user. We also use **JavaScript** is the Programming Language for update and change both **HTML** and **CSS** and mainly it can calculate, manipulate and validate data for **Backend Language**. People can easily log-in, submit, and notice browsing details from here. Everyone can gain benefit from this by accessing different types of feathers which prefer huge amount of data of research. Anyone can know the latest update and get notify at any time. The application is also very user friendly and one easily gain benefit from it. After all, the Extension application is flexible and trustworthy.

Table of Contents

Certification of Project work.....	i
Declaration.....	ii
Acknowledgement.....	iii
Abstract.....	iv
Table of figure.....	vii
Chapter 1	1
Introduction	1
1.1 Overview and Introduction.....	1
1.2 Objective.....	1
1.3 Motivation.....	1
1.4 Historical Background.....	2
1.4.1 Browser.....	2
1.4.2 History of Browser.....	2
1.4.3 Browser Extension.....	3
1.4.4 API Conformity.....	3
1.5 Goals of this System.....	3
1.6 Organization of This Report.....	3
Chapter 2	4
Web Browsers and Extensions	4
2.1 Overview	4
2.2 Browsers in the Market	4
2.2.1 Google Chrome Browser	4
2.2.2 Firefox Browser.....	4
2.2.3 UC Browser.....	5
2.3 Extensions in the Market	5
2.3.1 Name of some related extensions or add-ons	6
2.3.2 A Productive Browser Extension.....	7
2.4 Why Productive Browser Extension	7
Chapter 3	8
System Design and Development	8
3.1 Overview	8
3.2 Approaches	8
3.3 Design.....	8
3.4 Incremental model.....	8
3.4.1 System Design.....	8

3.4.2	<i>User Panel Design:</i>	9
3.5	Data Flow Diagram	9
3.6	E-R (Entity Relationship) Diagram	11
3.7	Software Process Model	11
3.8	Incremental Model	12
3.8.1	<i>Advantages of Incremental model</i>	12
3.8.2	<i>Disadvantages of Incremental model</i>	12
3.8.3	<i>When to use the Incremental model</i>	12
3.9	Programming Language	13
3.9.1	<i>HTML</i>	13
3.9.2	<i>CSS</i>	13
3.9.3	<i>JAVASCRIPT</i>	13
3.9.4	<i>JSON</i>	13
3.9.5	<i>Firebase</i>	13
Chapter 4		14
Results and Discussions		14
4.1	Overview	14
4.2	Functionality description of Productive browsing extension	14
4.3	Limitation of this System	16
4.4	Future Scope	16
4.5	Experimental Results	14
4.6	Achievements of this System	16
4.7	Conclusion	16
References		18
Appendix		19
	<i>Link to firebase as online database</i>	19
	<i>Popular marked sites in stat.js</i>	19
	<i>Log out pop up</i>	19
	<i>To do from extension.js</i>	20
4.9.5	<i>Add favorite links to server from server.js</i>	20

Table of figure

Figure 3.1: Data flow diagram.....	9
Figure 3.2: User case diagram	9
Figure 3.3: Class diagram.....	10
Figure 3.4: Database tree diagram	10
Figure 3.5: E-R Diagram	11
Figure 3.6: Incremental model.....	12
Figure 4.1: Home page of user specification	14
Figure 4.2: Log-in and registration information for user specification	15
Figure 4.3: To do list.....	15
Figure 4.4: Browsing time statistics.....	15
Figure 4.5: Favorite links	15

Chapter 1

Introduction

1.1 Introduction

Modern technology has revolutionized the way people all over the world communicate, interact and doing their day with internet. The most important technological change over the past 20 years is the advent and popularization of the Internet browsing. The Internet connects billions of people around the globe and allows a type of connectivity in ways which the world has never seen. Companies are able to do business with consumers from other countries instantaneously, friends and families are able to talk to one another and see each other regardless of location, and information sits at the fingertips of every person with a computer, tablet or phone. People can't think a single moment without technology. From morning to night everybody need help of the technology. This is the revolutionary time of computer technology. Most of the works depends on web application. For this reason anytime, anywhere, anyone can access internet through this extension and people can found the expectable productive browsing time and most update notification of their tasks. "PRODUCTIVE BROWSING EXTENSION" is web based Extension application that help all to know all information of time, location, weather and to do-lists. This web browser extension is openly accessible and can be used by the pc through several browsers like chrome, UC, Firefox etc. The system is an Internet based extension that can be accessed throughout by any location with user id and password because having an online database system (Firebase) connected with this system. All are benefited by this extension because it is collaborated with several features that is easy to use and avoid using many extension for productive browsing.

1.2 Objective

The primary objective of this project is to develop a browser extension for "A PRODUCTIVE BROWSING EXPERIENCE". The objective of this application is to implement a system that gives all opportunity to management all types of to do list scheduling of our user in online. This application helps us to know all information of task to be completed and notify about time, location weather and especially guides procrastinators to finish their tasks in time. This helps us to save a lot of time. This also helps us to access all information quickly. This also helps all to show updated tasks time to time. This also helps all to download and submit all type of researches. Anytime, anywhere, anyone can log in into his account of productive browsing and get the benefits of this extension by internet very quickly. To add any new task with priority it manages without any complexity. This will give quick outcomes.

1.3 Motivation

There is not avail ablation of any extension individual web store that have all features of productive browsing in one extension, that's the reason for "Productive Browsing". Those browsing extensions are

not properly real time with contain insufficient information. For this reasons we develop this web based application. This application contains open access features. Here anyone can easily add, delete and update all tasks dynamically. In this web extension application, there is a very important feature of submitting user id online with real time database experience. The static and local submission system it is occurred many problems such as data loss, no remote login, no real time database. We can't see any updated by real-time. Removing this problem, we want to develop an extension to manage system and browsing time, tasks. This helps us to save our time and do some productive works when we are online.

1.4 Historical Background

1.4.1 Browser

A web browser is a software application for retrieving, presenting and traversing information resources on the World Wide Web. It further provides for the capture or input of information which may be returned to the presenting system, then stored or processed as necessary. The method of accessing a particular page or content is achieved by entering its address, known as a Uniform Resource Identifier or URI. This may be a web page, image, video, or other piece of content. Hyperlinks present in resources enable users easily to navigate their browsers to related resources. A web browser can also be defined as an application software or program designed to enable users to access, retrieve and view documents and other resources on the Internet.

1.4.2 History of Browser

1990 – The Worldwide Web (not to be confused with the World Wide Web) was the first browser ever created by W3C Director Tim Berners-Lee, then renamed Nexus to differentiate from the actual World Wide Web. Unlike today, this was the only browser and the only way to access the web.

1992 – Lynx was a texted-based browser that couldn't display any graphic content.

1993 – Mosaic was the first browser to allow images embedded in text making it “the world's first most popular browser”.

1994 – A noticeable improvement to Mosaic came Netscape Navigator.

1995 – Internet Explorer made its debut as Microsoft's first web browser.

1996 – Opera started as a research project in 1994 that finally went public two years later. This was also arguably the beginning of the browser wars, mainly between IE 3 and Navigator 3 as Internet Explorer inched ahead with new capabilities.

2003 – Apple's Safari browser was released specifically for Macintosh computers instead of Navigator.

2004 – Mozilla launched Firefox as Netscape Navigator faded out.

2007 – Mobile Safari was introduced as Apple's mobile web browser and continues to dominate the iOS market.

2008 – Google Chrome appeared to soon take over the browser market.

2011 – Opera Mini was released to focus on the fast-growing mobile browser market.

2015 – Microsoft Edge was born to combat Google.

Taking it five years further into 2015, the user breakdown is Chrome (64.9%), Firefox (21.5%), Internet Explorer (7.1%), Safari (3.8%), and Opera (1.6%).

1.4.3 Browser Extension

A browser extension is a small software module for customizing a web browser. Browsers typically allow a variety of extensions, including user interface modifications, ad blocking, and cookie management. Internet Explorer was the first major browser to support extensions, with the release of version 5 in 1999. Firefox has supported extensions since its launch in 2004. Opera began supporting extensions in 2009, and both Google Chrome and Safari did so the following year. Microsoft Edge added extensions in 2016.

1.4.4 API Conformity

In 2015, a community working group formed under the W3C to create a single standard application programming interface (API) for browser extensions. While that goal is unlikely to be achieved, the majority of browsers already use the same or very similar APIs due to the popularity of Google Chrome. It was the first browser with an extension API based solely on HTML, CSS, and JavaScript. Beta testing for this capability began in 2009, and the following year Google opened the Chrome Web Store. As of June 2012, there were 750 million total installs of extensions and other content hosted on the store. In the same year, Chrome overtook Internet Explorer as the world's most popular browser, and its market share continued to grow, reaching 60% in 2018.

1.5 Goals of this System

1. It has open access opportunity.
2. User friendly and robust.
3. It is a dynamic system development.
4. It gives easy submit, view and management facilities.
5. It is a web based browser extension application that helps to get more productive while browsing time through internet.

1.6 Organization of This Report

Chapter 1: Discusses about what is the objectives problem statements, motivation and history of this web based application.

Chapter 2: Discusses about the background study of the project. This includes format and types etc.

Chapter 3: Discusses about what productive browsing extension all design of the extension is and how the work is completed.

Chapter 4: Discusses about how the extension is developed and what the content is.

Chapter 5: Discusses the overall result, scope of the system.

Chapter 2

Web Browsers and Extensions

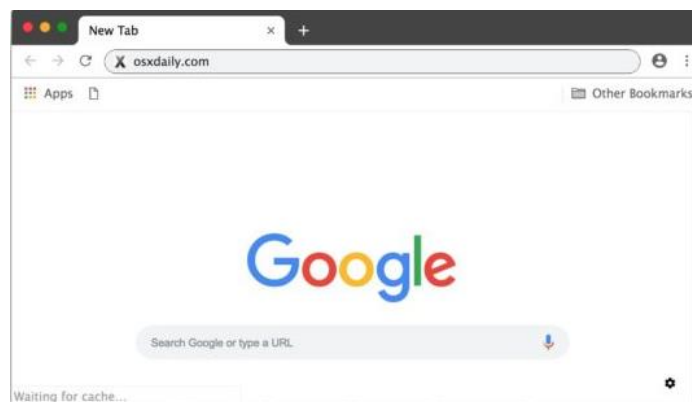
2.1 Overview

In this chapter there will be discussed about the historical background of Productive Browsing Extension, the approaches to implement the extension system and software support for this system. Programming Languages that are used to develop this project, required resources to implement this system of browsing extension will be focused in this chapter.

2.2 Browsers in the Market

2.2.1 *Google Chrome Browser*

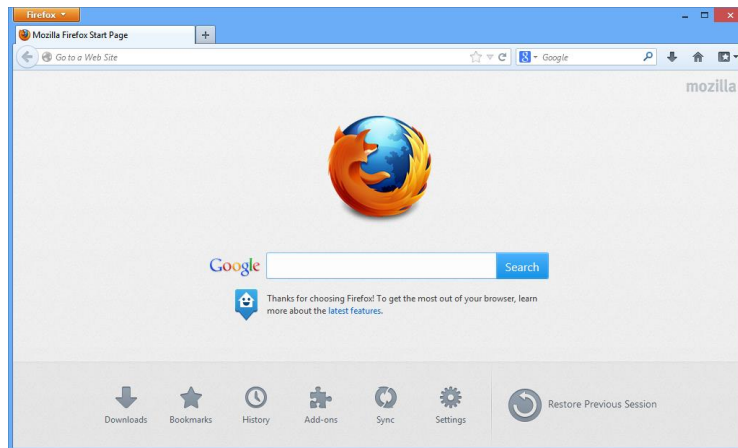
It is an open source program for accessing the World Wide Web and running Web-based applications. The Google Chrome Web browser is based on the open source Chromium project. Google released Chrome in 2008 and issues several updates a year. It is available for Windows, Mac OS X, Linux, Android and iOS operating systems. The Google Chrome browser takes a sandboxing-based approach to Web security. Each open website runs as its own process, which helps prevent malicious code on one page from affecting others (or the computer operating system at large). The browser also supports Web standards such as HTML5 and cascading style sheets (CSS).



2.2.2 *Firefox Browser*

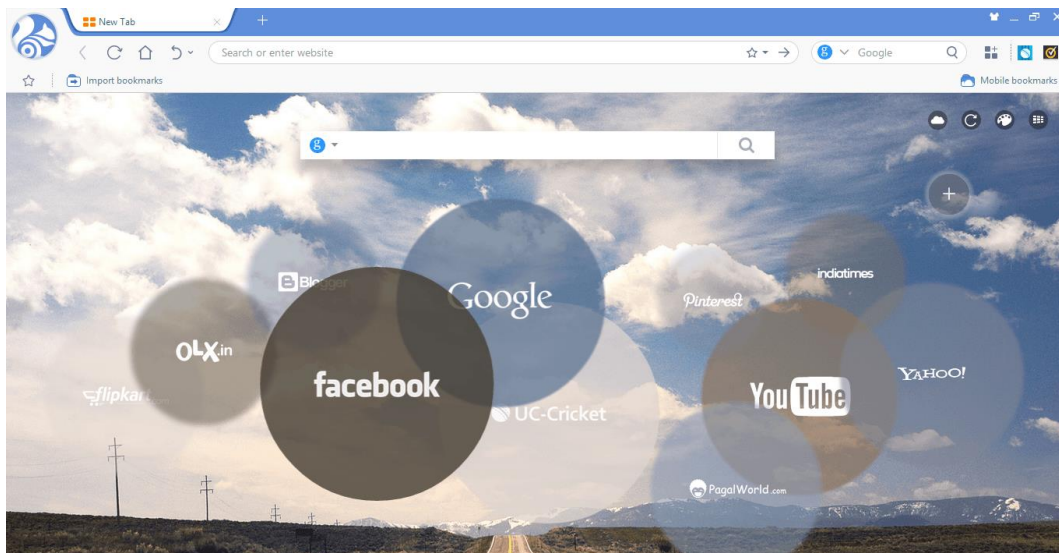
It is a free and open-source web browser but now Firefox is covering a family of products and services developed by the Mozilla Foundation and its subsidiary Mozilla Corporation. Firefox Browser is available for Microsoft Windows, macOS, Linux, BSD, illumos and Solaris operating systems. Its sibling, Firefox for Android, is also available. Firefox uses the Gecko layout engine to render web pages, which implements current and anticipated web standards. In 2017, Firefox began incorporating new technology under the code name Quantum to promote parallelism and a more intuitive user interface. An additional version, Firefox for iOS, was released on November 12, 2015. Due to platform restrictions, it

uses the Web Kit layout engine instead of Gecko, as with all other iOS web browsers. Firefox was created in 2002 under the codename "Phoenix" by the Mozilla community members who desired a standalone browser, rather than the Mozilla Application Suite bundle.



2.2.3 UC Browser

It is a web browser developed by the Chinese mobile Internet Company UC Web, which is in turn owned by the Alibaba Group. As of December 2018 it is the fourth most popular mobile browser in the world by market share, after Google Chrome, Safari, and Firefox. Originally launched in April 2004 as a J2ME-only application, it is available on a number of platforms including Android, BlackBerry OS and IOS, Java ME, Symbian, Windows Phone and Microsoft Windows.



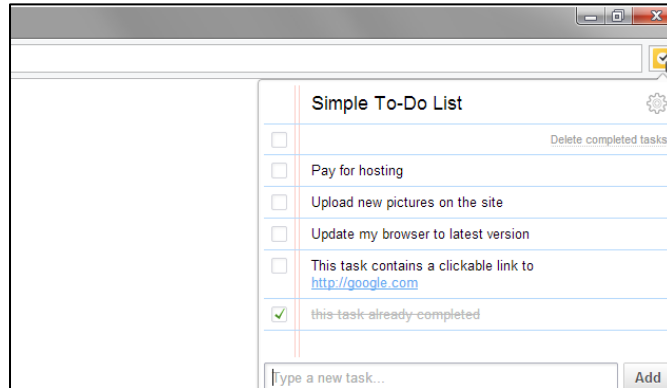
2.3 Extensions in the Market

Extensions are small software programs that customize the browsing experience. They enable users to tailor Chrome functionality and behavior to individual needs or preferences. They are built on web technologies such as HTML, JavaScript, and CSS. An extension must fulfill a single purpose that is narrowly defined and easy to understand. A single extension can include multiple components and a range of functionality, as long as everything contributes towards a common purpose. A screenshot of an extension's icon in the browser bar User interfaces should be minimal and have intent. They can range from a simple icon, such as the Google Mail Checker extension shown on the right, to overriding an entire page. Extension files are zipped into a single .crx package

that the user downloads and installs. This means extensions do not depend on content from the web, unlike ordinary web apps. Extensions are distributed through the Chrome Developer Dashboard and published to the Chrome Web Store. For more information, see the store developer documentation.

2.3.1 Name of some related extensions or add-ons

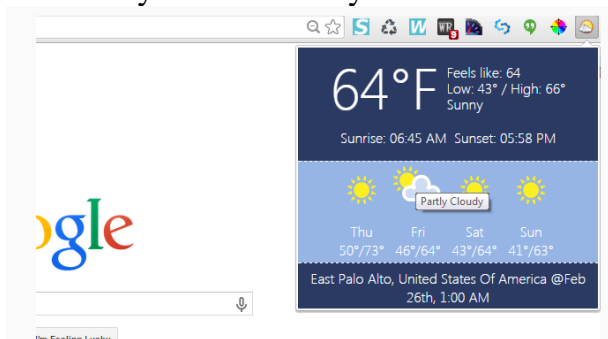
- **Simple To-Do List (chrome):** This simple extension allows you to easily manage your daily tasks right in your browser.



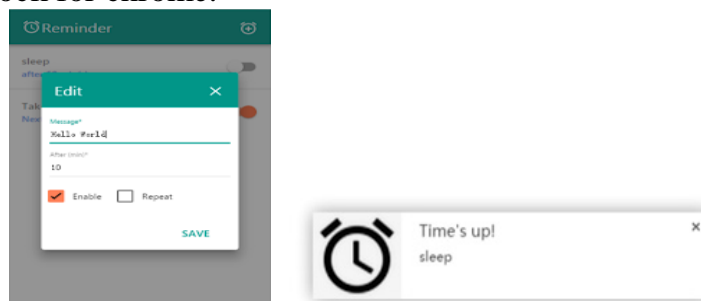
- **Web time Tracker:** keeps track of how you spend time on the web and presents the stats in a useful and intuitive way.



- **Quick weather:** Report based on your location by internet.



- **A simple alarm clock for chrome.**



2.3.2 A Productive Browser Extension

A browser extension which is an extension of maximum features and cross platform which makes it the best formatted extension in the store, which is not usually available on the Internet Productive Browser Extension have several advantages over traditional printed Extension. It provides as given:

- a) User friendly and cross platform.
- b) Less time consuming to run and set up.
- c) Latest updates of tasks with priority.
- d) Open access by cross platform form.
- e) Browsing result pie chart generator.
- f) Show time, location, weather etc.
- g) Login through internet with online database etc.
- h) Allows remote access.
- i) Can be used simultaneously by cross platform forming.
- j) Provides timely notifications and tasks list with priority.
- k) Supports real time database for no data loss.
- l) Accommodates unique features as: clock, location, weather showing.
- m) Show browsing results with Pie chart.
- n) Ease of use and user customization available in some cases.

To make the browsing time more efficient by this system and time management Extension with much more user friendly features and online database for no data loss and remote login, we developed this web browser based Productive Browser Extension.

2.4 Why Productive Browser Extension

Traditional browsing time and task management extension is a complex and time consuming, especially without the process automation, or if the manuscripts, reviews, and other information is circulated by **regular and real time checking**. The work overhead present in classical browsing time management extensions causes time delays and requires **additional information such as time ,locations, weathers (not mandatory)**. One of the most difficult requirements on this extension is remote login is to reduction of **data loss**. To reduce this lacking it is necessary to efficiently conduct the review process and this can only be done using a **productive browsing extension system** that speeds up the exchange of **information and notify all tasks with online real-time database** that we made.

Chapter 3

System Design and Development

3.1 Overview

System analysis is the process of studying an existing system to determine how it works and how it meets user needs. System analysis lays the groundwork for improvements to the system. The analysis involves an investigation, which in turn usually involves establishing a relationship with the client for whom the analysis is done and with the user of the system. This analysis phase is more of a thinking process. In this phase, we have improved logical aspects of the system. To develop the system, we have to consider about a key question “What must be done to solve the problem? In this phase we studied the system processes, gathering Operational data, understand the information flow, finding out weaknesses and evolving solutions for overcoming the weaknesses of the system so as to achieve the goals.

3.2 Approaches

It is necessary to determine whether the system to be built is suitable for incremental or not. This is decided depending on application area, complexity and projects characteristics. We choose it for the following reasons:

1. Requirements of the complete system are clearly defined and understood.
2. Incremental model improves the quality of software day by day.
3. Major requirements must be defined; however, some details can evolve with time.
4. There is a need to get a product to the market early.
5. A new technology is being used.
6. Resources with needed skill set are not available.
7. There are some high risk features and goals.

3.3 Design

There are the four phases of incremental model:

- Requirement Analysis: Requirement and specification of the software are collected
- Design: Some high-end function are designed during this stage
- Code: Coding of software is done during this stage
- Test: Once the system is deployed, it goes through the testing phase

3.4 Incremental model

3.4.1 System Design

System design is most creative and challenging. The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input

formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

3.4.2 User Panel Design:

In user panel design we have done our task for user. Here we provide facility about “PRODUCTIVE BROWSER MANAGEMENT”. In index page user can select any options which is needed by him/her. By selecting options he/she can see the desired page.

3.5 Data Flow Diagram

A data flow diagram is a short road map for that graphically represents how the data moves through the existing system .we have used data flow diagram in design process. The data flow diagram provides facilitating communication between us and user. DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

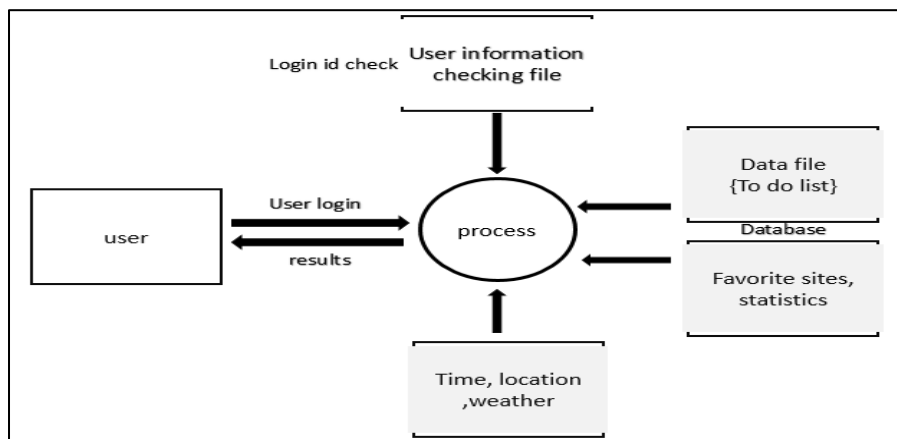


Figure 3.1: Data flow diagram

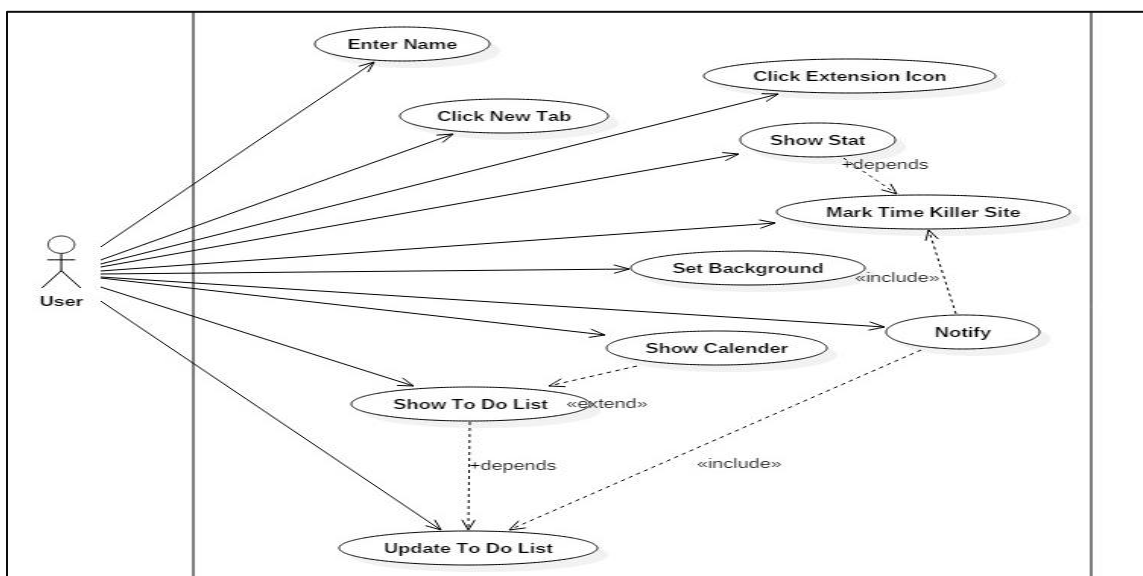


Figure 3.2: User case diagram

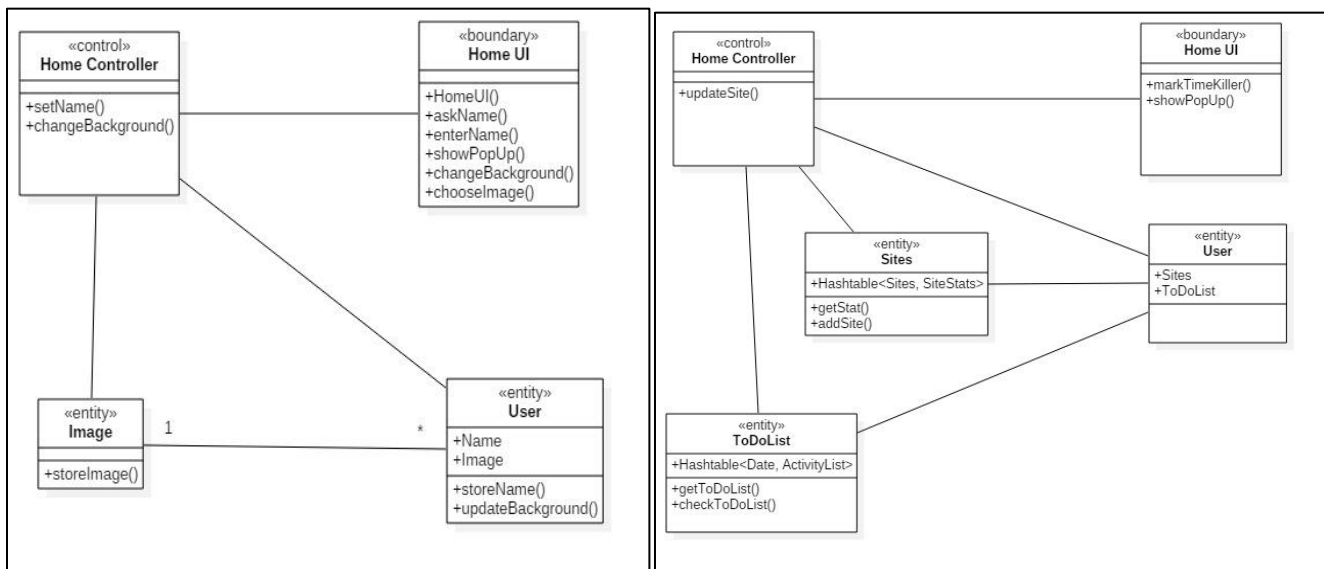
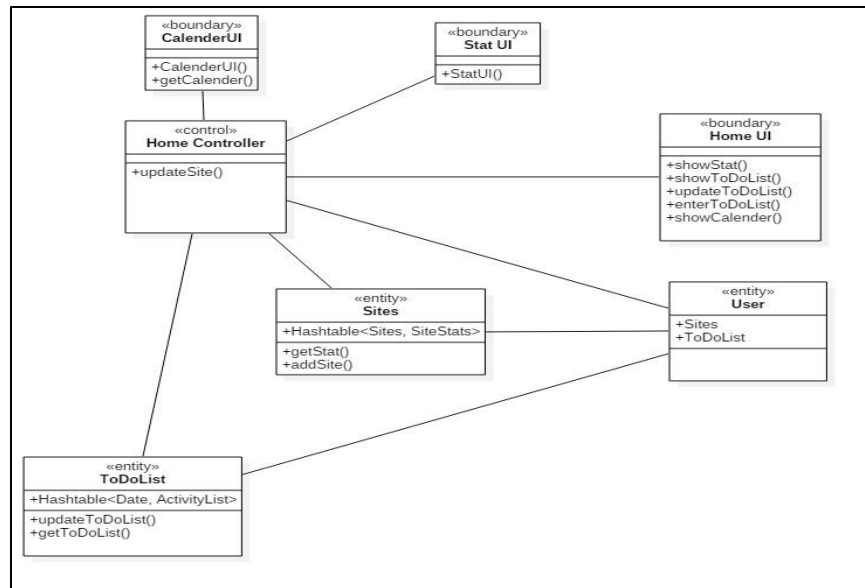


Figure 3.3: Class diagram

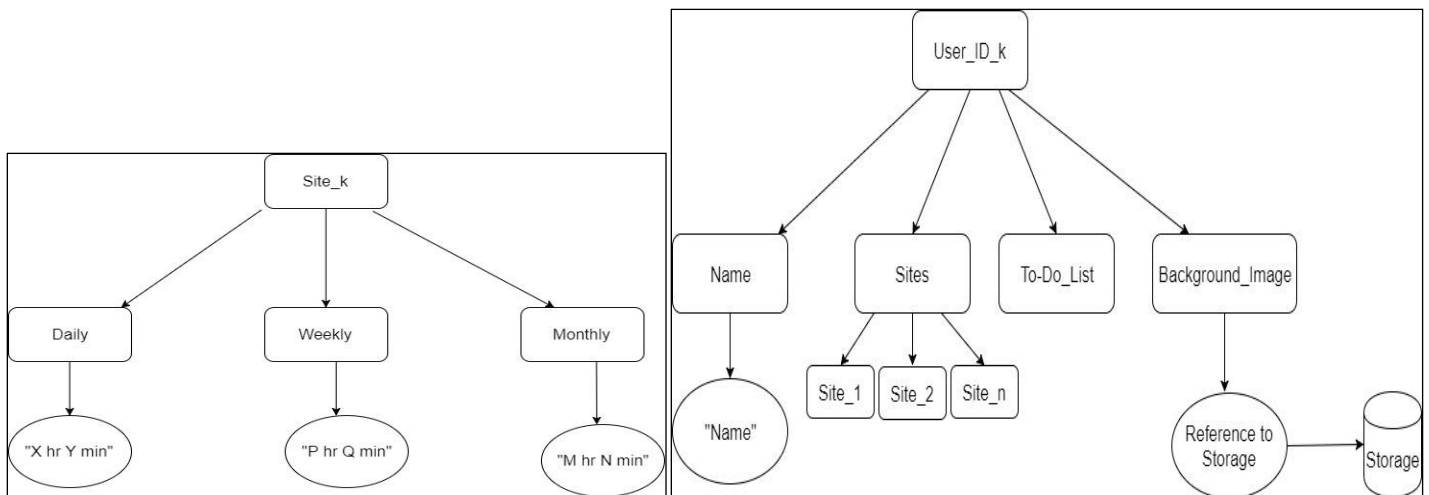





Figure 3.4: Database tree diagram

3.6 E-R (Entity Relationship) Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an ERD are:

- Entities 
- Relationships 
- Attributes 

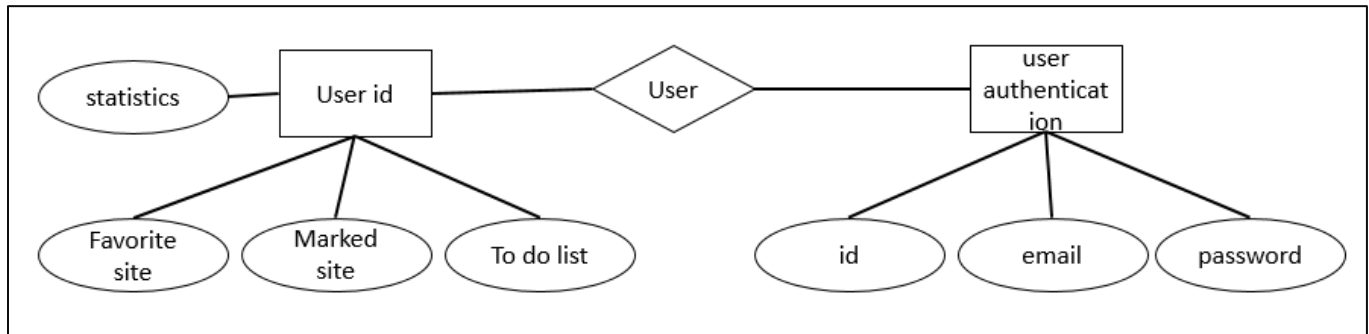


Figure 3.5: E-R Diagram

3.7 Software Process Model

A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective as:

- Specification.
- Design.
- Validation.
- Evolution.

General Software Process Models are

1. Waterfall model:
2. Prototype model.
3. Rapid application development model (RAD).
4. Evolutionary development:
5. Incremental model.
6. Iterative model.
7. Spiral model.
8. Component-based software engineering.

From the above we have chosen Incremental Model for its advantages over other Models.

3.8 Incremental Model

In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved. In the diagram below when we work incrementally we are adding piece by piece but expect that each piece is fully finished. In adding the pieces until it’s complete.

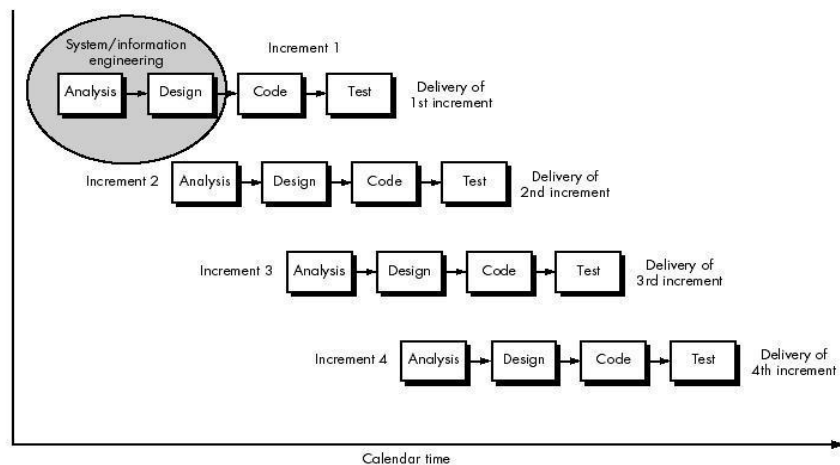


Figure 3.6: Incremental model

3.8.1 Advantages of Incremental model

- Generates working software quickly and early during the software life cycle.
- More flexible – less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

3.8.2 Disadvantages of Incremental model

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than waterfall.

3.8.3 When to use the Incremental model

- Requirements of the complete system are clearly defined and understood.

- Major requirements must be defined; however, some details can evolve with time.
- There is a need to get a product to the market early.
- A new technology is being used
- Resources with needed skill set are not available
- There are some high risk features and goals.

3.9 Programming Language

The project background model specially designed on the basis of certain web programming language like JSON (**JavaScript Object Notation**), JAVASCRIPT, CSS, and HTML etc. We use FIREBASE as a database. In following section here we are going to give a brief description about this language in this project.

3.9.1 HTML

- Stands for Hypertext Markup Language.
- Used for frontend design.
- Especially for style sheets and linking with other pages.

3.9.2 CSS

- Cascading Style Sheets (CSS)
- Simple mechanism
- Easy for adding style (e.g., fonts, colors, spacing) to Web documents.

3.9.3 JAVASCRIPT

- Affects the behavior of HTML processors like web browsers.
- **HTML** to define the content of web pages
- **CSS** to specify the layout of web pages
- **JavaScript** to program the behavior of web pages

3.9.4 JSON

- JSON: JavaScript Object Notation.
- JSON is syntax for storing and exchanging data.
- JSON is text, written with JavaScript object notation.

3.9.5 Firebase

Firebase is a backend platform for building Web, Android and IOS applications. It offers real time database, different APIs, multiple authentication types and hosting platform. This is an introductory tutorial, which covers the basics of the Firebase platform and explains how to deal with its various components and sub-components.

Chapter 4

Results and Discussions

4.1 Overview

To build up our project we use Software Testing Process for executing a program with the intent of finding errors that is uncovering errors in a program makes it a feasible task and also trying to find the errors (whose presence is assumed) in a program. As it is a destructive process.

4.2 Functionality description of Productive browsing extension

- **To do list:** To list tasks to be done with priority and alarm by using date through calendar.
- **Favorite link:** Listing favorite sites for quick access and history to save links.
- **Statistics:** Making chart of using marked site with time.
- **Marked sites:** Mark time killing sites to count wastage of time and make statistics chart.
- **Weather:** Show weather through website.
- **Time:** Show current time through desktop and internet.
- **Login and out:** Authentication of user with mail and password.
- **Background change:** Change background image from default or pc.
- **Color inversion:** White or black can be selected for text type items.
- **Remover background:** Remove current back ground.

4.3 Experimental Results

- Homepage showing time, location and weather.

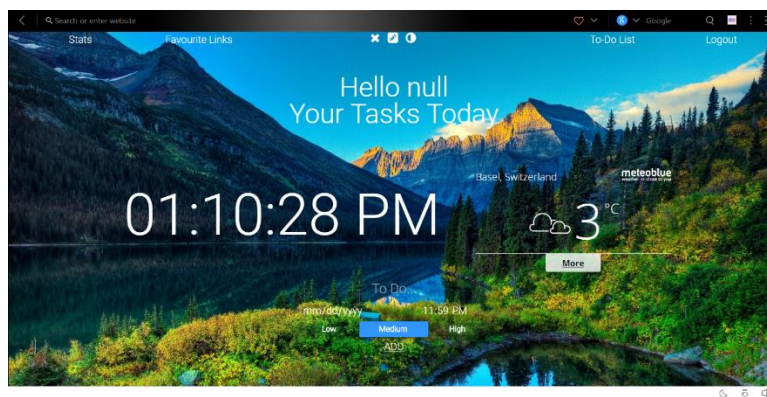


Figure 4.1: Home page of user specification

- Authentication of user form any device.

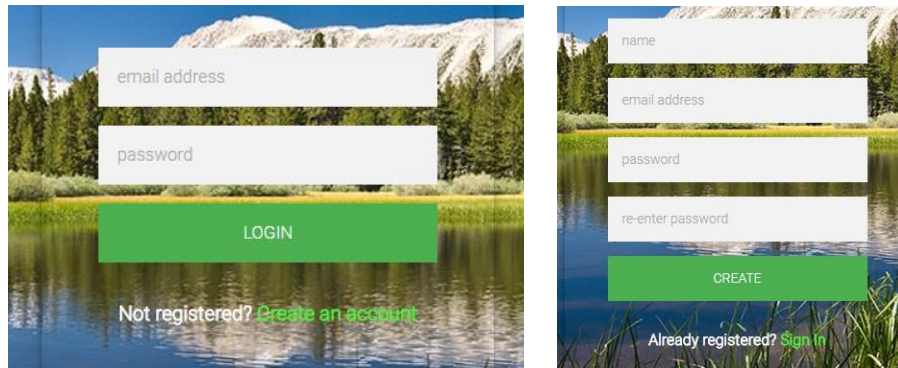


Figure 4.2: Log-in and registration information for user specification

- Working with to do list.

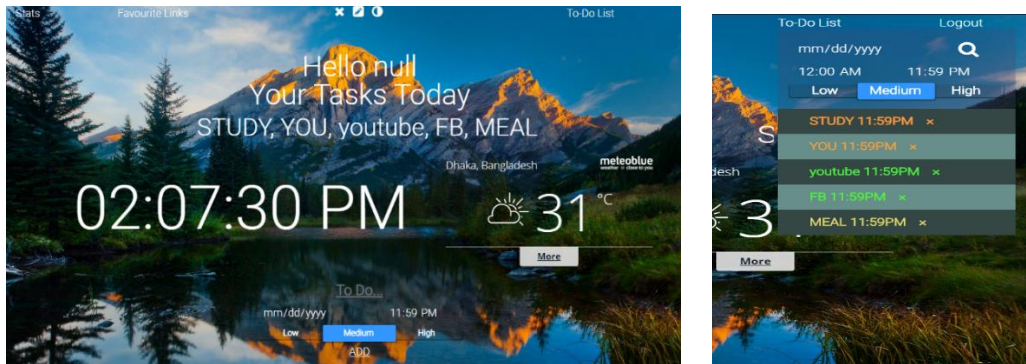


Figure 4.3: To do list

- Statics of browsing time of marked sites.

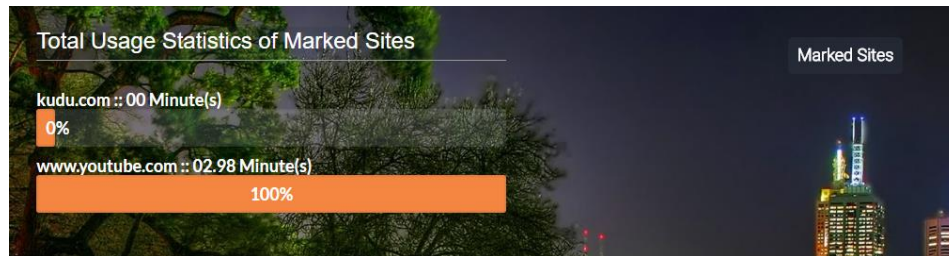


Figure 4.4: Browsing time statistics

- Favorite links shows marked favorite or quick links.

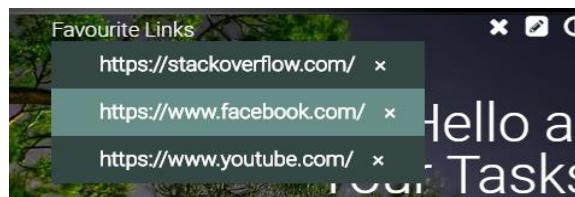


Figure 4.5: Favorite links

- Showing user name by authentication from database and some small feathers.



4.4 Achievements of this System

1. Managing To-Do List for focusing on working modules.
2. Keeping browsing statistics
3. (E.g. how much time spent in which website per day/week/month), for Self-realization of time spending on internet.
4. Notifying the user when he/she uses a time killing site for a certain time.
5. Ensuring the best use of time and inter to be very productive and recommending productive works.
6. Tasks showing their priority along with time.
7. Add favorite sites and get statistics of browsing states.
8. Time weather and location monitoring.
9. User can custom background options.

The most attractive point in this browser extension is it supports cross plat forming.

4.5 Limitation of this System

The system is a unique application, but it is not the most perfect one. We have tried our level best to make this application most preferable one. We will extend this application at large volume in future.

1. No mailing system.
2. Does not keep all your data neatly organized.
3. Add blocker some time cause abnormal behavior.
4. Sometime problem generated in real-time update in database
5. Cant access without internet.

4.6 Future Scope

We have many scopes to extend this project in large volume in future. Here we have included the future goals and scopes of our system:

1. The future scope of our project is valuable
2. This project can be improved by enhancing the level form time to time.
3. If this project is get a big sponsor, it will be an essential extension.
4. In future if we get chance we will develop this extension for more beneficial features.
5. A user login functions can be add through Facebook, twitter, phones which will also will be send to the users code for verification.

4.7 Conclusion

We develop web-based browser extension application “**PRODUCTIVE BROWSER EXTENSION**”. To establish this extension we use various methodology. To develop this extension we face many problem but we hardly tried to develop this project. Our supervisor helps us by giving his valuable opinion, decision and time. The future scope of our project is valuable. Our project time duration was only 6 months .In this time interval we developed our project. It was very difficult to complete project.

In future if we get chance we will develop this extension with larger scopes and possible features. The project report entitled “**PRODUCTIVE BROWSER EXTENSION**” has come to its final stage. The Browser Extension system has been developed with much care that it is free of errors and at the same time it is efficient and less time consuming. The important thing is that the system is robust. We have tried our level best to make the site as dynamic as possible. Also provision is provided for future developments in the system. The entire system is secured. This online system will be approved and implemented soon.

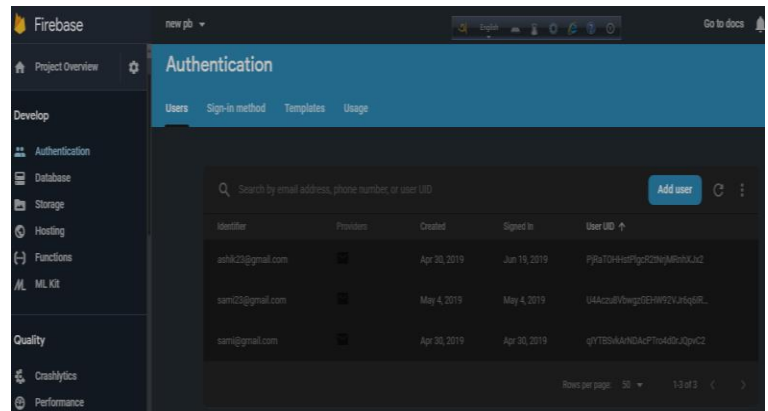
References

- [1] T. Ball and S. K. Rajamani. The SLAM project: Debugging system software via static analysis. In ACM POPL, 2002.
- [2] S. Bandhakavi, S. T. King, P. Madhusudan, and M. Winslett. VEX: Vetting browser extensions for security vulnerabilities. In USENIX Security, 2010.
- [3] A. Barth, A. P. Felt, and P. Saxena. Protecting browsers from extension vulnerabilities. In NDSS, 2010.
- [4] CA Technologies. Virus details: Win32/clspring family.
<http://www.ca.com/us/securityadvisor/virusinfo/virus.aspx?ID=42280>, 2006. Last Accessed on 12.03.19
- [5] J. Chen, R. Chugh, and N. Swamy. Type-preserving compilation of end-to-end verification of security enforcement. In ACM PLDI, 2010.
- [6] R. Chugh, J. A. Meister, R. Jhala, and S. Lerner. Staged information flow for JavaScript. In ACM PLDI, 2009.
- [7] L. de Moura and N. Bjorner. Z3: An efficient SMT solver. In TACAS, 2008.
- [8] M. Dhawan and V. Ganapathy. Analyzing information flow in JavaScript-based browser extensions. In ACSAC, 2009.
- [9] M. Egele, C. Kruegel, E. Kirda, H. Yin, and D. Song. Dynamic spyware analysis. In USENIX Security, 2007.
- [10] A. P. Felt, K. Greenwood, and D. Wagner. The effectiveness of install-time permission systems for third-party applications. Technical report, University of California at Berkeley, 2010. UCB/EECS-2010-143.
- [11] M. Fredrikson and B. Livshits. RePriv: Re-envisioning inbrowser privacy. In IEEE S&P, 2011.
- [12] I. Goldberg, D. Wagner, R. Thomas, and E. A. Brewer. A secure environment for untrusted helper applications. In USENIX Security, 1996.
- [13] Google Chrome. Popular Google Chrome extensions.
<http://chrome.google.com/extensions/list/popular>, 2010. Last Accessed on 26.03.19
- [14] A. D. Gordon and A. Jeffrey. Authenticity by typing for security protocols. Journal of Computer Security, 11, July 2003.
- [15] C. Grier, S. Tang, and S. T. King. Secure web browsing with the OP web browser. In IEEE S&P, 2008.
- [16] O. Grumberg and D. E. Long. Model checking and modular verification. ACM TOPLAS, 16, 1994.
- [17] S. Guarnieri and B. Livshits. GateKeeper: Mostly static enforcement of security and reliability policies for JavaScript code. In USENIX Security, 2009.
- [18] S. Guarnieri and B. Livshits. Gulfstream: Incremental static analysis for streaming JavaScript applications. In USENIX Conference on Web Application Development, 2010.
- [19] A. Guha, C. Saftoiu, and S. Krishnamurthi. The essence of JavaScript. In ECOOP, 2010.

Appendix A

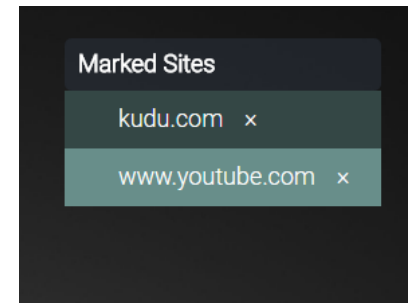
Link to firebase as online database

```
var config = {
  apiKey: "AIzaSyCZdjUt576S-
zQ49v3ImVuGCkgliHm7zzI",
  authDomain: "new-pb.firebaseio.com",
  databaseURL: "https://new-pb.firebaseio.com",
  projectId: "new-pb",
  storageBucket: "new-pb.appspot.com",
  messagingSenderId: "345542359013"
};
firebase.initializeApp(config);
```



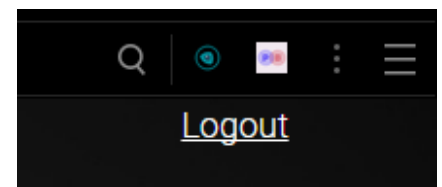
Popular marked sites in stat.js

```
function marked_site_onclick_handler(ev) {
  if (ev.target.tagName === 'LI') {
    var div = ev.target;
    var site = div.textContent.substring(0, div.textContent.length - 1);
    var index = sites_for_bar_chart.indexOf(site);
    if(index !== -1)
    {
      document.getElementById("site_usage").style.display = "block";
      document.getElementById("site_name").innerText = site;
      document.getElementById("daily_usage").innerText = min_to_hour(daily_time[index]);
      document.getElementById("weekly_usage").innerText = min_to_hour(weekly_time[index]);
      document.getElementById("monthly_usage").innerText = min_to_hour(monthly_time[index]);
    }
  }
}
```



Log out pop up

```
function log_out() {
  firebase.auth().signOut()
    .then(function () {
      chrome.storage.sync.remove(["uid", "name", "marked_sites"]);
    })
    .catch(function (error) {
      alert(error.message);
    });
}
```



```

});
toolbar.style.display="none";
log_in_div.style.display = "block";
mark_div.style.display= "none";
unmark_div.style.display ="none";
return false;
}

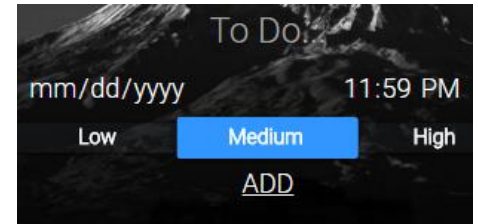
```

To do from extension.js

```

function search_to_do() {
    var date = document.getElementById("Search_date").value;
    var maxTime = document.getElementById("maxTime").value;
    var minTime = document.getElementById("minTime").value;
    search_to_do_from_server(uid, date, minTime, maxTime);
    //get_to_do_from_server(uid,date);
    return false;
}

```



Add favorite links to server from server.js

```

function add_fav_link_in_server(uid, link) {
    var userRef = ref.orderByChild("UID").equalTo(uid);
    userRef.once("child_added").then(function (dataSnapshot) {
        var tmp = {
            site : link
        };
        ref.child(dataSnapshot.key + '/Favourite Sites').push(tmp)
        .then(function () {
            console.log("Site: " + link + " favoured for uid: " + uid);
        }).catch(function (error) {
            console.log(error.code);
            console.log(error.message);
        });
    });
}

```

