**KABARAK UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND IT**

**DEPARTMENT OF COMPUTER SCIENCE AND IT**

PROJECT PROPOSAL

PROJECT PROPOSAL TITLE: UPTIME MONITORING WEBAPP

NAME: DENNIS KESTONE

ADMISSION NUMBER: BMIT/MG/1415/09/17

**A Project Report Documentation Submitted in The Department of Computer Science and IT in partial fulfillment of the degree of Business Management and Information Technology**.

**DECLARATION**

I would like to acknowledge that this is my work and that any outsourced data has been cited appropriately.

**ACKNOWLEDGEMENT**

I wish to acknowledge the help and contribution of the following for their contribution in making this research report a huge success.

The Almighty God, for giving me the zeal for knowledge, hard work, perseverance and dedication through this journey.

My sincere thanks my lectures for their tireless efforts, guidance and training in the conceptualizing, developing and write up of this research paper.

My loving mother, Betty Cherotich for her insight and immense support throughout this entire university period.

I would like to extend my gratitude to Vera, for her reviews and constructive critism in the project enabling me to make it better.

May God bless you all.

**CONTENTS**

DECLARATION I

ACKNOWLEDGEMENT II

TABLE OF CONTENT III

ABSTRACT IV

CHAPTER ONE: INTRODUCTION

1.1. BACKGROUND OF THE STUDY 1

1.2. PROBLEM STATEMENT 2

1.3. OBJECTIVES

1.3.1 main objective…………………………………………………………………….3

1.3.2 specific objectives……………………………………………………………….3

1.4. SIGNIFICANCE OF THE STUDY 4

1.5. SCOPE 5

CHAPTER TWO: LITERATURE REVIEW

2.1. INTRODUCTION……………………………………………………………………….8

2.2. THE RISKS OF WEBSITE DOWNTIME 9

2.3. THE EFFECTS OF DOWNTIME ON CUSTOMERS 10

2.4. IMPORTANCE OF QUANTIFYING RISK AND THE IMPACT OF DOWNTIME………...……………………………………………………………11

2.5. METRICS TO BE MONITORED FOR UPTIME ………………………………...…11

2.6. TRENDS EMERGING FROM LITERATURE REVIEW 12

2.7. THE BENEFITS OF UPTIME MONITORING ……………...……………………...12

CHAPTER THREE: [RESEARCH METHODOLOGY 11](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995079)

[3.1 INTRODUCTION 11](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995080)

[3.2 DATA COLLECTION METHODS 11](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995081)

[3.2.1 PRIMARY COLLECTION METHODS 11](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995082)

[3.2.1.1 INTERVIEWS 11](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995083)

[3.2.2.2 QUESTIONAIRES 12](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995084)

[3.2.2 SECONDARY COLLECTION METHODS 12](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995086)

[3.3 VALIDITY 13](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995087)

[3.4 RELIABILITY 13](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995088)

[3.5 ETHICAL CONSIDERATIONS 14](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995089)

[3.6 TESTING 14](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995091)

[3.7 IMPLEMANTATION 14](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995092)

[3.7.1 Software Design Life Cycle. 14](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995093)

[3.7.2 Phases 15](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995094)

[CONTEXT DIAGRAM 16](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995095)

[DATA FLOW DIAGRAMS 16](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995096)

[7](file:///C:\Users\KESTONE\Desktop\notes\4.1%20notes\reseach%20proposal\PROPOSAL1.docx#_Toc55995098)

**CHAPTER FOUR: SYSTEM DESIGN AND IMPLEMENTATION**

4.1 Introduction

4.2 system architecture

4.3 Front end development (you can put some code extract )

4.3.1 User interface Design

4.3.2 User Interface modules ( screenshot + explanation)

4.4 Back-end development

4.4.1 Database Design models

4.4.2 Tables (if any) or data models CSV file Artff etc.

4.4.3 Code testing

4.5 Deployment methods

4.6 Conclusion and Future work

**ABSTRACT**

We live in an interconnected world powered by informational technologies. In this world, **a strong always-on presence on the web is an absolute necessity for success**. The Web is, usually, the place where most companies face the majority of their clients and where the bulk of their revenue is created.

As a SaaS or other cloud-based production team, you may often struggle with creating the right infrastructure to optimize your production cycle and improve service levels. Because cloud services are used around the clock, production environments need to run 24/7, with service providers expected to be responsive at all times. This condition requires teams to work smart, not hard when it comes to the production environment by automating many of its processes. Uptime monitoring is one of those very important measurements that help you do a better job at monitoring the quality of your services, but what is uptime monitoring exactly?

Uptime monitoring is a method that helps you get a bird’s eye view of your online services, and involves monitoring a collection of metrics and measurements relevant to a specific business. If, for example, you provide cloud-based services on the web, like many SaaS companies do, then your customers must have access at any given time. In this situation, what could be the worst outcome? That your online services go down. In the world of uptime monitoring, there are only two states—uptime and downtime. Uptime is when your online services are up and running, and downtime means they are unavailable.

Since customers experiencing downtime will be detrimental for your customer success rate, you must take the necessary precautions to decrease potential downtime. From the number of times that services are temporarily unavailable to the amount of time it takes for your team to get them up and running again, uptime monitoring makes sure that downtime scenarios can both be minimized and mitigated. The less often customers encounter downtime, the more they will value your service levels and continue to trust that your platform meets their needs.

Without a website that runs smoothly and is readily accessible to customers, any of your web-related investments could go to waste and the entire company could be compromised.

Uptime monitoring helps to minimize costly and detrimental downtime by provision of timely alerts in case of any failures.

**CHAPTER ONE**

**INTRODUCTION**

* 1. **Background of the study**

It is an inescapable fact that in today’s world, web-related processes have become so complex that no one can work in a vacuum and one must work with the help of real-time monitoring of their websites to get things done.

Theoretically, the rapid emergence of social networking systems within organizations has been attributed to the significance of 'knowledge capital'. This is anchored on the idea that, despite the fact that knowledge can be stored in various forms ranging from print to electronic, the cognitive knowledge still resides in individuals. In this sense, effective interaction between individuals is of great importance in enhancing sharing of the cognitive knowledge.

This though has a downside because it can be harmful to the organization in many ways that the study aims to explore and create measures that may assist in combating these pitfalls to ensure that only the good portions of social media usage are harnessed.

* 1. **Problem statement**

Website crashes and downtime is costly to the owner and leads to considerable loss of traffic and eventually potential profit. This may be due to:

* **Traffic spikes**: Probably the number one cause of websites going down are sudden spikes in traffic. These can be caused by a surge of viral traffic from sites like Reddit, BuzzFeed, or Quora. A reliable hosting provider who has the necessary means to guarantee uptime even in such situations is really the only way how you can keep your site up and running.
* **Scheduled maintenance**: Even websites with excellent optimization require scheduled maintenance, from time to time. Maintenance should happen during off-peak hours, to minimize its potential negative impact.
* **Hacker attacks**: Hackers can bring a website down in multiple different ways: they can overload the servers with DDoS attacks, inject malicious code, or steal security credentials, thus forcing the website owners to shut off the site and solve the problem.
* **Data center problems**: Some periods of downtime can be caused by the [web](https://digital.com/web-hosting/) hosting company itself. In this case, website uptime monitoring tools come in particularly handy because they allow you to accurately judge how reliable your [web hosting company](https://digital.com/web-hosting/inmotion-hosting/) really is.
* **Website’s code issues**: Poor programming can cause an unnecessarily heavy load on servers, result in various database errors, and, ultimately, make the site totally unresponsive and in the long run detrimental to your profits.

other factors that could bring a website down, such as domain name (DNS) expiration, domain name seizure, takedown by law enforcement, but these tend to be much less common.SSS

* 1. **Objectives**
     1. **Main objective**

The main objective of this study is to determine the effects of downtime and outages of availability on website productivity and its eventual accrued revenue loss and how can organizations use these behaviors to help to drive successful growth strategies and avoid losses.

* + 1. **Specific objectives**

Determining what downtime of website can do the SEO ranking of the website and creation of future preventive measure.

To know the potential risks and dangers posed by unplanned and unscheduled downtime the potential measures that can curb this menace.

To determine the amount of possible workload and bad code that can lead to lagging of website and eventual downtime and how to improve in hosting of fast code that has been written well.

* 1. **Significance of the study**

A growing challenge for applications is obtaining optimal availability at all times. Today, cloud-based infrastructures are often built with a large number of systems geared for elastic scalability while hardware costs should be kept to a minimum. These flexible scenarios mean that certain components are geared to fail.

The study aims to design an application to inform you of occasional downtimes, so that you can apply corrective measures fast.

Delivering availability for each application, at the right time, requires a considerable understanding of usage patterns. By nature, each application is designed to sustain certain capacities. Designating fixed availability is usually not a viable option as certain factors, like patterns of usage, are not being considered.

* 1. **Scope**

Uptime Management is a set of services and tools designed for controlling, monitoring and optimising operational productivity. Proper uptime management is indeed crucial in averting emerging issues, solving critical situations and reducing downtime. Furthermore, Uptime Management encompasses a disaster recovery mechanism in the event of an emerging issue.

Here are the 7 main services that Uptime Management should encompass:

* 24/7 NOC Centre
* Real-Time Monitoring platform
* Tier 1+2 services
* DevOps
* Run-book operation and centralized dashboard
* Infrastructure Maintenance
* DR Management

**CHAPTER TWO:**

**LITERATURE REVIEW**

**2.1. Introduction**

Most major brands including Google, eBay, Microsoft, Twitter and Amazon have suffered downtime and it has led to considerable losses in their revenue. In 2013 **Amazon** went down and they apparently lost $66,240 for each minute they were not available. That is $1,104 per second!!

According to a survey conducted in 2007, Amazon made around $29,000 every minute. That means that their two hours of downtime back in 2008 led to a loss of $3.48 million.

This goes on to show that a small business website or online shop suffering outage will impact the bottom-line revenue.

Another aspect which is harder to quantify is the **negative impact on reputation and credibility,** like loss of customer loyalty, losing business to your competitors, and the impact on your business's reputation. That is much harder to measure, but that doesn’t make it any less important.

Uptime is the amount of time the web host server (and your website) is available and operational. **Uptime is an important metric** to factor in when choosing a web host and this metric is often stated as a percentage.

For example, a **99.9% uptime equates to a yearly downtime of 8.76 hours.**

**2.2. The Risks of website downtime**

If your website is a critical component of your business, depending on how essential it is to your business’ function i.e., an ecommerce website or an eLearning site, you should be thinking about the criticality of uptime and methods to ensure this similar to how you ensure against other forms of loss.

* opportunity and Productivity Cost

Perhaps the most serious potential effect of site downtime is the cost of lost opportunity. For an eCommerce site, that could mean lost sales. For an educational institution or eLearning sites, it could mean lost registrations or missing out on new leads. Lost productivity is also a major problem – when a website goes down, countless hours can be spent resolving the problem or managing secondary issues. Basically, what work can employees, visitors, customers not**accomplish** while your site is down?

* Damage to Brand Perception

Of equal concern, but more subjective, is the potential damage to your reputation. You only ever get one chance to make a first impression. Imagine if you were a customer and you visit a website only to be welcomed with an error 404 Not Found message. If your site is down when a new prospect arrives that sets an immediate red flag to them. Even if it’s a simple transient issue that takes your site down for a few minutes, from that visitor’s perspective, you’re down.

* Damage To SEO

Depending upon the length and frequency of your site’s downtime, it’s possible that your ranking in the SERPs could be affected. A single event of short duration will likely have minimal long-term effects (although your ranking could be displaced for several hours or more). Extended or repeated downtime will probably give Google reason enough to replace your site with one that is more reliable and working effectively.

**2.3. The effects of downtime on customers**

Nobody likes a website with erratic availability, sluggish performance, or that simply doesn’t work correctly. In the case of downtime, some of your loyal customers will wait, but frequently they simply move on to your competitor. If that competitor offers a better, more reliable experience, you will lose them.

A study carried out by Akamai found that 9% of visitors that encounter a down site will not return to the site again. You may think 9% isn’t so bad, but those outages typically happen during your peak hours. Sure, planned maintenance also creates downtime, but planned outages take place during the slow times when your site has fewer visitors. So, 9% during peak hours may turn out to represent thousands of customers, and if you’re one of the big corporations, we are talking about millions of visitors permanently abandoning the site. During your peak times, what will an outage cost you? Food for thought.

**2.4. Importance of quantifying risk and the impact of downtime**

Successful risk mitigation requires that you first understand the risk. Only then can you can take the appropriate steps to reduce or eliminate the potential effects.

In regards to website downtime, you are faced with the option of risk acceptance or risk limitation. Neither risk avoidance nor transference is possible.

By understanding the costs associated with downtime, you are in a better position to make educated decisions. Meaning, you could choose to reduce the risk of downtime or accept it. But which one is the right choice?

Risk Acceptance

If your website is not responsible for generating leads and revenue or maintaining brand perception, then acceptance might be an appropriate strategy. There is no sense insuring against a risk that does not exist.

Risk Limitation

If your website plays a key role in your business, risk limitation is usually the appropriate strategy. Successful risk limitation involves making sure the cost of limitation does not exceed the actual risk itself. For example, maintaining an insurance policy with annual premiums that exceed the potential cost of downtime would not make financial sense.

**2.5. Metrics to be monitored for uptime**

There are many things which can be monitored to get complete data of your server, websites, and apps but not all the things are important. So, some top metrics which should be monitored on your servers are:

1. **Physical memory:**Memory is required for everything and if by chance your storage gets filled or is working on heavy usage for a very long time, it can fail which can cause unnecessary downtime.
2. **CPU load:**CPU processes must not exceed 80% of your total CPU power for best performance of your servers. However, in some situations load can increase due to any reason so by monitoring it you can know how to optimize it properly.
3. **Other processes and services:**There are many processes which are necessary and run on the background but sometimes any malware or any unexpected thing can cause an unnatural number of processes and services which can ultimately shut down your server. By monitoring them, you can know what process to keep and what to kill.
4. **Server cores:** This will help to know if the server is powerful enough to handle the requests coming and out of the website.

**2.6. Trends Emerging from Literature Review**

A number of key concepts that were discovered through the Literature Review and speak to the risks related to downtime and uptime concepts of websites. These concepts include:

* The benefits of high uptime of websites.
* The risks that exist in downtime of websites.
* The different approaches that can be taken by management to ensure a 99% uptime is achieved.

**2.7. The Benefits of uptime monitoring**

You only receive notifications when anything happens. No false alerts and no wastage of resources. Alerts of any hardware failure such as hard disk or CPU failures will also be sent to you. Allocate resources to your websites or apps properly by using the monitored data that is reliable and available to you in real time.

**CHAPTER THREE:**

**RESEARCH METHODOLOGY**

# 3.1 INTRODUCTION

Research methodology represents the framework that the researcher used to illustrate the procedures for collecting data for this study. The methodology for this study involves research design, examining the types of data required and their sources, methods of data collection, data analysis and presentation techniques.

# 3.2 DATA COLLECTION METHODS

The data collection methods targeted various companies that offer webhosting services and website creation within Nakuru County. I have administered interviews, questionnaires and in order to get their responses and feedback on the existing issues and problems in website downtime and their effect on the companies.

# 3.2.1 PRIMARY COLLECTION METHODS

# 3.2.1.1 INTERVIEWS

In an interview, the investigator and the respondent are set together in a face-to-face conversation between the two for a specific purpose. The interviews targeted the management of the companies and developers targeted.

Advantages of an interview are;

Less costly – Very simple, prompt and low cost.

Time-saving – Within a very short time, the conversation can be done.

Flexible – An interview is feasible therefore information obtained is abundant.

Sufficient – Question and Answer captures adequate information.

Development of Relationship—Increases mutual and beneficial understanding.

# 3.2.2.2 QUESTIONAIRES

An instrument of data collection that involves asking a given subject to respond to a set of written questions. The questionnaires that were administered targeted the developers, the clients who will in turn give credible information which was used in the analysis of data to determine how they are affected.

Advantages of questionnaires are;

* Covers every aspect of the topic.
* Answers are easy to analyze and get the right information.
* A large number of people can be easily reached.
* Less costly.
* Maintains the privacy of the respondent.

# 3.2.2 SECONDARY COLLECTION METHODS

A researcher can obtain secondary data from various sources. Secondary data may be published data or unpublished data.

Published data are available in: Publications, Journals, Reports, Public Records, etc.

Unpublished data may be found in: Letters, Diaries, etc.

Advantages

* The primary advantage of secondary data is that it is cheaper and faster to access.
* Secondly, it provides a way to access the work of the best scholars all over the world.
* Thirdly, secondary data gives a frame of mind to the researcher that in which direction he/she should go for the specific research.
* Fourthly secondary data save time, efforts and money and add to the value of the research study.

Disadvantages

* The data collected by the third party may not be a reliable party so the reliability and accuracy of data go down.
* Data collected in one location may not be suitable for the other one due variable environmental factor.
* With the passage of time the data becomes obsolete and very old
* Secondary data collected can distort the results of the research. For using secondary data, a special care is required to amend or modify for use.
* Secondary data can also raise issues of authenticity and copyright.

This study entailed identifying relevant documents in budgeting in various financial departments. This was mainly done using online review of the documents.

# 3.3 VALIDITY

Validity is the degree to which an instrument measures what it purports measure. The research concentrated on content validity by performing a pre-test so as to adjust the research tool to meet the required standards. The results of the study were validated by reviewing it with other similar researches done.

# 3.4 RELIABILITY

This is a measure of the degree to which a research instrument would yield the same results after repeated trials. The reliability of the questionnaire was be established through split half techniques where the pretest data set was split into 2 equal data sets and the Cronbach Alpha evaluated. The results indicated a Cronbach Alpha score of 0.75 greater than 0.7 which shows that the study results were reliable.

# 3.5 ETHICAL CONSIDERATIONS

Prior to data collection exercise, the respondents were informed of what the research was about, its research goals and objectives. Additionally, all respondents were guaranteed confidentiality during this process. Some of the problems encountered may have involved companies and clients that would not like their clients and end-users to know that their data may have been vulnerable at one time.

# 3.6 TESTING

In this stage the system was tested to check on its functionality. The test is also to verify that if works well and determine of other plugins that could be added to it to make it more useful.

# 3.7 IMPLEMANTATION

# 3.8.1 Software Design Life Cycle.

This is a process that describes the steps involved in the creation of software. It covers a detailed plan for building, deploying and maintaining the software. These steps were modified into phases where each phase contained a specific action. The phases include; Requirements gathering, Analysis, and Design, Implementation, Testing, Deployment, and Maintenance.

The Waterfall Methodwas used in the implementation the system.

Some of the advantages of waterfall model include:

* It is a simple method which is easily understood.
* Phases are done step by step.
* It is easily manageable; not complex.

# **3.8.2 Phases**

1. **Requirement Gathering**

At this phase, all information collected from the clients was drafted and used to come up with the final product. It contained the basis and foundation of the system that will be constructed. This methodology involved the collection of all system requirements at this stage. Here the SRS- Software Requirement Specification document is created. The output of the requirement phase was be the input of the analysis and design phase.

1. **Analysis and Design**

The requirements of the system were analyzed until the functionality of the system is understood. This phase was accompanied by documentation for each requirement. If the requirements are not well understood, it will allow jumping back to the requirement phase. Documents which are inputs of the next phase will be derived from this phase.

1. **Implementation**

Once the analysis and documentation are done, coding and development commences in this phase and all the software requirements was implemented in this phase. The software developed will be the input of the next phase.

1. **Testing**

This phase begins once the coding is complete and each module has been released. The modules will be run through a number of tests to identify any defects that may make the system fail and these will be corrected.

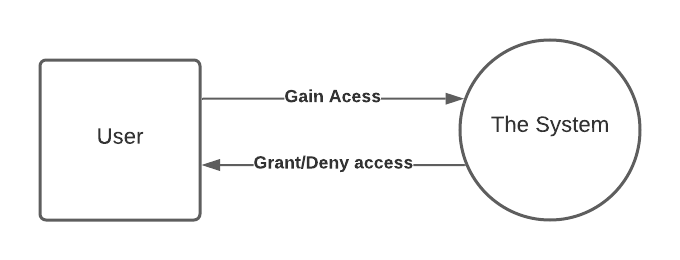
1. **Deployment**

Once the system has been tested and is free from errors, it will be deployed.

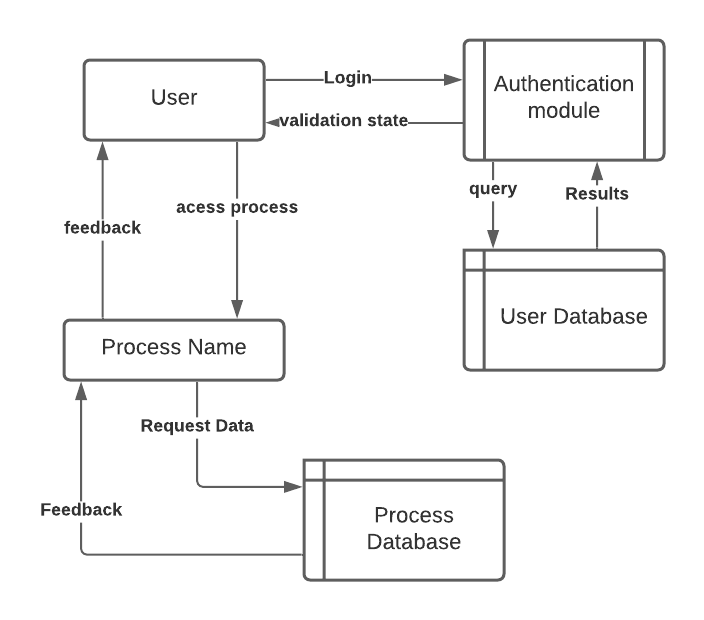
1. **Maintenance**

After deployment of the system, maintenance will be necessary because the system could later develop issues that need immediate attention or the system may need improvement in functionality.

# CONTEXT DIAGRAM

****

# DATA FLOW DIAGRAMS



# APPENDICES

APPENDIX 1: SCHEDULE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ACTIVITY | 1ST MONTH | 2ND MONTH | 3RD MONTH | 4TH MONTH |
| Software requirement |  |  |  |  |
| Logical design |  |  |  |  |
| Physical design |  |  |  |  |
| Documentation |  |  |  |  |
| Presentation |  |  |  |  |

The above schedule represents the timeline which will take me to complete my project. The 1st month represents my first month of my semester and so on.

APPENDIX 2: BUDGET

|  |  |  |  |
| --- | --- | --- | --- |
| ITEM | QUANTITY | UNIT PRICE IN KSHS | COST IN KSHS |
| LAPTOP/DESKTOP | 1 | 50,000 | 40,000 |
| DISK STORAGE | 1 | 5,000 | 5,000 |
| OTHER COSTS | - | 10,000 | 10,000 |
| TOTAL |  |  | 55,000 |

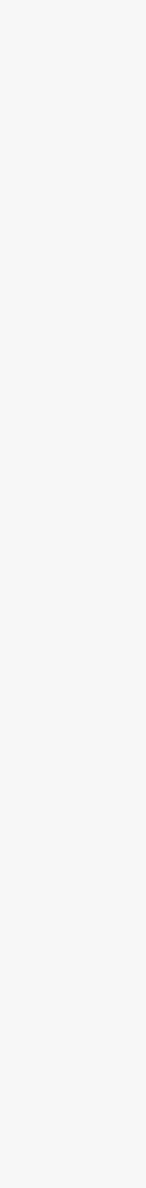
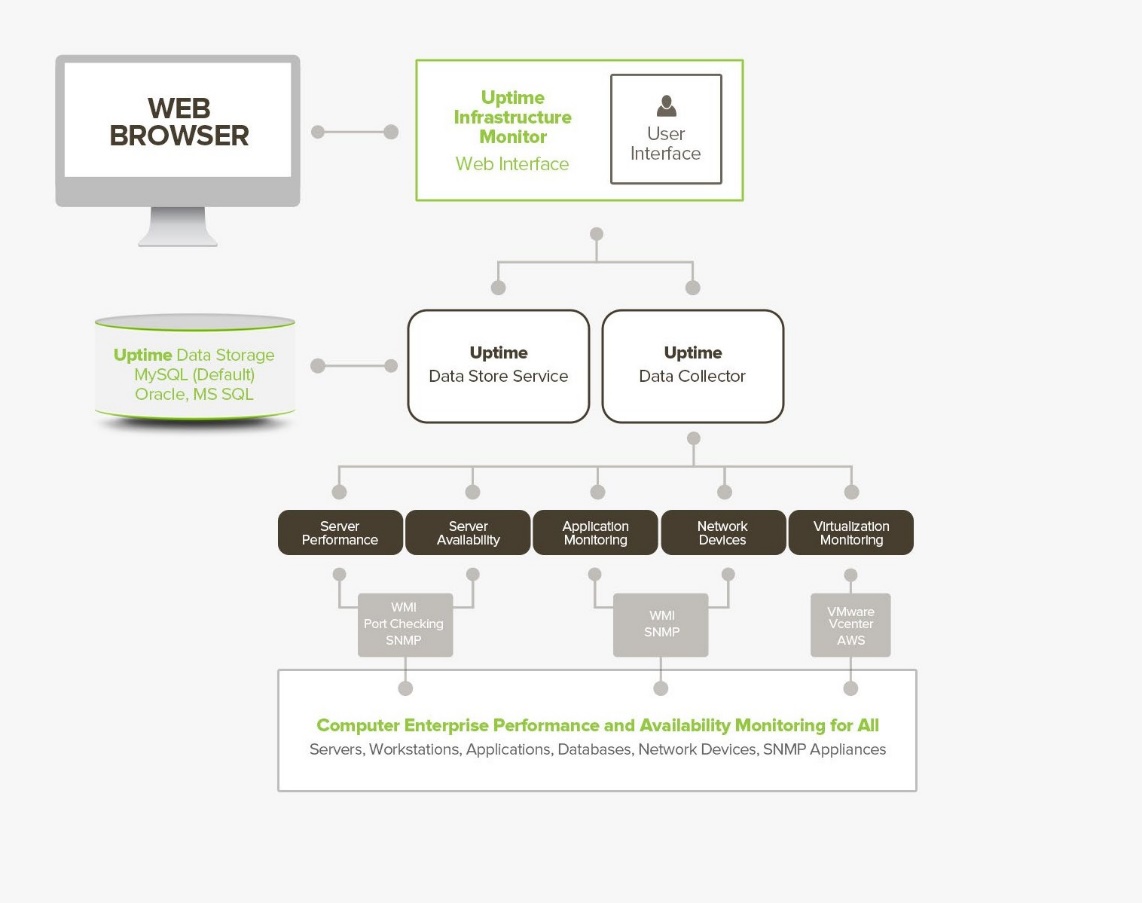
**CHAPTER FOUR: SYSTEM DESIGN AND IMPLEMENTATION**

# 4.1 INTRODUCTION

Upwork monitoring system is complex website monitoring system that aims to accomplish its task of monitoring websites that are experiencing downtime to their respective owners for prompt action. Central to the development of the monitoring system is the fact that it is there to serve the requirements of the overall project and as such, effort to adhere to scheduled meetings and assemble all the required materials and human resource for development and production.

4.2 System Architecture

Uptime monitoring webapp has been designed to be scalable; capable of monitoring thousands of elements with a very simple architecture. Most monitoring does not require an agent to be installed on the monitored system. However, an agent installation is available for systems which require scripted actions to run directly on the system when an alert condition is raised.



4.3 Front end development

4.3.1 User interface Design

The user interface of the uptime monitoring app is simple and convenient for the users to explore and use. On visiting the page, you are welcomed with a homepage that eases you with a simple but detailed explanation of what the app is all about. You can then navigate to create your account because non-account holders cannot access its services.

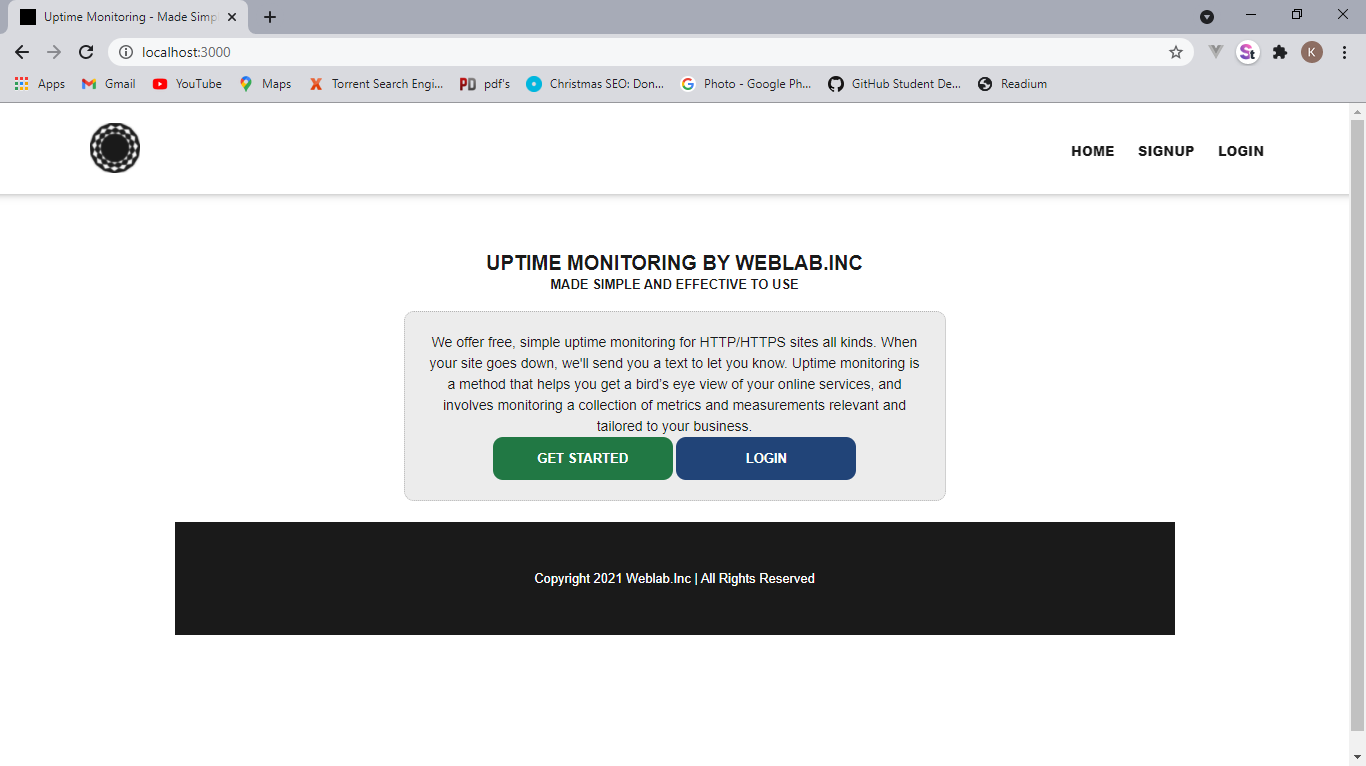


Image 4.1: Homepage

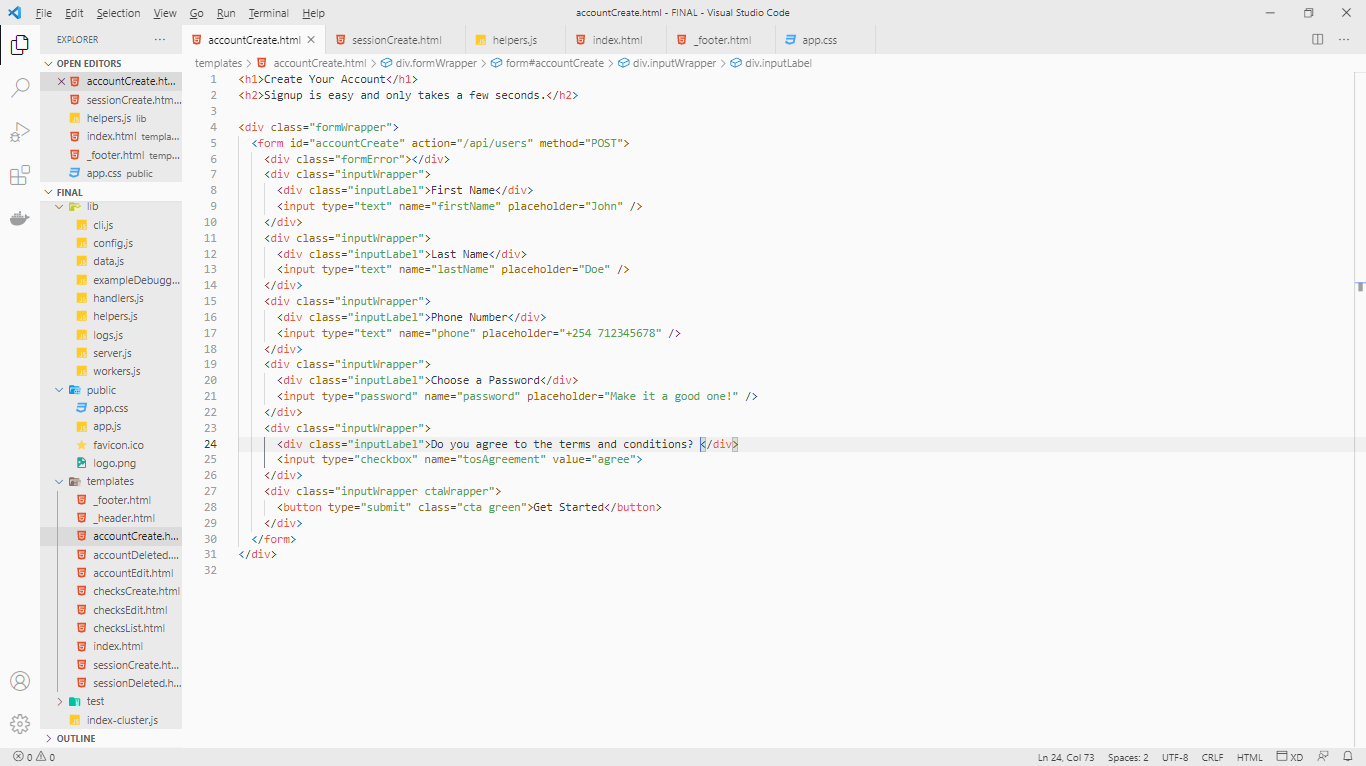
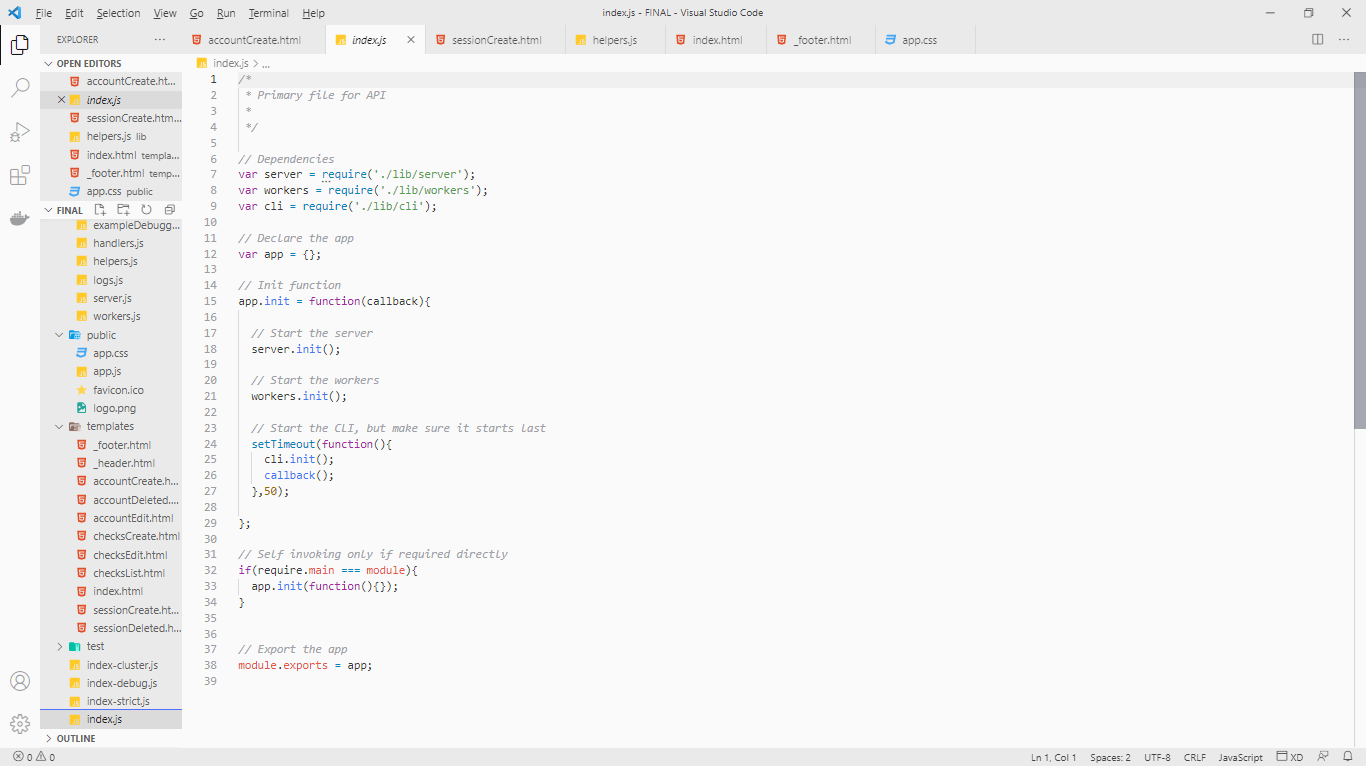


Image 4.2: Code snippet from the sign up



After account creation, you will be redirected to the inner workings of the webapp from which you can choose what URL is to be checked for. The response codes that can be checked for by the user include:

* 200 – OK. The request has succeeded.
* 201 – created. The request has succeeded and a new resource has been created as a result. This is typically the response sent after POST requests, or some PUT requests.
* 301 – moved permanently. The URL of the requested resource has been changed permanently.
* 302 – found. This response code means that the URI of requested resource has been changed temporarily. Further changes in the URI might be made in the future. Therefore, this same URI should be used by the client in future requests.
* 400 – bad request. The server could not understand the request due to invalid syntax.
* 403 – forbidden. The client does not have access rights to the content; that is, it is unauthorized, so the server is refusing to give the requested resource.
* 404 – not found. The server cannot find the requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 to hide the existence of a resource from an unauthorized client.
* 406 – not acceptable. This response is sent when the web server, after performing server-driven content creation, doesn't find any content that conforms to the criteria given by the user agent.
* 500 – internal server error. The server has encountered a situation it doesn't know how to handle.

If on the check box the user checks several codes, it will return “up” to show if the website’s URL entered displays any of the codes chosen.

4.3.2 User Interface modules

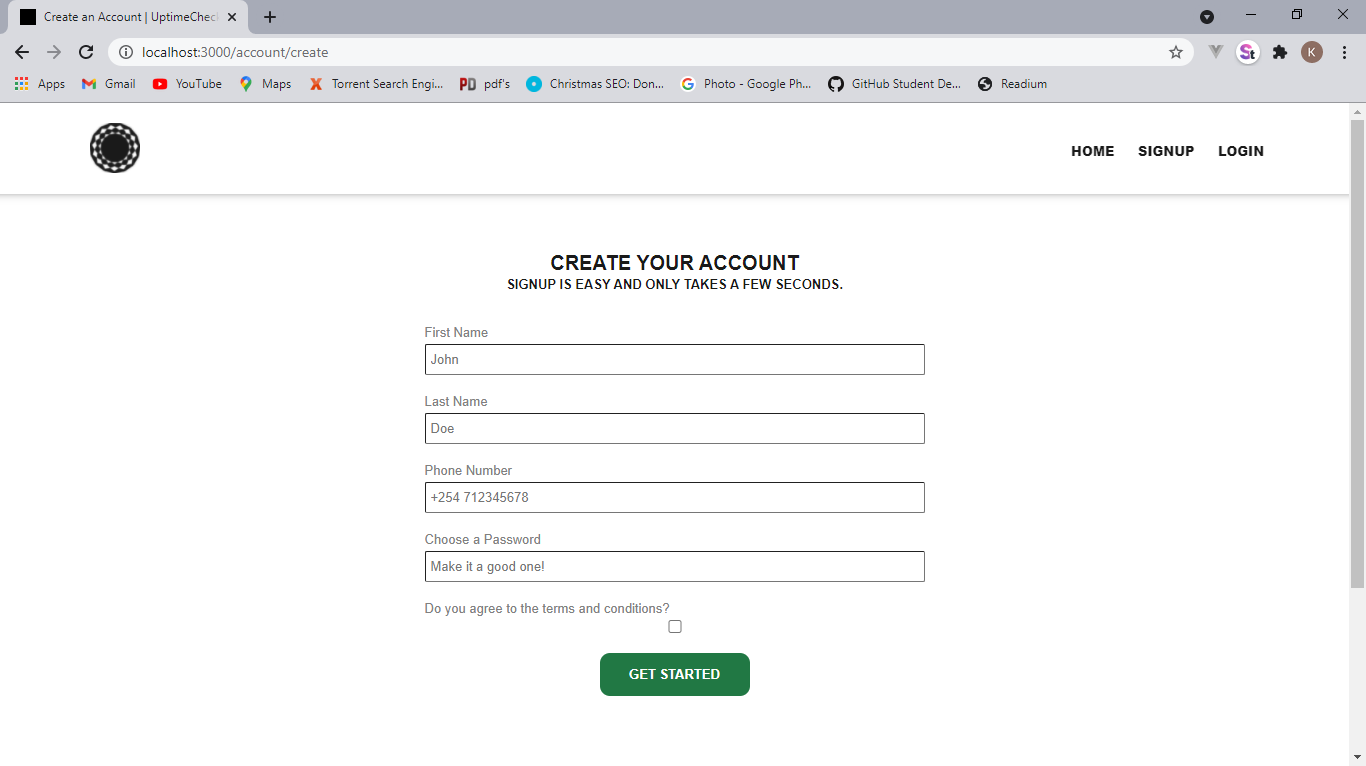


Image 4.4: Account creation page

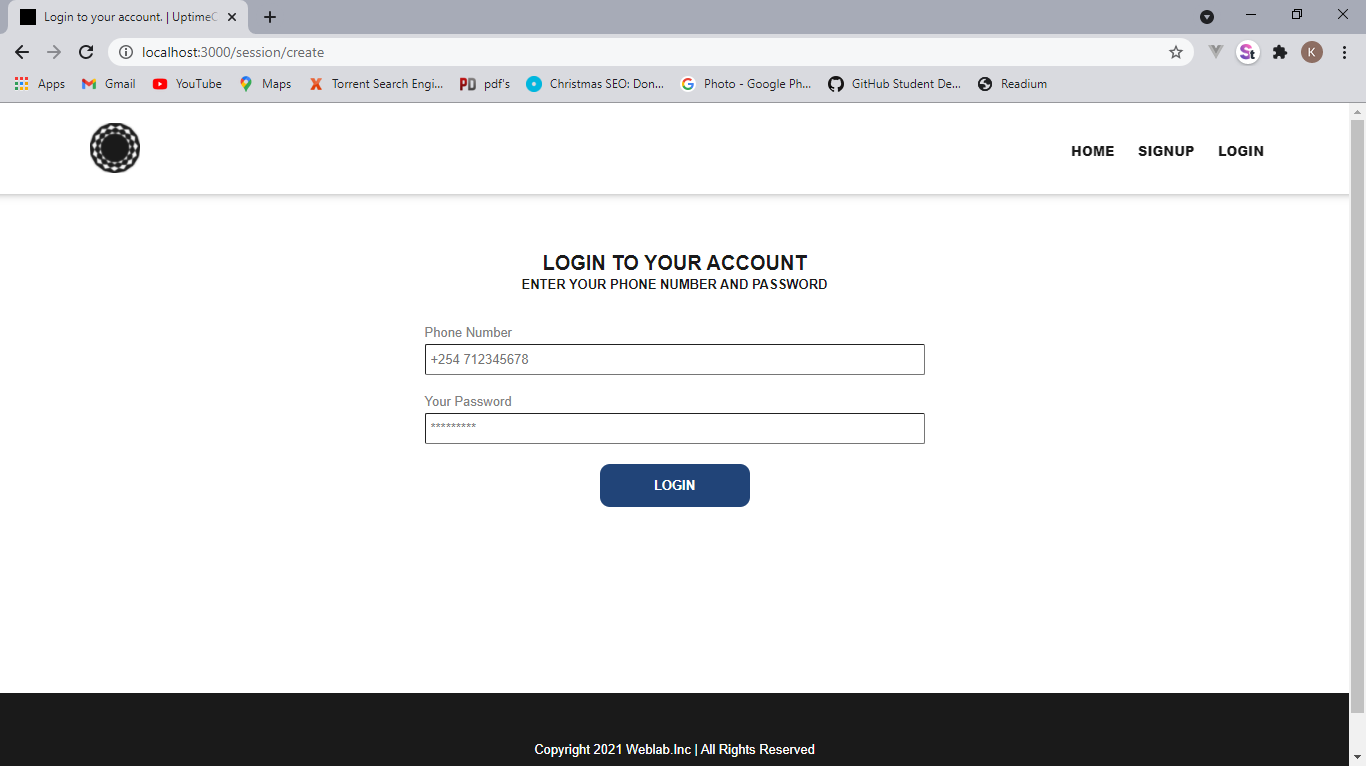


Image 4.5: Log in page

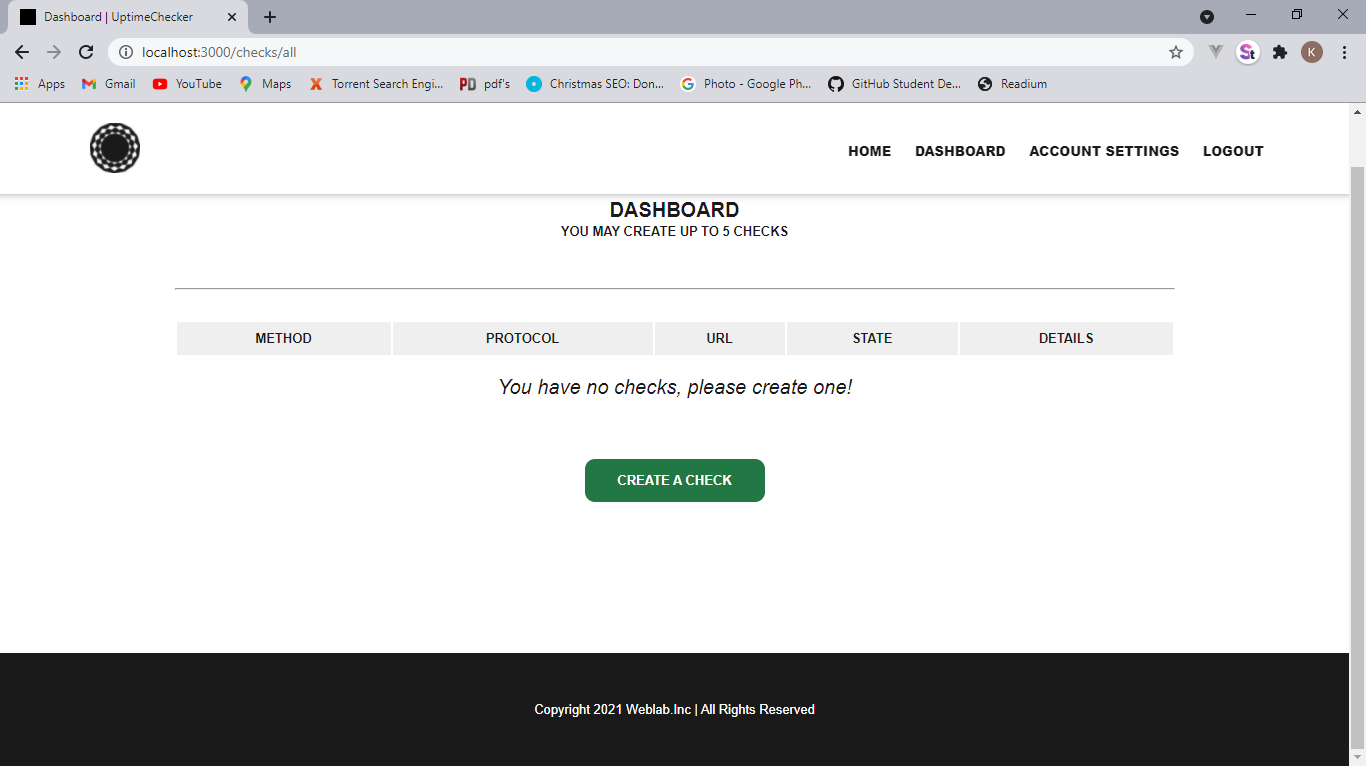


Image 4.6: the dashboard

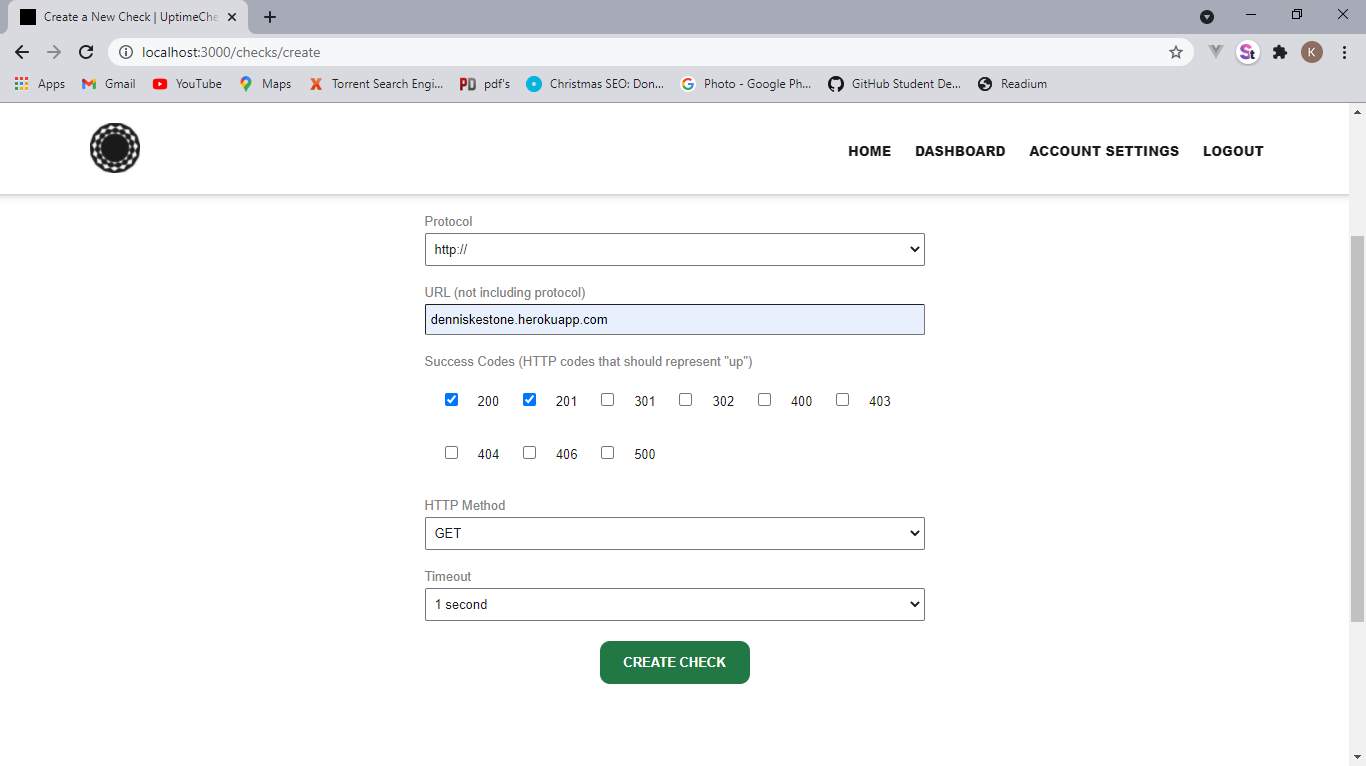


Image 4.7: configuring protocols

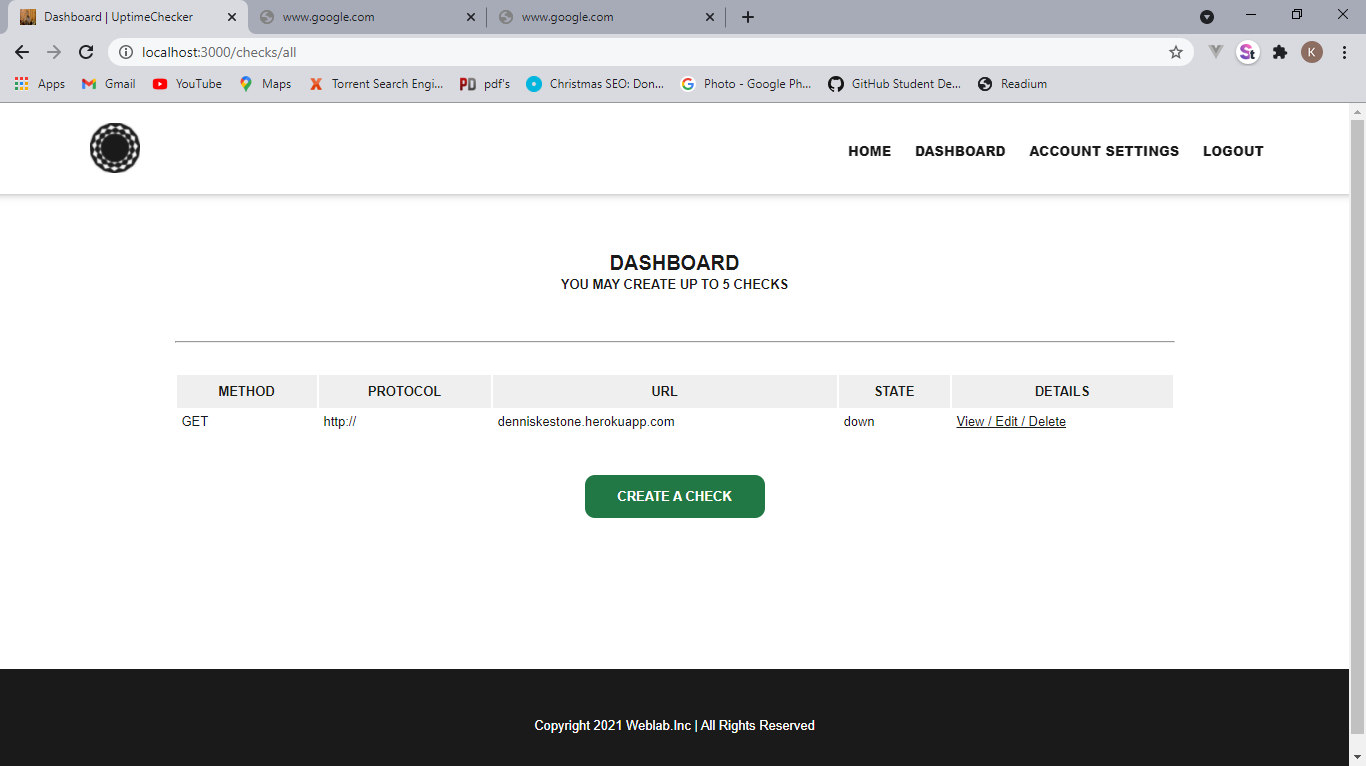


Image 4.8: Check the status of the already created check

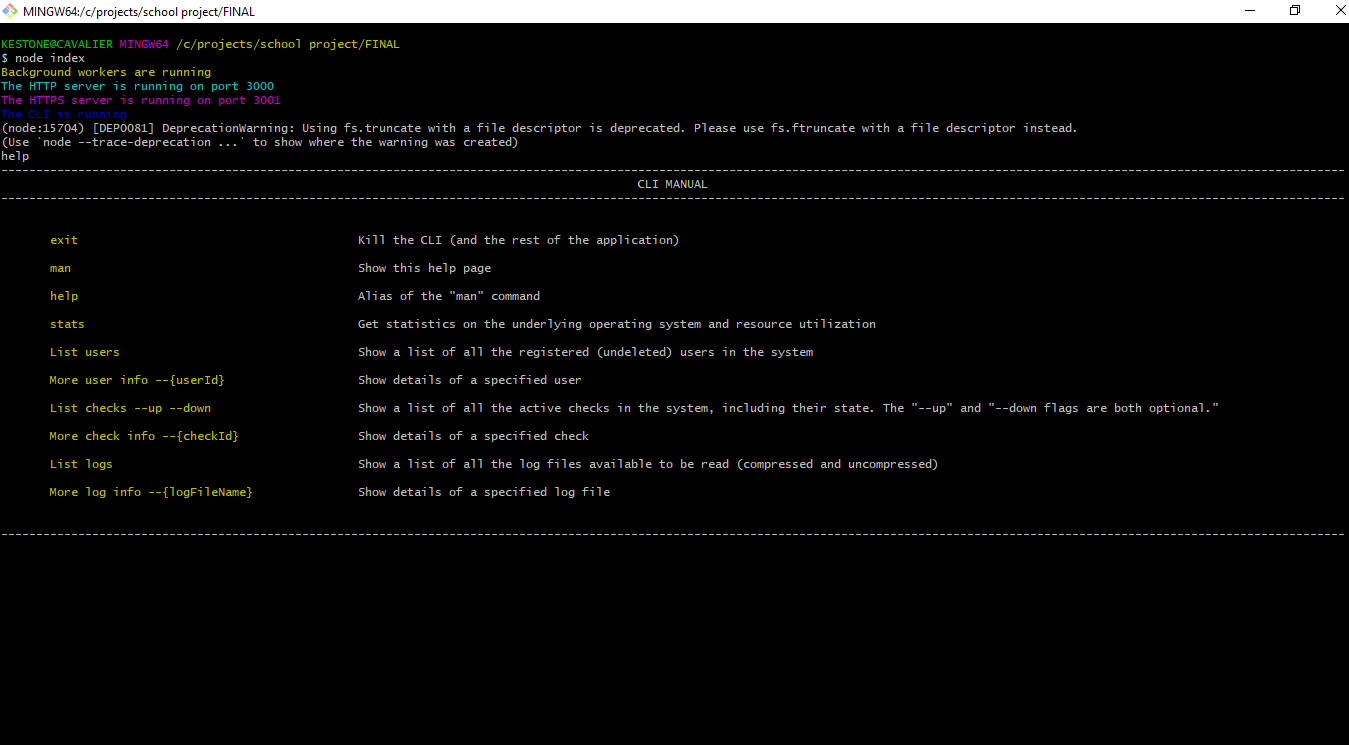


Image 4.9: admin panel CLI

4.4 Back-end development

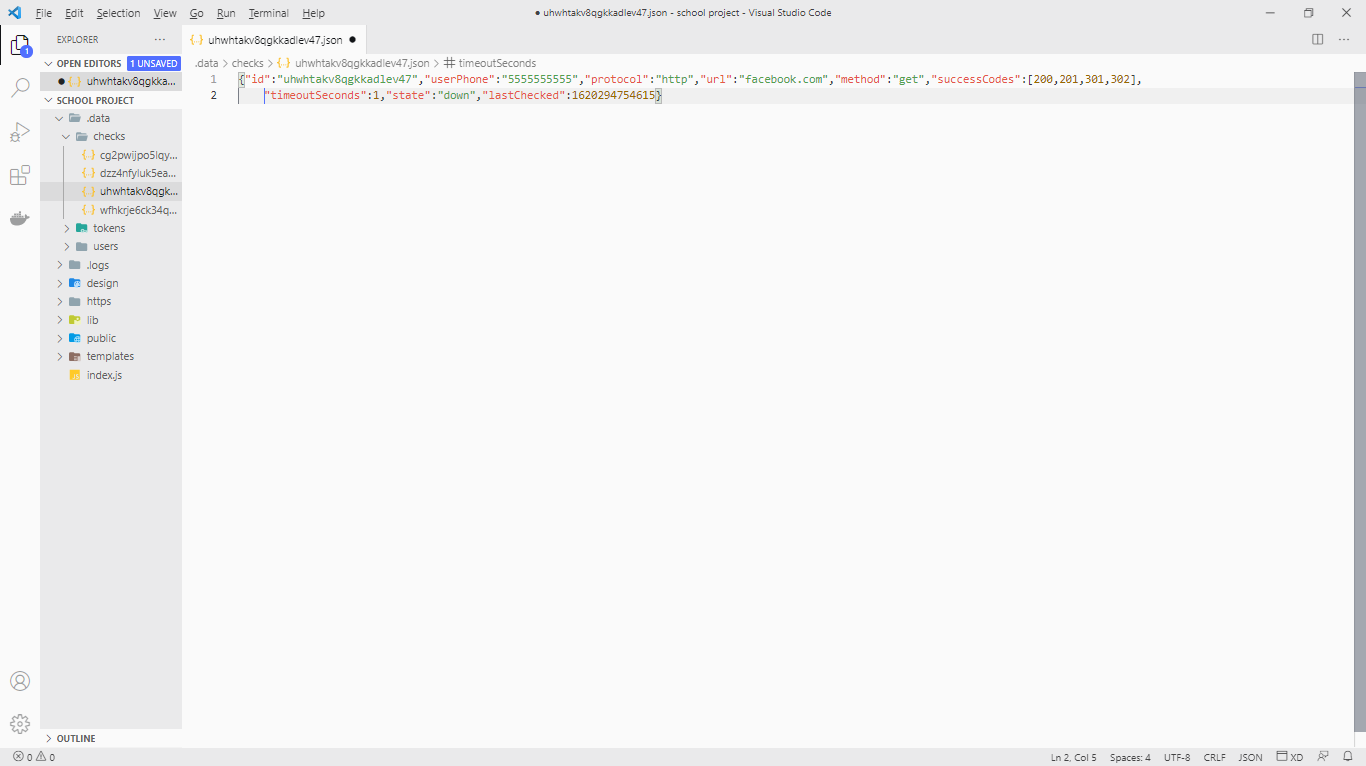


Image 4.10:Json code

Here, data was stored in json format and can be moved to a relational database in the case scaling up is needed. The user on creation of an account, their data is stored in a .data folder as a json document and their password is hashed using crypt.js. this ensures their privacy and even the admin cannot know the password of their users.

4.5 Deployment methods

Based on user access, d**eployment of this project is done by an Administrator by use of a host provider i.e., Heroku, Microsoft azure or even AWS.**

This is a standard deployment approach in which Desktop Central allows administrators to distribute applications to target machines remotely right from the web console. Administrators can deploy software from a centralized web console with silent installation switches in place. SaaS might also be an option.

4.6 Conclusion and Future work

Hosted applications have been gaining a lot of popularity lately. The term everyone is using for this way of delivering software is SaaS, Software as a Service. The question is if SaaS is just a hyped, temporary trend, or if it is here to stay.

We will first look at the different pros and cons of the SaaS model from both a customer and developer point of view and then look at what is necessary for SaaS to reach widespread acceptance.

**Customer point of view:**

Pros:

* Access your applications and documents from any computer.
* No installation needed.
* No administration costs.
* Runs on any computer with a Web browser and Internet connectivity.

Cons:

* Potential privacy/security issues (important data being stored outside your control).
* If you’re offline, you can’t work.

**SaaS developer (SaaS company) point of view:**

Pros:

* Cross-platform compatibility (you target Web browsers, not operating systems).
* Central point of maintenance. New updates immediately available to all customers.
* No piracy.
* Potentially less support issues, and you only need to support the current version of the software.

Cons:

* Vast amounts of computer power, storage space and bandwidth needed on the backend, depending on the application.
* Potentially added administration costs maintaining the infrastructure.
* If something happens centrally, it affects all users at once.

**The challenge – How to tackle the cons**  
For the SaaS model to become a long-term success and reach mainstream acceptance the pros need to outweigh the cons. Not only that, the cons need to be dealt with.

* **Internet connectivity issues:** Reliable broadband Internet connections are already very common both at work and at home, and wireless Internet access is available in many places you would work outside the office. There are even airlines with Internet access available during flights. As the world becomes more and more connected, this issue will simply cease to exist.
* **Protecting confidential data:** There are a lot of third-party services that have critical and often confidential information on their hands: banks, electronic exchanges (stocks, electricity, etc.), law firms, advertising agencies, etc. This kind of trust needs to be extended to SaaS companies, and they need to show themselves worthy of it.
* **Site downtime affecting all customers:** The SaaS company needs to have redundancy for as many elements of their hosted architecture as possible. Ideally they would even have geographically divided setups, so if one site goes down, there is still one working location.
* **Added administrative costs for the SaaS company:** The cost of maintaining the infrastructure the software is running on is most likely offset by the other benefits such as less piracy, less support issues, immediate automatic upgrades for all users, etc. Since this is a problem for the developer and not for the user, it will at least not affect the general acceptance of SaaS in the eyes of the customers.

**The SaaS model doesn’t suit all applications**  
Any application that stretches either RAM, CPU, storage or bandwidth usage on the server end will most likely be very inefficient as a hosted application. Examples include games, 3D modelling and rendering software, image and video editing software, etc. It will remain more practical to keep this as regular software running on local workstations.

**Conclusion**  
Are hosted applications the future? Yes, definitely, but not for everything. There will still be demanding software that is better suited for running locally on a workstation and uptime monitoring might function well if placed on this path.