Independent University, Bangladesh

Cloud Based System Monitoring Application

An undergraduate internship report submitted by

Tasmia Omar Rahi

ID: 1720993

In consideration of the partial fulfillment of the requirements for the degree of

Bachelor of Science

In

Computer Science

Supervised by: Mohammad Noor Nabi

Department of Computer Science & Engineering

School of Engineering, Technology & Sciences

Letter of Transmittal

Date:

Mohammad Noor Nabi
School of Computer Science and Engineering
Independent University Bangladesh.
Subject: Submission of Internship Report.

Dear Sir,

It is a great pleasure to submit my report on Internship at Accomium. I have tried to narrate my project works, achievements and experiences in this report. All the works presented here are done with utmost sincerity and honesty.

During the internship period, I have served in Accomium for three months where I have not only gained real life work experience but understood the process of the department and its various aspects. This report includes a detailed review of the office as well as the functionalities of the department. As a document of my effort during the internship periods I have conducted all the project works that I have done during my internship periods, especially their requirement, functionalities and technical specifications.

I pray and hope this report will be quite interesting and fulfil your expectations. I have tried my best to avoid my deficiencies and hope that my report will satisfy you. I also would like to thank you again for giving me the opportunity to submit this report.

Sincerely,
Tasmia Omar Rahi
Id - 1720993

Letter of Endorsement

To Whom It May Concern
Subject: Approval of the report.

This letter is to certify that all the information mentioned in this document is true and confidential to the company. The project mentioned here has been successfully done with the involvement of Tasmia Omar Rahi, Bachelor in Computer Science, Independent University, Bangladesh (IUB).

I wish her all the best and hope she will lead a successful career.

Internship Supervisor
Signature ……………………...
Rajib Hossain Pavel
Head
Office of Career Guidance
Accomium

Approval

This report entitled
Junior Developer
By
Tasmia Omar Rahi
Has been approved by
The Department of Computer Science and Engineering
Independent University, Bangladesh (IUB)
Supervisor: **Mohammad Noor Nabi**

The final copy of this report has been examined by the signatory and I find that both the contents and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Signature ……………………..

Date ……………………..

Originality Statement

This is to certify that the report titled “Cloud Based System Monitoring Application” was completed by Tasmia Omar Rahi (1720993) submitted in partial fulfillment of the requirement for the Degree of Computer Science from Independent University, Bangladesh (IUB). It has been completed under the guidance of Mohammad Noor Nabi (Internal Supervisor) and Rajib Hossain Pavel (External Supervisor). I also certify that all my work is original and has not been submitted earlier to this university or any other institution. All the sources of information used in this Project Report has been duly acknowledged in it.

Signature

…………………..

(Tasmia Omar Rahi)

Table of Contents

Table of Figure

Acknowledgement

I am using this opportunity to express my gratitude to everyone who supported me throughout the course. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the internship. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the course.

I am grateful to **Independent University, Bangladesh (IUB)** for offering me this course and I express my thanks to my faculty and internship supervisor, **Mohammad Noor Nabi**.

I want to stretch out my earnest gratitude to him. I am profoundly obligated to **Independent University, Bangladesh (IUB)** for their direction and consistent supervision just as for giving important data with respect to the course.

I am very much grateful towards my external supervisor, **Rajib Hossain Pavel** for his guidance and support that I needed for the fulfillment of my internship and project. I might want to offer my extraordinary thanks and gratitude to the **Accomium** developers and employees for giving me such consideration and time and directing through the internship.

I also want to thank my parents for who have not only supported me financially but have always believed in me and always motivated me to accomplish the things that I have achieved.

Abstract

My internship was at the Frontend development team of Accomium. I was initially tasked with handling their user interfaces using web scraping and other techniques. As data from different sources have different formatting, I had to format the interfaces into a singular format that was suitable for users or clients. I was also expected to enter the data to make sure that it doesn work without any corruption, or partial entries that might not cause problems with the functions of the applications that use the data. The web scraping is done using Python and with the aid of Jupyter Notebook. The data from web scraping is saved in the form of comma separated value (.csv) files. While data collection is handled by Python, the data processing is done partially by a Laravel application. The Laravel application takes in the csv files and inserts them into the database and MySQL queries, designed to clean the data in a way that is required by the user of that data, are then used. My final assignment and the main topic of this report is to automate the whole process described above. This was to be done in order to make the client's life easy but store their important document and also eliminate the chances of human error while performing the actions described above. While working at Accomium, I was fortunate enough to be introduced with technologies such as HTML, CSS, Javascript, PHP, Laravel, MySQL, Python.

Chapter 1

Introduction

1.1 Introduction

"The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers. Cloud servers are located in data centers all over the world. By using cloud computing, users and companies don't have to manage physical servers themselves or run software applications on their own machines.
The cloud enables users to access the same files and applications from almost any device, because the computing and storage takes place on servers in a data center, instead of locally on the user device. This is why a user can log into their Instagram account on a new phone after their old phone breaks and still find their old account in place, with all their photos, videos, and conversation history. It works the same way with cloud email providers like Gmail or Microsoft Office 365, and with cloud storage providers like Dropbox or Google Drive.

On the other side "Monitoring" refers to regular observation and recording of activities taking place in a programme. It is a process of routinely gathering information on all aspects of the project. To monitor is to check on how activities are progressing. It is observation ─ systematic and purposeful observation. Monitoring also involves giving feedback about the progress of the project to the donors, implementers and beneficiaries of the project. It enables gathered information to be used in making decisions for improving project performance.

With the importance of cloud and monitoring stated above, I will now move on to the project this report is based upon. As the name, " Cloud Based System Monitoring Application ", suggests; this project is based on the use of manual or automated IT monitoring and management techniques to ensure that a cloud infrastructure or platform performs optimally.

1.2 Project Background

Accomium, is responsible for running and maintaining various websites and also designing and developing new websites. Due to the type of work, there is continuous demand for different types of services that is used to maintain various websites. The idea of the project came from the businessman’s request for a unified cloud that would perform the actions that they were performing at the time of the inception of the project.

Accomium had existing softwares and applications in place to manage the cloud and process them. But the applications were not linked together, for example, when a client asked to put their data on the cloud, they can easily complete their work but there was no mechanism in place to monitor the data in the database. So, a person had to manually start the next step of the process. This was really time consuming and was totally based on the timing of the person assigned to perform the steps.

The project was a brainchild of the Chief Technical Officer and the Software Architect, who realized that it would be a great service and that time can be redistributed across many other tasks as it will work automatically. The project is primarily the automation of a series of tasks in a manner that’d see them being executed sequentially. The project work was distributed across programming languages such as Python and PHP, frameworks such as Laravel, database of MySQL and lastly, programming on the cloud front using the Amazon S3 platform. The steps of the projects are clearly explained in the Objectives and are also explained in more detail in the next few chapters.

1.3 Objectives

Automate the following steps:

- **Check Integrity of cloud:** This step consists of actions performed by the data analyst/developer to check for errors in the cloud that were scraped or collected by any other means. These actions include, going through the files manually or using MySQL queries to check for errors.
- **Compress the file in the cloud:** This is done using softwares such as Winrar, and this is performed manually by the user. The reason for this step is that scraped files are usually saved in CSV files and when a compression algorithm such as zip is applied, the file sizes get reduced significantly.
- **Upload File to Cloud:** Most of our file is being hosted in the cloud platform of our own and there are specific S3 buckets for each project. Developers upload the file to the bucket that is designated to their project. It must be mentioned up to this time, the file is classified as raw data, although it has gone through some checks by the developers, the developers didn't run any cleaning or formatting scripts. This is done to preserve a copy of the scraped file, for it to be used in any other projects.
- **Send the notification to a Laravel Application:** The notification for the file that was uploaded to cloud is generated and sent to the Laravel Application using a POST request performed from Python.
- **Download and Decompress:** The file is then downloaded and decompressed by the Laravel Application.
- **Insert and Clean:** The Laravel application will insert the file into a database and run a batch of MySQL queries to clean the file.
- **Export Data:** The application will then export the file, in the form of an sql dump file and upload it to the cloud.
- **Notify:** The application would then notify the developers via email about the new addition of data to the cloud and also provide the link to the cleaned data.

1.4 Major Responsibilities

- Plan and design the approach to be taken in order to make this project a reality.
- Discuss with the team lead about the approach and give him an approximate time in which I expect to complete the project.
- Divide the project into parts of equal sizes.
- Start the implementation of the project and give weekly progress reports to the Software Architect and the Team Lead.
- While the automation project gets developed, manage the old system and keep feeding properly processed data to the system, manually.

1.5 Minor Responsibilities

- Coordinate with and help other developers who use the cloud that I processed.
- Write SQL queries when raw queries are needed to be run on production servers.
- Do research on various topics that are of interest to the organizations and prepare reports to present to the decision makers.

Chapter 2

Company Profile

2.1 Company Background

Since their inception in 2018, the people in Accomium have relentlessly pursued excellence in creating the best solutions for their clients. Their motto has been set from the start to rack their collective brains and constantly push the envelope of what can be done with their technical process. The employees and upper management view themselves as problem solvers. This is the reason they wanted their name to reflect that very fact. After brainstorming what the name of the company should be, they settled on Accomium. The people at Accomium help their clients to find their “Premium” way out of all problems in which the clients find themselves to be lost. And they are considered cost which is “Accounting” to be less compared to the other company. That’s how the organization became Accomium.

2.2 Vision, Mission and Values

To pioneer highly sophisticated software solutions to contribute to the technological integration and evolution within the business world.

- **Client Centric:** No matter what we put our clients above all else without exception and work towards giving them the possible service.
- **Tech Oriented:** We always work in teams to benefit from the collaboration of multiple “idea-generators” and formulate the best possible solution.

● **Integrity:** Our success is derived from the trust of our clients that we have earned through hard work and delivering quality output

● **Resolute:** We tackle all problems head- on and chase after the optimal solution in a steadfast manner. 16

● **Forever Students:** For us, learning never stops. We seize every opportunity to learn something new to better ourselves to serve our clients better.

● **Forward Thinkers:** You will find the team lead assigned to you available during your work hours and the rest of your team will also maintain overlapping hours.

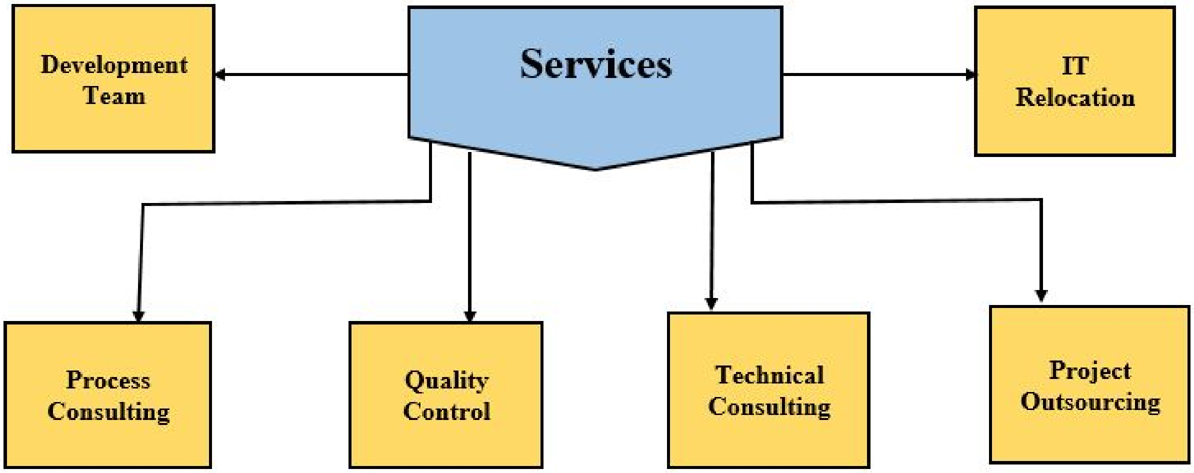


Figure: 2.2.1

2.3 Services

● **Development Team:** We will build the right team for you from the ground up. Enjoy your freedom to manage your dedicated offshore team with the added benefit of scalability.

● **Project Outsourcing:** Take advantage of our wide range of core technical expertise. Clients have the option to engage any resource at our disposal and scale up and down as per their current need.

● **Process Consulting:** We will re-engineer your existing processes and construct new ones to optimize your entire development life cycle and maximize overall efficiency.

● **Quality Control:** Our design will help you trace problems and improve quality. We will set up an elaborate regiment of tests to detect anomalies or discrepancies and draw necessary attention for expedient resolution.

● **Technical Consulting:** Our goal is to guide you to the right technical recipe. Utilize our experts and they will help you to plan and execute your development schedule using their exhaustive know-how.

● **IT Relocation:** We make it easier for you to relocate resources from our location to your office premises. You will benefit immensely through access to our “cream of the crop” resources.

2.4 Technologies Supported

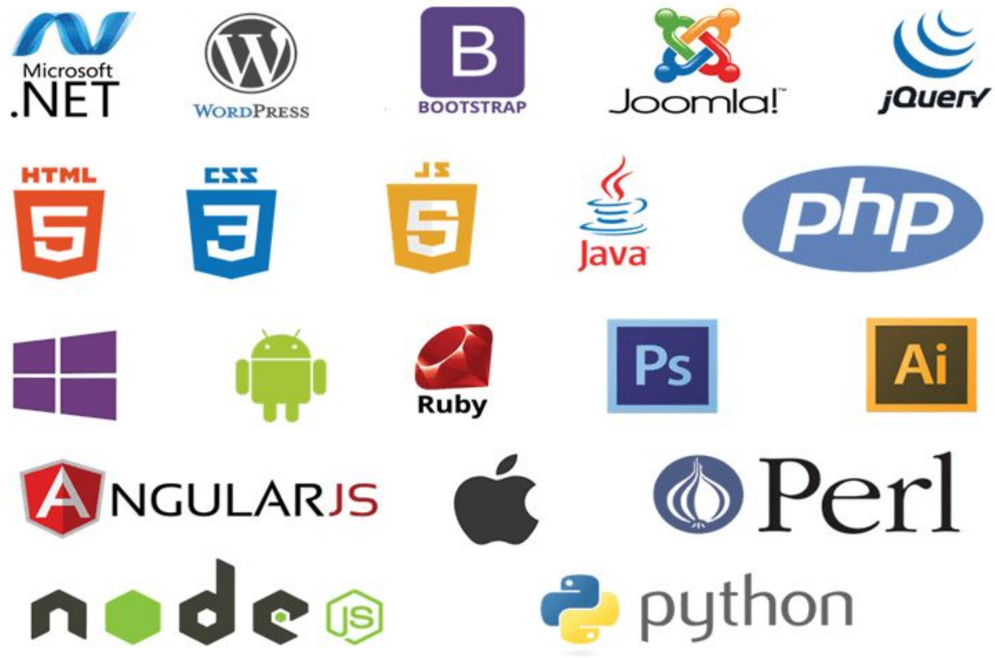


Figure: 2.4.1

Figure 2.4.1 shows all the technologies supported for development in Accomium. This includes; PHP, Microsoft .Net Framework, Android, iOS, Javascript, Node.js, Angular, Joomla Python, Ruby on Rails.

2.5 Contacts and Address

Accomium
Dokkhin-Khan,Uttara, Dhaka-1230
+8801907750518
www.accomium.com

Chapter 3

Literature Review

3.1 Data Storage

Data storage is a general term for archiving data in electromagnetic or other forms for use by a computer or device. Different types of data storage play different roles in a computing environment. In addition to forms of hard data storage, there are now new options for remote data storage, such as cloud computing, that can revolutionize the ways that users access data.
The amount of data stored in digital form worldwide has on the average doubled every 9 months, over many years, which is twice the rate for increase of computing power, predicted by Moore's law. This doubling of stored information, called storage law, is one of the reasons for motivation for cloud. Irrespective of whether this increasing volume of data will support exploration in commercial or scientific activity, the data is potentially valuable information
New technologies and tech theory promote the continual expansion of data storage capability. New solid state drives can hold enormous amounts of data in a very small device, enabling various kinds of new applications for many industries, as well as consumer uses. Cloud services and other new forms of remote storage also add to the capacity of devices and their ability to access more data without building additional data storage into a device. [1]

- **Cloud:** The definition for the cloud can seem murky, but essentially, it's a term used to describe a global network of servers, each with a unique function. The cloud is not a physical entity, but instead is a vast network of remote servers around the globe which are hooked together and meant to operate as a single ecosystem. These servers are designed to either store and manage data, run applications, or deliver content or a service such as streaming videos, web mail, office productivity software, or social media. Instead of accessing files and data from a local or personal computer, we are accessing them online from any Internet-capable device. [2]

[1]

[2]

- **cloud storage:** For businesses, switching to cloud computing removes some IT costs and overhead: for instance, they no longer need to update and maintain their own servers, as the cloud vendor they are using will do that. This especially makes an impact for small businesses that may not have been able to afford their own internal infrastructure but can outsource their infrastructure needs affordably via the cloud. The cloud can also make it easier for companies to operate internationally, because employees and customers can access the same files and applications from any location. [3]

- **Physical data storage:** It is between random access memory (RAM) and associated formats, and secondary data storage on external drives. Random access memory is stored in integrated circuits for immediate use. While the data stored on hard drives, disks, flash drives and new solid state data storage units is archived for event-based access or research activities initiated by an end user. [4]
- **Monitoring:** Monitoring is the systematic process of collecting, analyzing and using information to track a programme's progress toward reaching its objectives and to guide management decisions. Monitoring usually focuses on processes, such as when and where activities occur, who delivers them and how many people or entities they reach. Monitoring is conducted after a programme has begun and continues throughout the programme implementation period. Monitoring is sometimes referred to as process, performance or formative evaluation. [5]

3.2 Web Application

Web Applications can be defined as any application program that is stored in a remote server and that can perform some function for the user. Web applications can be designed for a wide variety of uses and can be used by anyone; from an organization to an individual for numerous reasons. Commonly used Web applications can include webmail, online calculators, or e-commerce shops. Some Web apps can be only accessed by a specific browser; however, most are available no matter the browser. Many web applications exist to automate tasks such as automated 21

[3]
[4]
[5] .

emailing systems, application programming interfaces, and user monitoring systems which serve the organization behind the web platform. Similar to the development of websites, web application development can be divided into two parts, Front End and Back End development. This project is partially based on the development of the back end of a web application that is used to clean and store data. [6]

3.3 Cloud monitoring:

According to both cloud and monitoring we can simply describe that cloud monitoring is a method of reviewing, observing, and managing the operational workflow in a cloud-based IT infrastructure. Manual or automated management techniques confirm the availability and performance of websites, servers, applications, and other cloud infrastructure. This continuous evaluation of resource levels, server response times, and prediction of vulnerabilities in future issues before they arise. Basically it is a use of manual or automated IT monitoring and management techniques to ensure that a cloud infrastructure or platform performs optimally.

- **Types of Cloud Services to Monitor:**
  There are multiple types of cloud services to monitor. Cloud monitoring is not just about monitoring servers hosted on AWS or Azure. For enterprises, they also put a lot of importance into monitoring cloud-based services that they consume. Including things like Office 365 and others.
  1. **SaaS** – Services like Office 365, Salesforce and others.
  2. **PaaS** – Developer friendly services like SQL databases, caching, storage and more.
  3. **IaaS** – Servers hosted by cloud providers like Azure, AWS, Digital Ocean, and others.
  4. **FaaS** – New serverless applications like AWS Lambda and Azure Functions.

[6]

3.4 Related Software:

- **Amazon Cloudwatch:** Amazon Web Services offers to monitor cloud resources and applications running on Amazon AWS. AWS resources such as Amazon EBS volumes and Amazon RDS DB instances. We can also use it to set alarms, store log files, view graphs and statistics, and monitor or react to AWS resource changes.

  Amazon Cloudwatch gives us an insight into our system's overall health and performance. We can use this information to optimize our application's operations. The best part of this monitoring solution is we don't need to install any additional software. It is an excellent practice to have multi-cloud management strategies. They give us cover in case of incidences such as when Amazon Web Services went dark. [7]

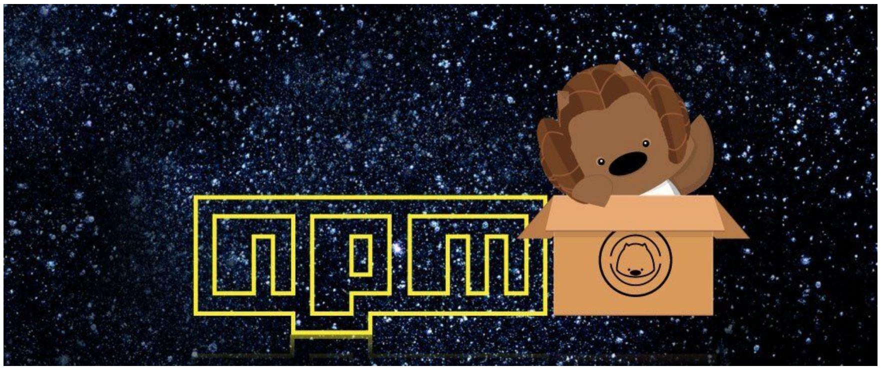
  - **AppDynamics:** AppDynamics provides cloud-based network monitoring tools for assessing application performance and accelerating operations shift. We can use the system to maximize the control and visibility of cloud applications in crucial IaaS/PaaS platforms such as Microsoft Azure, Pivotal Cloud Foundry, and AWS. AppDynamics competes heavily with other application management solutions such as SolarWinds, Datadog, and New Relic. The software enables users to learn the real state of their cloud applications down to the business transaction and code level. It can effortlessly adapt to any software or infrastructure environment. The new acquisition by Cisco Systems will only magnify AppDynamic's capabilities. [8]

[7]

[8]

3.4 Technologies Used

3.4.1 npm for software Package Manager



npm is committed to making JavaScript development elegant, productive, and safe. The free npm registry has become the center of JavaScript code sharing, and with more than one million packages. The largest software registry in the world. Our other tools and services take the registry, and the work. This package contains all the files needed for a module and modules are the JavaScript libraries that can be included in Node project according to the requirement of the project. [9]

3.4.2 HTML Script Development

[9]

The HTML is an incredibly powerful tool for interactively developing and presenting data science projects. It integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. This intuitive workflow promotes iterative and rapid development, making CSS an increasingly popular choice at the heart of contemporary data science, analysis, and increasingly science at large. The project requires a tool that can hold on to data from a previous section of the code even if the next section of the code gets an exception. HTML is great for SQL development because HTML supports almost all of the modules of SQL. HTML also is a cross platform tool and can be used with any mainstream operating system; that makes it an ideal software to run on a server. [10]

3.4.3 Backend Web Development using Laravel

Laravel is an open source framework for web development based on PHP. It was created by Taylor Otwell, and was intended for web application development. After the introduction of this framework, it has gained massive popularity among the web developers who were previously using PHP or any other framework based on PHP. It's architectural pattern is modeled on the Model View Controller (MVC) and is based on the PHP web application framework Symfony. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar. One of the reasons for

[10]

it's industry wide adaptation is that it has a comprehensive set of documentation and outstanding support from the Laravel developers community. Regular updates and seamless integration of new features has also contributed to its popularity. It is widely debated on whether its merits as an excellent web framework has kept the PHP server side programming relevant in this era of cutting edge technology. The server side application for this project was created using Laravel framework. [11]

3.4.4 Database manipulation using MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube. [12]

[11] .

[12]

3.4.5 Vue.js

Vue.js is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. [13]

3.5 Achievements

3.5.1 Research Skills

Working for this software has increased my research capabilities to a great extent. Due to the type of work that I do here, I always delve into research and development for various projects and while doing this project, I have learned numerous techniques by which I can efficiently retrieve information from a wide range of sources. The art of differentiating false information from useful information is also something that has come to me after researching for this firm.

[13]

3.5.2 Enhancement of Corporate Skills

As the new person in the firm, I have been given a big opportunity to observe and learn about every aspect of the industry. From communicating with the executives of the firm to drafting documents for those very people, I’ve learned the nooks and crannies of the corporate machine. I have also learned many conventions and traditions that are not very easily accessible by industry outsiders.

3.5.3 Communication Skills

As an intern at this, I was required to provide regular presentations and participate in stand up meetings where I was required to give the review of my work and also provide future actions that need to be taken. I was also asked to analyze future projects and whether or not I would be able to complete my task. This improved my communication skills and also helped me learn a lot of techniques for effective communication.

Chapter 4

Cloud Based System Monitoring Application

4.1 Existing System

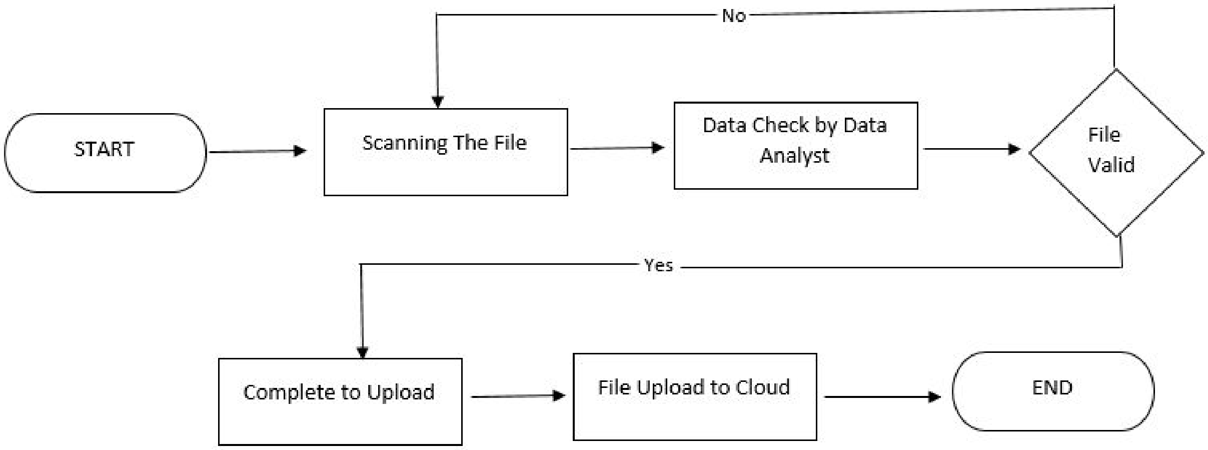


Figure: 4.1.1

The existing system consists of four different parts,

1. Scanning The File: This part is solely designed to scan the file to store on the host computer/server in the form of comma separated value (.csv) files. The parts are primarily written in Python programming language and Jupyter Notebook is used to 29 develop and run the file. After the file is done scanning from a website for a particular topic, it stops and the person in charge of monitoring the file initiates the second part of the system.
2. Mapping and Entry of data to the database: After the scanning file stops and the csv files are collected. These csv files are then fed into a Laravel project that maps the data in the csv files and commits the data of that file to the database of the project. The third step of the process is then started.

3. File Validation Complete: The reason behind this step is because of the undesirable characters that often are present in the file that is collected. These characters cause problems when the file is being used and can often lead to false results and errors. The file is cleaned by using MySQL raw queries written specifically for the purpose of cleaning this file. The queries can be divided into two types,

   i. "Select" queries, these queries are used by the data analyst to find the characters that need to be replaced/removed from the database file.

   ii. "Update" queries, these queries are used by the data analyst to replace/remove characters from the database.

4. Generation of SQL Dump file: The data analyst then generates a sql dump file using the software called "SQLyog." This file is then uploaded to the cloud storage that is allocated for this project.

4.2 Problem Analysis of the current system

4.2.1 Employee Requirement Analysis

The employees who use the current system were surveyed and their usage of the manual system was monitored during the planning phase of the project. As I am also a user of the system and am primarily responsible for meeting the data needs of the firm, I can also share my insights on the existing system.

- **Observations on usage of the current system:** The major problem of the current system is timing, it is solely dependent on how the operators and employees on the project handle things and perform certain tasks. For example, the scanning file may complete its scanning on Friday at 8:00pm but Saturday is a weekend and the office doesn't open before Monday, so nothing gets done in the days in between. When the data analyst arrives in office on Monday, he/she then initiates the next process in the system. Then there is the fact that file Validation is a very time consuming task and the employee designated to do that, could be used to perform other tasks that'd contribute to the company's revenue more. The points below summarizes the current system perfectly,

○ Time Consuming

○ Inefficient

○ Unfeasible

- **Results from the interview with employees:** From interviews that were conducted and also from my point of view, a common theme arises. Employees spend more time than they are supposed to while collecting, cleaning and storing file. Below is a pie chart generated using the data from a system used by our firm to monitor the actions of the employee during work hours. Every employee is obligated to enter their task in "Clockify" for record keeping. This chart was generated from the averages of monthly data.

Employee Time Records Clickify

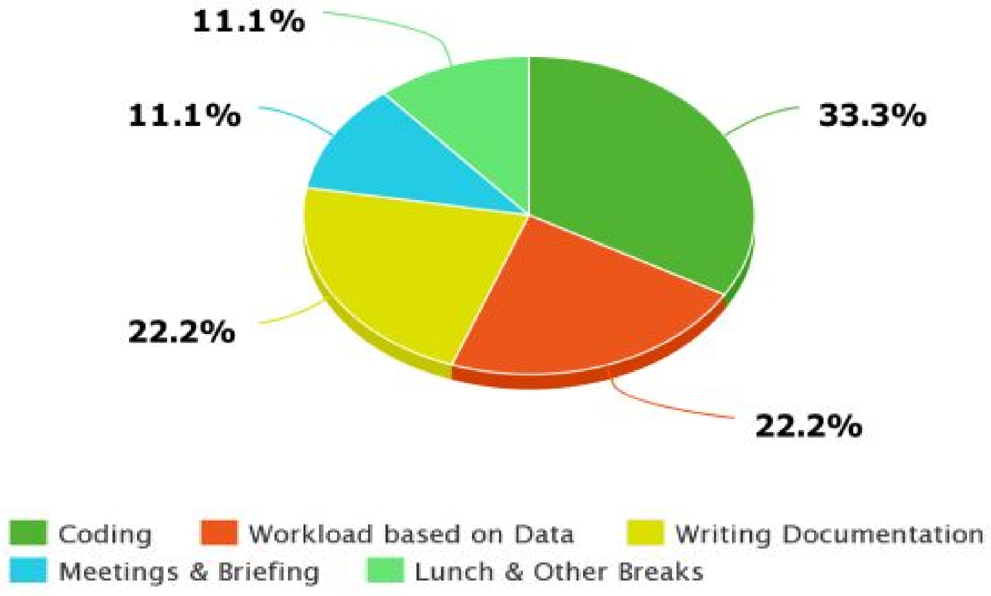


Figure: 4.2.1.1

In the Fig. 4.2.1, we can see that 22.2% of the 9 hour office day is spent on workloads related to data. The reason behind this is explained by one person very clearly, he said, “There is no set of rules assigned to the employees while file validation, and this is partly due to the lack of planning and partly because of the ever changing sources of file for a particular project. The former can be solved using a fixed of validation queries that is used all over the firm by all the developers and later is much more tricky to solve. He also said that transferring the task to an automated system can also help solve the problem of keeping track of the characters to remove and the errors to fix in a particular set of files. All the employees agreed that automation is the way to go.

4.3 Proposed System

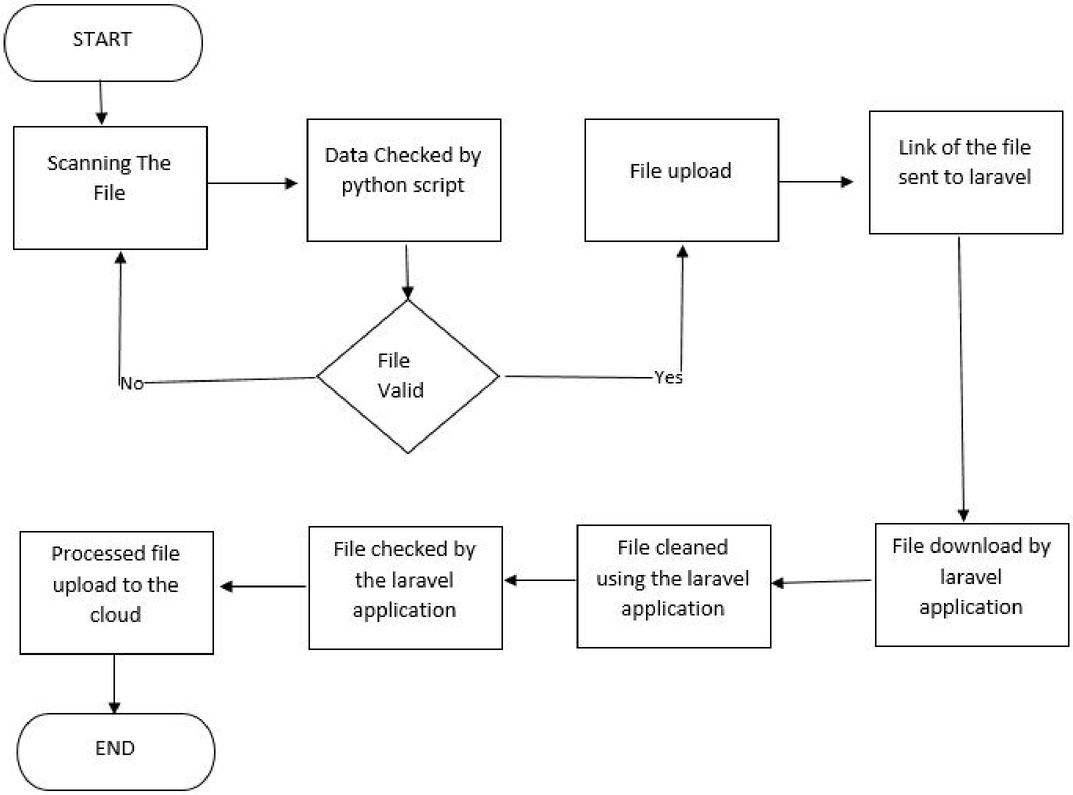

START
Scanning The File
Data Checked by python script
File upload
Link of the file sent to laravel
File Valid
No
Yes
Processed file upload to the cloud
File checked by the laravel application
File cleaned using the laravel application
File download by laravel application
END

Figure: 4.3.1.1

The proposed system chains all the processes mentioned in 4.1 and creates an integrated system that automates the tasks that are carried out by the employees of the organization. The scope of the automation emcompasses various softwares such as Jupyter Notebook, Intellij IDEA, and MySQL database. Taking into account the cross platform nature of the project, tools that are 33 used for this project need to be universal. One such tool or rather a callback command is webhook which was implemented to pass information from Python to Laravel. The proposed system saves time from the perspective of the file and also the perspective of the employees in charge of data. Due to automation, the scanning will be able to bring in more files in a given time and also, more files can be processed and be ready stored in the cloud, in a comparatively small amount of time. The proposed system also gets rid of human errors that might occur during the handling and cleaning of files. These human errors can take a lot of time to rectify and eliminating them is a welcoming addition to the moral of the organization.

4.4 Methodology

4.4.1 The Extent of Automation

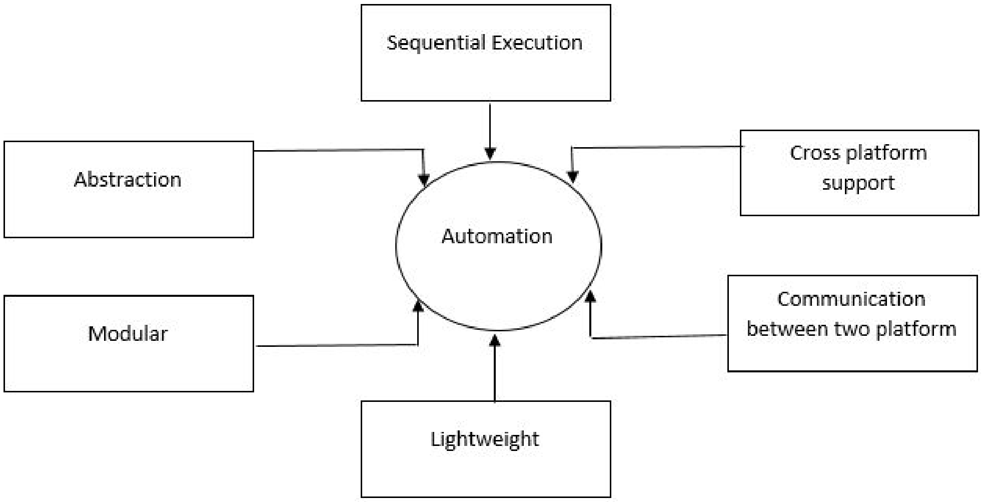


Figure: 4.4.1.1

Currently, the automation is being applied to the parts of the system where we are almost sure it won't cause any problems that would never be caused by the current system. That is one of the reasons behind the use of Laravel applications to process files although we know that Python would do a better job at it. But revamping the system to make it totally Python based would add additional work to the mostly preoccupied software developers. File processing is an important part of the work that this organization does but it merely a part of the total functionality of the company. So, to summarize, we are applying automation to 34 specific parts of the project and not the total project. The reason behind this is because we don't want any disruption in the current workflow.

4.4.2 Abstraction

The scanning and jobs that stitch the processes of the project are designed in such a way that they could be used to join any other tasks with the slightest of modification. They were planned with object oriented programming in mind and made out to be as general as possible.

4.4.3 Functional Requirements

- Supports both Python (Jupyter Notebook) and Laravel.
- Send information from Jupyter Notebook to Laravel Application using webhook.
- Execute Laravel Jobs after information is received via webhook.
- Laravel Job to clean file that is modeled after the SQL raw queries.
- The whole thing should be able to operate in cross-platform, in different operating systems.

4.4.4 Non-functional Requirements

- Properly timed and actions executed in a sequential pattern.
- Good performance and lightweight on the server.
- Modular, so that it is easy to edit and maintain.

4.5 Limitations

I developed this automation software keeping in mind the requirements put forward by our Software Architect. So this makes the software inherently specialized towards the needs of Accomium. So the software cannot be used for automation of any general system, or any other system in that case. Another limitation of this software is that it can't restart a scanning file when it stops running, this is because we are yet to find a way to analyze and act upon the exceptions that scanning might face during runtime. Although we handle all of the known exceptions in the project.

4.6 Future Work

There are plans to create a software using either Django framework from Python or Laravel framework from PHP, to handle all the scanning file and data validation. This mass integration would get rid of bottlenecks that exist in the system while transferring from one platform to another. Two new experimental scanning files that write scanned data directly to the database have already been implemented and are in their testing phase. I single handedly developed one of those scanners and was involved in the development of another.

Chapter 5

Design And Analysis

5.1 USE CASE Diagram

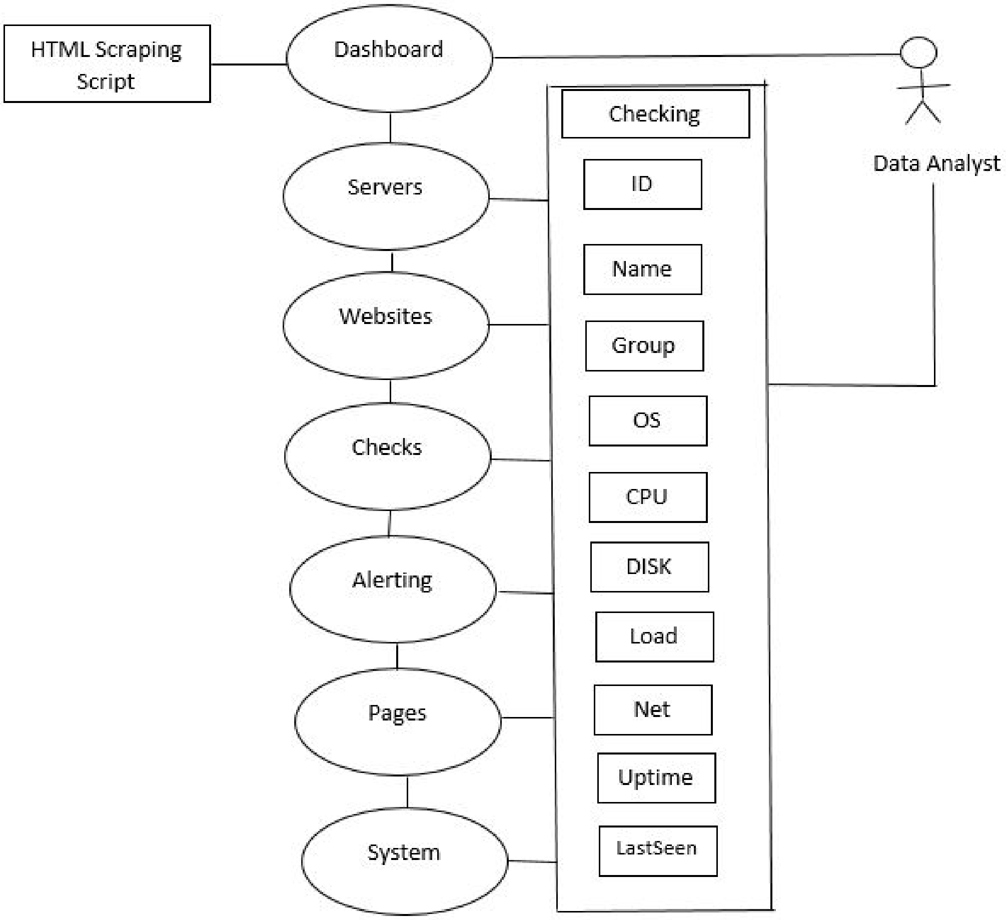


Figure: 5.1.1

In Figure 5.1.1, explains the processes between the Data Analyst, HTML, and Laravel Application. We can clearly see each of the tasks performed by scripts and applications, demonstrating the steps of the project.

5.2 Entity Relationship Diagram

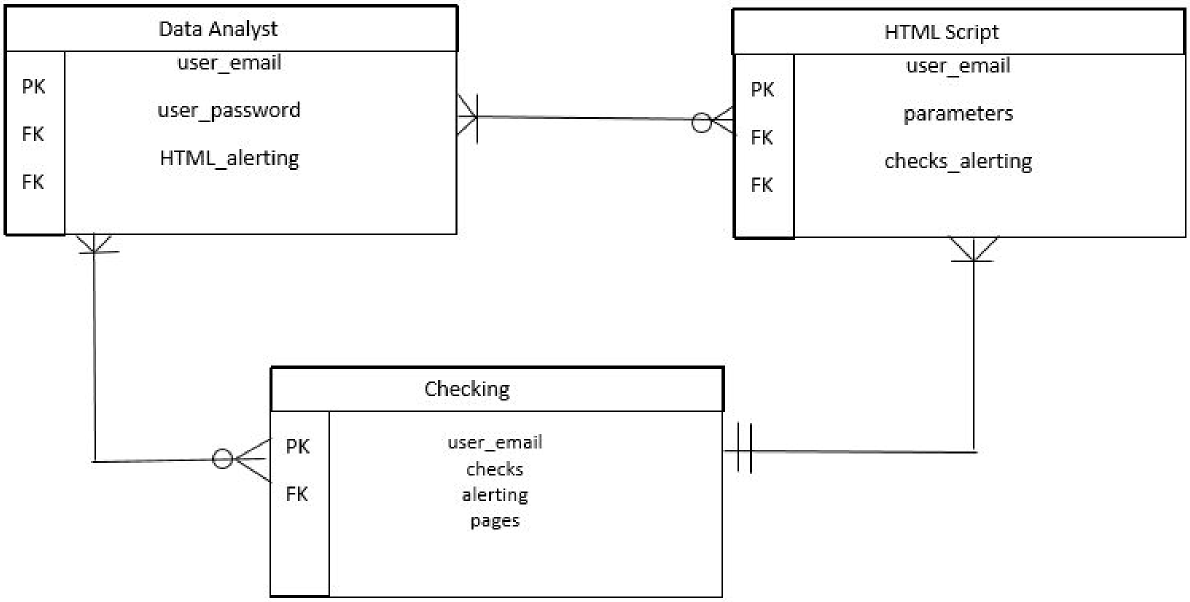


Figure: 5.2.1

The entity relationship diagram shows the cardinality between the individual entities, which in case of this project are Data Analyst, HTML script and Laravel Application. Here we can see that the Data Analyst has zero or many relations with the HTML Script and also zero or many relations with the Laravel Application. This is because it is not mandatory for the Data Analyst to have a script and an application to function. The HTML script has one or many relations with the Data Analyst and one and only relation with the Laravel Application. This is because the script needs at least one Data Analyst to operate it and each script is associated with one and only Laravel Application. The Laravel Application has one or many relations with both Data Analyst and the HTML Script. This is because the application can be used by numerous users and needs to have at least one to monitor it. Similarly, the Laravel Application can have many HTML Scripts associated with it but it needs to have at least one script to function.

5.3 Relation Schema

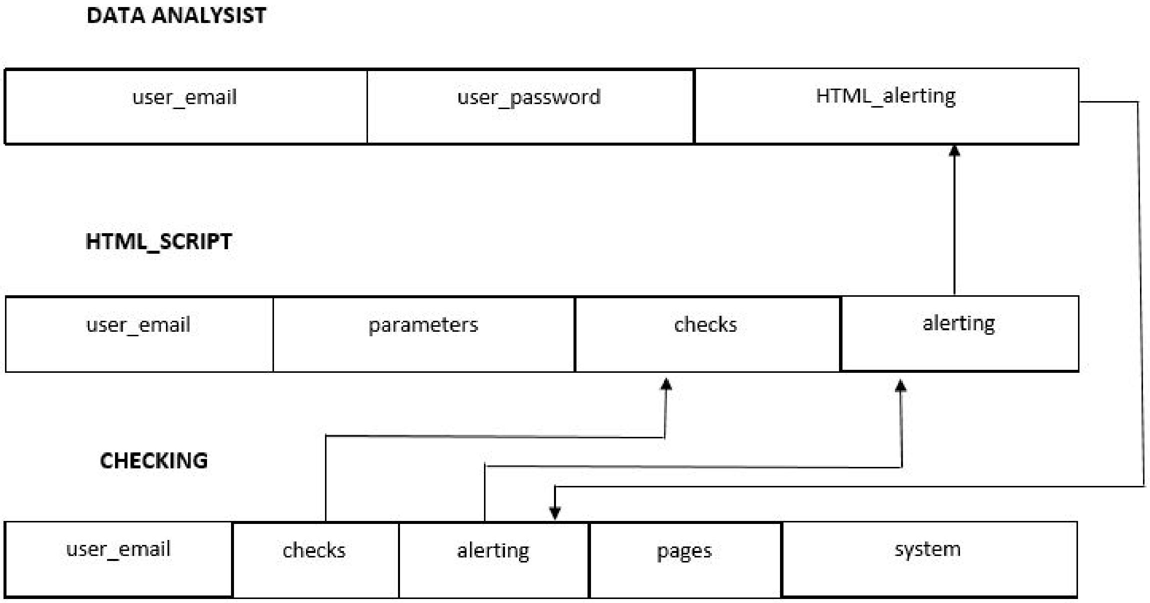


Figure: 5.3.1

In Figure 5.3.1, we can see a relation schema created from the Entity Relationship Diagram. Here we can see the flow of information from table to another, for example, we can see that the columns for laravel_notification and HTML_notification in the data analyst table originate from 39 the tables LARAVEL APPLICATION and HTML APPLICATION respectively. The primary keys are denoted using underlines for each table.

5.4 Normalization

All the entities are in normalized form. Normalization is a systematic way of ensuring that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics like insertion, update, and deletion anomalies that could lead to a loss of data integrity. Normal forms in a database or the concept of Normalization makes a Relation or Table free from insert/update/delete anomalies and saves space by removing duplicate data.

Chapter 6

Implementation

6.1 Modifications to the Existing Scraping Script

Some modifications were needed in order to execute the required functions in the existing HTML scripts, a sample of the code is given above. Please note that this an existing HTML script that was used to scrape data from websites, few lines of code where added to check the integrity of the data that was scraped. The pseudo-code shows the code addition of moving data from one folder to another and then compressing it. It aims to integrate several monitoring solutions to provide a comprehensive cloud solution. It allows cloud administrators to automatically configure monitor-ing solutions, providing flexibility and dynamicity to cloud providers. Flexibility and dynamicity improve operational aspects of cloud providers such as billing and SLA. To reach such automatic configuration, FlexACMS is based on rules defined by cloud administrators that determine the met-rics that must be monitored on each cloud slice and what monitoring solutions must be used to monitor each slice.

6.2 Recommendation

This is a cloud based project, so before using this system user should able to understand some basic things that needed for using this system. This will help to use this system use easier and interactive. So, some recommendations are listed below:

- This is cloud-based project. Before using this application, user should understand using internet.

- User should able to know using browers.

- Before use this sofware, user should check their device internet connections.

- Users should able to know how to login to their account for logging into their account.

- User doesn’t have to be an IT guy but should have basic knowledge about software and it uses.

Chapter 7

Conclusion

The internship has been a very fruitful and worthy experience for me. I was able to work, hands-on, in an industry that I had no prior knowledge about. The process of transforming the rich theoretical knowledge with the practical knowledge of the industry has dawned on me and driven to seek excellence in the craft of software engineering. Interns don’t usually get to work on live projects and contribute to the workflow of an ongoing project in the office. But the people at Accomium , felt that I was worth giving a chance to and tasked me with such projects that would help me grow in every aspect of my career. Being the youngest there and also the least experienced of the bunch, I got a plethora of advice from the people of the offices. I also learned the tools and techniques that were utilized by industry hardened software developers and engineers alike. On top of that I was taught etiquettes of the corporate life and also how to maintain proper rapport with my co-workers. These are the skills that can’t be learned using books and have to be applied to assure proper implementation. It was a blessing for me to be in the presence of such good people who were willing to help me at each part of my journey through the internship. In the end, I’d like to thank both my internal and external supervisors whose guidance and motivations have persuaded me to strive for the success in this project and also for the endless projects to come in my way in the future.

References

1. "Data Storage Corporation | Secure Infrastructure & Software ...." https://www.datastoragecorp.com/. Accessed 26 Jan. 2021.

2. "What is the Cloud? | Cloud Definition | Cloudflare." https://www.cloudflare.com/learning/cloud/what-is-the-cloud/. Accessed 26 Jan. 2021.

3. "Cloud Storage | Google Cloud." https://cloud.google.com/storage. Accessed 26 Jan. 2021.

4. "Physical Data Storage - Finnish Social Science Data Archive ...." https://www.fsd.tuni.fi/aineistonhallinta/en/physical-data-storage.html. Accessed 26 Jan. 2021.

5. "What is monitoring and evaluation?." 31 Oct. 2010, https://www.endvawnow.org/en/articles/330-what-is-monitoring-and-evaluation-.html. Accessed 26 Jan. 2021

6. "What is a Web Application? | How a Web Application Works." 31 May. 2016, https://blog.stackpath.com/web-application/. Accessed 26 Jan. 2021.

7. "How Amazon CloudWatch Works - Amazon CloudWatch." https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch_architecture.html. Accessed 26 Jan. 2021.

8. "AppDynamics - Wikipedia." https://en.wikipedia.org/wiki/AppDynamics. Accessed 26 Jan. 2021.

9. "npm Package Manager." 8 Jan. 2021, https://www.npmjs.com/package/npm. Accessed 26 Jan. 2021.

10. "The Script element - HTML: HyperText Markup Language | MDN." 13 Jan. 2021, https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script. Accessed 26 Jan. 2021.

11. "Laravel - The PHP Framework For Web Artisans." https://laravel.com/. Accessed 26 Jan. 2021

12. "MySQL Data Manipulation and Query Statements." 28 May. 2010, https://www.databasejournal.com/features/mysql/article.php/3884346/MySQL-Data-Manipulation-and-Query-Statements.htm. Accessed 26 Jan. 2021.

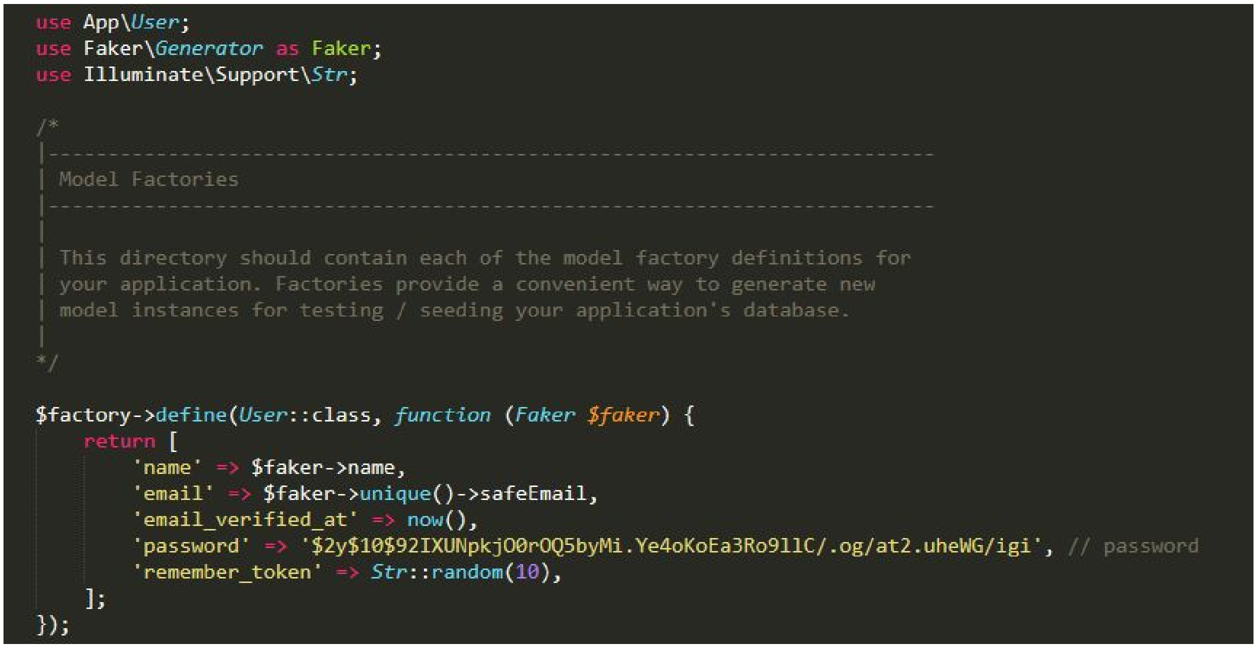
13. "vuejs/vue: Vue.js is a progressive, incrementally ... - GitHub." https://github.com/vuejs/vue. Accessed 26 Jan. 2021.

Appendix

Coding Snippets

HTML Script For Copy And Compression for User

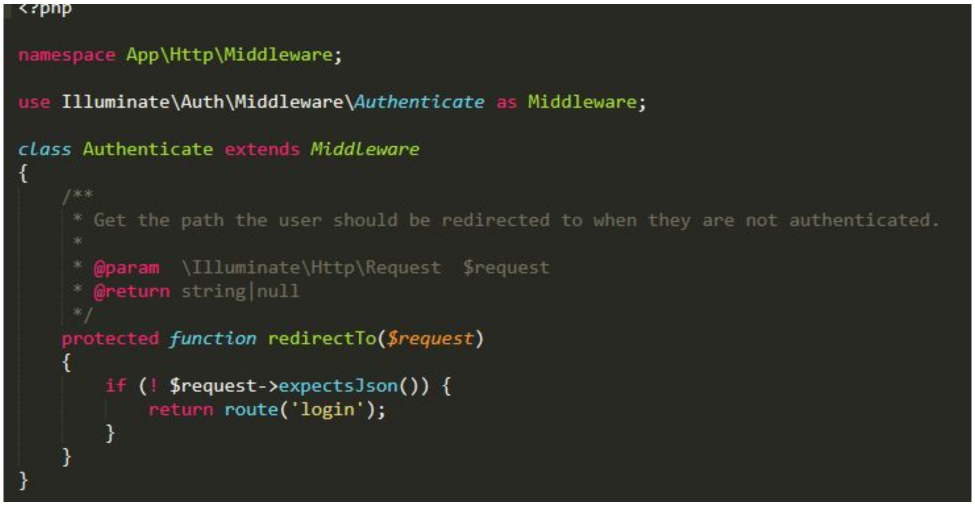
For User



```
use App\User;
use Faker\Generator as Faker;
use Illuminate\Support\Str;

/*
|--------------------------------------------------------------------------
| Model Factories
|--------------------------------------------------------------------------
|
| This directory should contain each of the model factory definitions for
| your application. Factories provide a convenient way to generate new
| model instances for testing / seeding your application's database.
|
*/

$factory->define(User::class, function (Faker $faker) {
    return [
        'name' => $faker->name,
        'email' => $faker->unique()->safeEmail,
        'email_verified_at' => now(),
        'password' => '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password
        'remember_token' => Str::random(10),
    ];
});
```

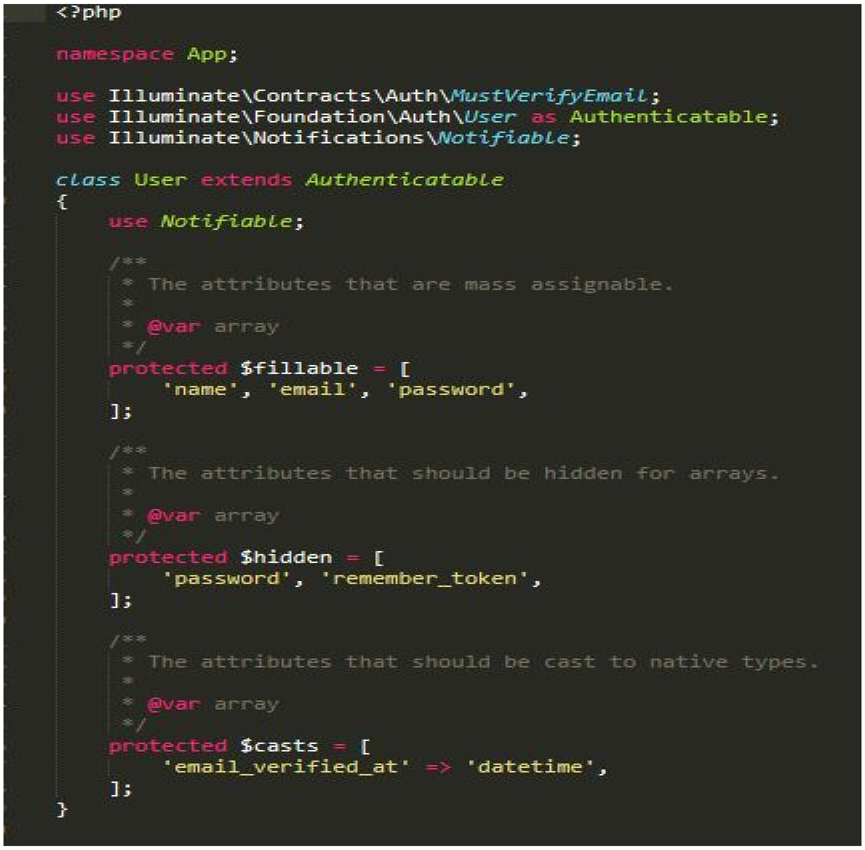
Figure: A.1

For Authentication



```php
<?php

namespace App\Http\Middleware;

use Illuminate\Auth\Middleware\Authenticate as Middleware;

class Authenticate extends Middleware
{
    /**
     * Get the path the user should be redirected to when they are not authenticated.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return string|null
     */
    protected function redirectTo($request)
    {
        if (! $request->expectsJson()) {
            return route('login');
        }
    }
}
```

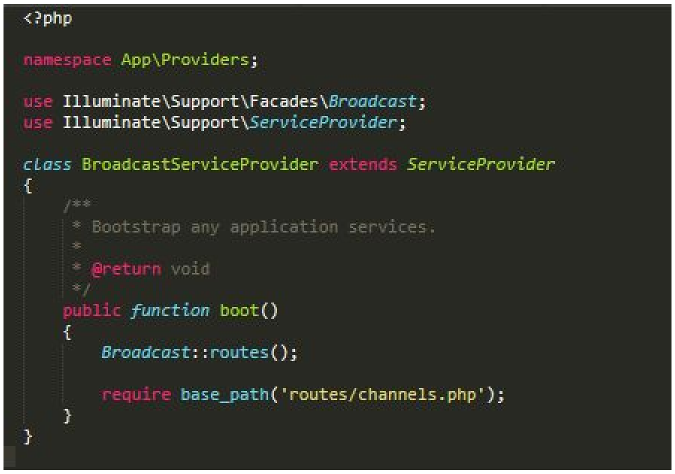
Figure: A.2.1



```php
<?php

namespace App;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

Figure: A.2.2

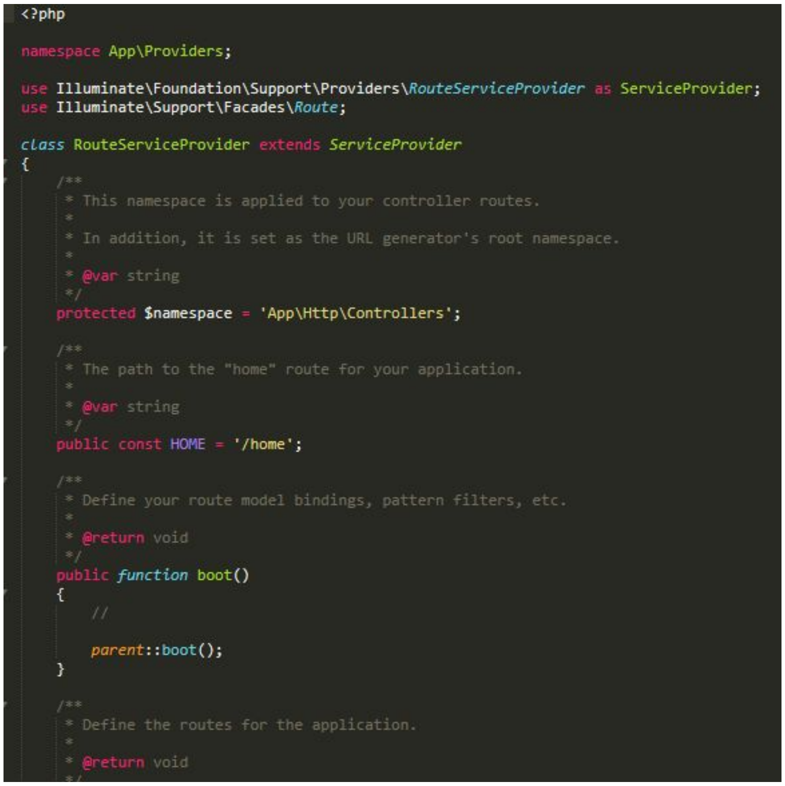
For broadcasting



```php
<?php

namespace App\Providers;

use Illuminate\Support\Facades\Broadcast;
use Illuminate\Support\ServiceProvider;

class BroadcastServiceProvider extends ServiceProvider
{
    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {
        Broadcast::routes();

        require base_path('routes/channels.php');
    }
}
```

Figure: A.4

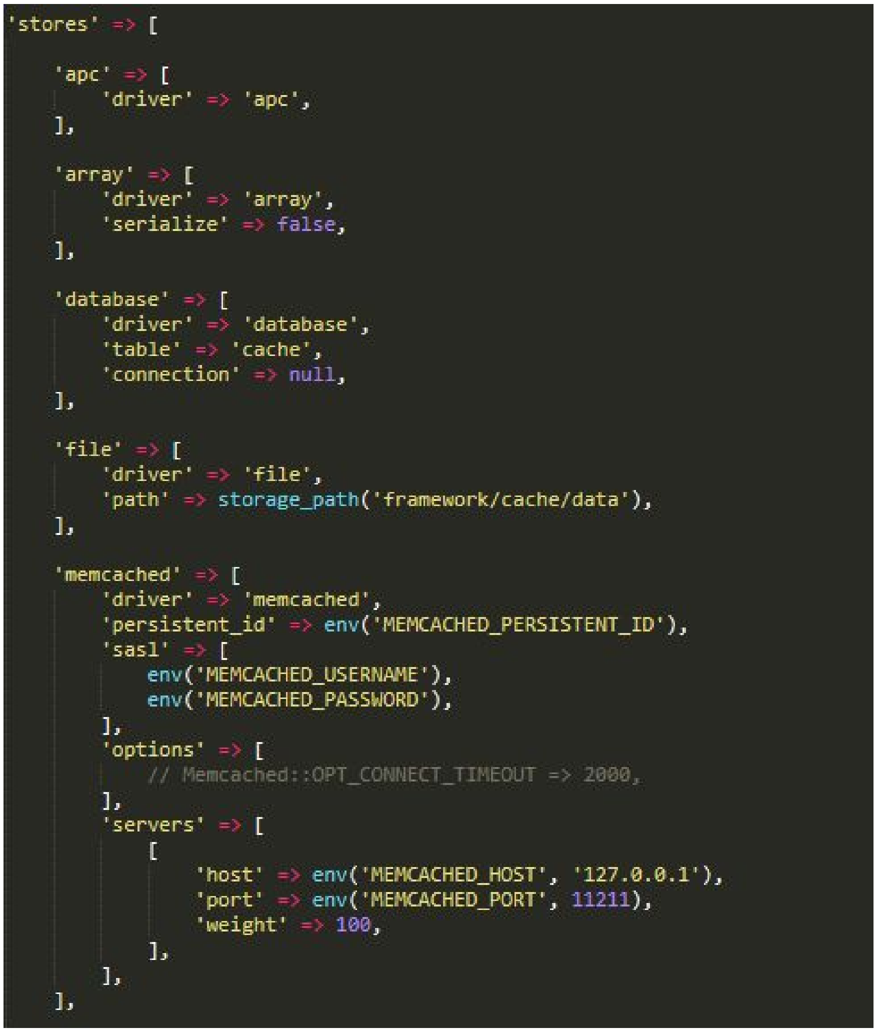
For service provider



```php
<?php

namespace App\Providers;

use Illuminate\Foundation\Support\Providers\RouteServiceProvider as ServiceProvider;
use Illuminate\Support\Facades\Route;

class RouteServiceProvider extends ServiceProvider
{
    /**
     * This namespace is applied to your controller routes.
     *
     * In addition, it is set as the URL generator's root namespace.
     *
     * @var string
     */
    protected $namespace = 'App\Http\Controllers';

    /**
     * The path to the "home" route for your application.
     *
     * @var string
     */
    public const HOME = '/home';

    /**
     * Define your route model bindings, pattern filters, etc.
     *
     * @return void
     */
    public function boot()
    {
        //

        parent::boot();
    }

    /**
     * Define the routes for the application.
     *
     * @return void
     */
```

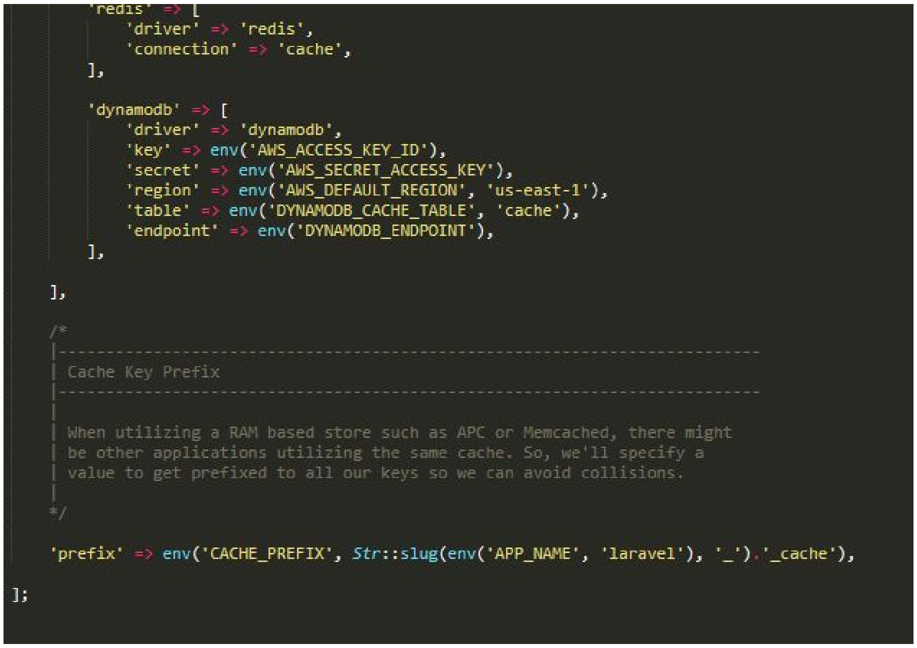
Figure: A.5

For storage data



```
'stores' => [

    'apc' => [
        'driver' => 'apc',
    ],

    'array' => [
        'driver' => 'array',
        'serialize' => false,
    ],

    'database' => [
        'driver' => 'database',
        'table' => 'cache',
        'connection' => null,
    ],

    'file' => [
        'driver' => 'file',
        'path' => storage_path('framework/cache/data'),
    ],

    'memcached' => [
        'driver' => 'memcached',
        'persistent_id' => env('MEMCACHED_PERSISTENT_ID'),
        'sasl' => [
            env('MEMCACHED_USERNAME'),
            env('MEMCACHED_PASSWORD'),
        ],
        'options' => [
            // Memcached::OPT_CONNECT_TIMEOUT => 2000,
        ],
        'servers' => [
            [
                'host' => env('MEMCACHED_HOST', '127.0.0.1'),
                'port' => env('MEMCACHED_PORT', 11211),
                'weight' => 100,
            ],
        ],
    ],
```

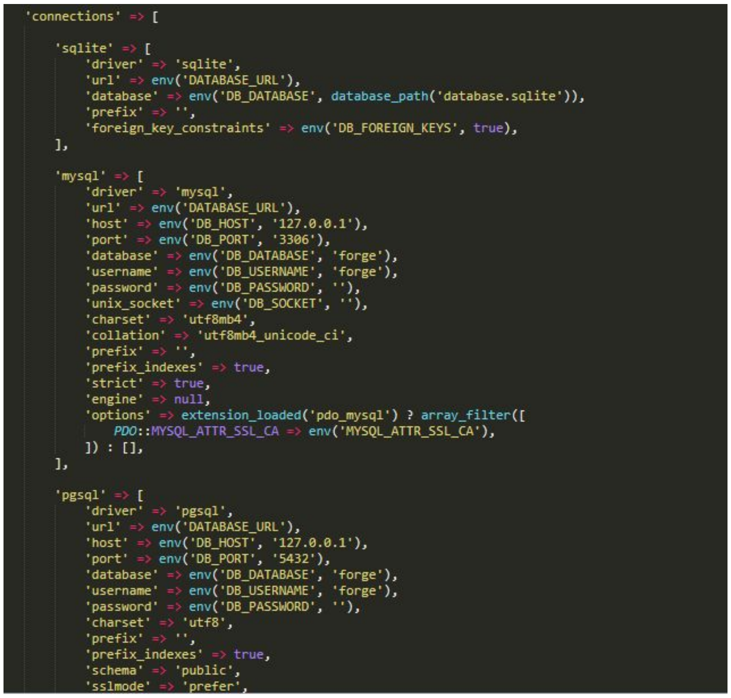
Figure: A.6.1



```
        'redis' => [
            'driver' => 'redis',
            'connection' => 'cache',
        ],

        'dynamodb' => [
            'driver' => 'dynamodb',
            'key' => env('AWS_ACCESS_KEY_ID'),
            'secret' => env('AWS_SECRET_ACCESS_KEY'),
            'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
            'table' => env('DYNAMODB_CACHE_TABLE', 'cache'),
            'endpoint' => env('DYNAMODB_ENDPOINT'),
        ],

    ],

    /*
    |--------------------------------------------------------------------------
    | Cache Key Prefix
    |--------------------------------------------------------------------------
    |
    | When utilizing a RAM based store such as APC or Memcached, there might
    | be other applications utilizing the same cache. So, we'll specify a
    | value to get prefixed to all our keys so we can avoid collisions.
    |
    */

    'prefix' => env('CACHE_PREFIX', Str::slug(env('APP_NAME', 'laravel'), '_').'_cache'),

];
```

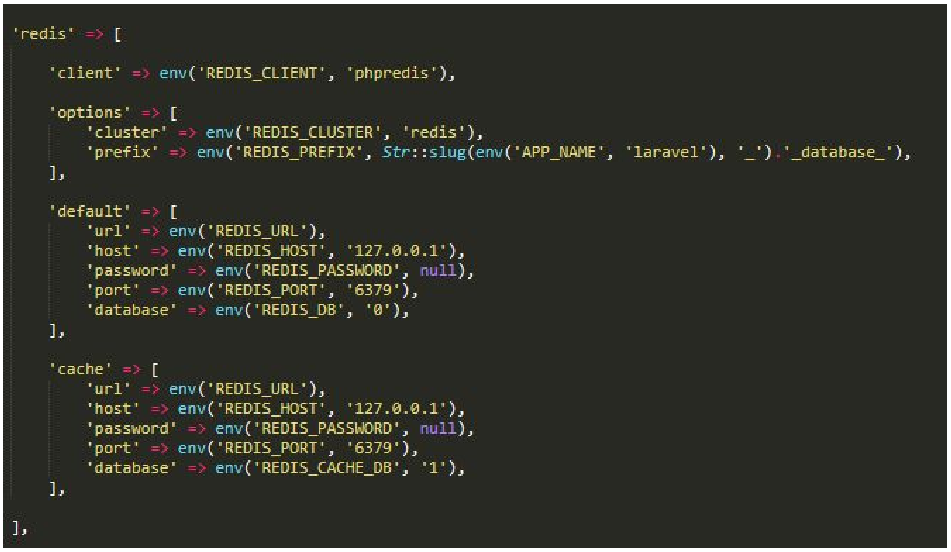
Figure: A.6.2

For Database



```
    'connections' => [

        'sqlite' => [
            'driver' => 'sqlite',
            'url' => env('DATABASE_URL'),
            'database' => env('DB_DATABASE', database_path('database.sqlite')),
            'prefix' => '',
            'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),
        ],

        'mysql' => [
            'driver' => 'mysql',
            'url' => env('DATABASE_URL'),
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '3306'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
            'prefix' => '',
            'prefix_indexes' => true,
            'strict' => true,
            'engine' => null,
            'options' => extension_loaded('pdo_mysql') ? array_filter([
                PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
            ]) : [],
        ],

        'pgsql' => [
            'driver' => 'pgsql',
            'url' => env('DATABASE_URL'),
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '5432'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'charset' => 'utf8',
            'prefix' => '',
            'prefix_indexes' => true,
            'schema' => 'public',
            'sslmode' => 'prefer',
```

Figure: A.7.1



```
'redis' => [

    'client' => env('REDIS_CLIENT', 'phpredis'),

    'options' => [
        'cluster' => env('REDIS_CLUSTER', 'redis'),
        'prefix' => env('REDIS_PREFIX', Str::slug(env('APP_NAME', 'laravel'), '_').'_database_'),
    ],

    'default' => [
        'url' => env('REDIS_URL'),
        'host' => env('REDIS_HOST', '127.0.0.1'),
        'password' => env('REDIS_PASSWORD', null),
        'port' => env('REDIS_PORT', '6379'),
        'database' => env('REDIS_DB', '0'),
    ],

    'cache' => [
        'url' => env('REDIS_URL'),
        'host' => env('REDIS_HOST', '127.0.0.1'),
        'password' => env('REDIS_PASSWORD', null),
        'port' => env('REDIS_PORT', '6379'),
        'database' => env('REDIS_CACHE_DB', '1'),
    ],

],
```

Figure: A.7.2