**Aryan School of Engineering**

**Department of Information and Technology**

AFFILIATED TO: PURBANCHAL UNIVERSITY



**Minor Project Proposal on:**

**COLLEGE APP**

**[Code No: BIT279CO]**

**By**

**Anup Karki- Symbol No: xxxxxx**

**Jit Bdr Rana- Symbol No:xxxxxx**

**Pratik Shrestha- Symbol No: xxxxxx**

**Seazone Joshi- Symbol No: xxxxxx**

**Kathmandu, Nepal**

**2021**

**Aryan School of Engineering**

**Department of Information and Technology**

AFFILIATED TO: PURBANCHAL UNIVERSITY

**COLLEGE APP**

**[Code No: BIT379CO]**

**PROJECT PROPOSAL SUBMITTED TO THE DEPARTMENT OF INFORMATION TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF INFORMATION TECHNOLOGY**



**By**

**Anup Karki- Symbol No: xxxxxx**

**Jit Bdr Rana- Symbol No:xxxxxx**

**Pratik Shrestha- Symbol No: xxxxxx**

**Seazone Joshi- Symbol No: xxxxxx**

**Kathmandu, Nepal**

**Jan 2021**

**PURBANCHAL UNIVERSITY**

**Aryan school of Engineering**

**Department of Information and Technology**

**Aryan College of Engineering**

**Department of Information Technology**

**Affiliated To: Purbanchal University**

**CERTIFICATE**

The undersigned certify that they have read and recommended to the Department of Information Technology, a minor project work entitled "COLLEGE APP" submitted by Anup Karki (xxxxxx) Jit Bdr Rana (xxxxxx) Pratik Shrestha (xxxxxx) and Seazone Joshi (xxxxxx) in partial fulfillment of the requirements for the degree of Bachelor of IT.

(Head of Department)

Department of Electronics and

Computer Engineering.

Aryan School of Engineering

(Project Coordinate)

Department of Electronics and

Computer Engineering.

Aryan School of Engineering

**ACKNOWLEDGEMENT**

We would like to extend our heartiest thanks with a deep sense of gratitude and respect to all those who provides me immense help and guidance during my training period. We have been greatly benefited from their regular critical reviews and inspiration throughout my work.

We would also like to thank my project supervisor **Mr.sample and Er. sample** for his unfailing cooperation and sparing his valuable time to assist us in our work for preparing proposal of the project.

We would like to express our sincere thanks to our Head of Department **Er. Nisha Karki** andour internal guide **Mr. Sample Sample,** who gave us an opportunity to undertake such a great challenging and innovative work. We are grateful to them for their guidance, encouragement, understanding and insightful support in the development process.

Last but not the least we would like to mention here that we greatly indebted to each and everybody who has been associated with our project at any stage but whose name does not find a place in this acknowledgement.

**ABSTRACT**

**List of Figure**

**TABLE OF CONTENT**

[CHAPTER 1: INTRODUCTION 1](#_Toc10315590)

[4.1 BACKGROUND 2](#_Toc10315591)

[4.2 PROBLEM OF STATEMENT 3](#_Toc10315592)

[4.3 SCOPE 4](#_Toc10315593)

[4.4 PROJECT FEATURES 5](#_Toc10315594)

[4.5 OBJECTIVE 6](#_Toc10315595)

[4.6 SYSTEM REQUIREMENT 7](#_Toc10315596)

[1.6.1 Software Requirement (while developing) 7](#_Toc10315597)

[1.6.2 Hardware Requirement (while developing) 7](#_Toc10315598)

[CHAPTER 2:LITERATURE REVIEW 8](#_Toc10315599)

[2.1 BACKGROUND 9](#_Toc10315600)

[CHAPTER 3:METHODOLOGY 11](#_Toc10315601)

[3.1 FEABILITY STUDY 12](#_Toc10315602)

[3.1.1 Operational Feasibility 12](#_Toc10315603)

[3.1.2 Technical Feasibility 12](#_Toc10315604)

[3.1.3 Economical Feasibility 13](#_Toc10315605)

[3.2 BEHAVIOUR MODELING 14](#_Toc10315606)

[3.2.1 FLOW CHART 14](#_Toc10315607)

[3.2.2 DATA FLOW DIAGRAM 15](#_Toc10315608)

[3.2.3 ER DIAGRAM 16](#_Toc10315609)

[3.2.4 USE CASE DIAGRAM 17](#_Toc10315610)

[3.2.5 STATE DIAGRAM 18](#_Toc10315611)

[CHAPTER 4:SYSTEM TESTING 20](#_Toc10315612)

[4.1 COST ESTIMATION 21](#_Toc10315613)

[4.2 GANTT CHART 22](#_Toc10315614)

[CHAPTER 5:EXPECTED OUTPUT 23](#_Toc10315615)

[5.1 SCREEN SHOTS 24](#_Toc10315616)

[CHAPTER 6:CONCLUSION 25](#_Toc10315617)

[6.1 CONCLUSION 26](#_Toc10315618)

[REFRENCE 28](#_Toc10315619)

# 

# CHAPTER 1: INTRODUCTION

## BACKGROUND

## PROBLEM OF STATEMENT

## SCOPE

## PROJECT FEATURES

## OBJECTIVE

## SYSTEM REQUIREMENT

### 

### Software Requirement (while developing)

* Windows 10 or Linux
* Programming languages: Html, CSS, JavaScript, PHP
* Framework: Laravel
* Text editor: Sublime Text/VS Code

### Hardware Requirement (while developing)

The following is a list of computer hardware specifications that are suggested by the developer as the minimum requirements for a computer to efficiently run the system.

* Processor: Pentium IV or higher
* Memory: 2 GB minimum
* Hard drive: 40 GB free space
* 1024 \* 768 Resolution Color Monitor

# CHAPTER 2:LITERATURE REVIEW

## BACKGROUND

An Internet bot, web robot, robot or simply bot, is a software application that runs automated tasks (scripts) over the Internet.[1] Typically, bots perform tasks that are simple and repetitive, much faster than a person could. The most extensive use of bots is for web crawling, in which an automated script fetches, analyzes and files information from web servers. More than half of all web traffic is generated by bots.[2] Some bots communicate with users of Internet-based services, via Instant Messaging (IM), Internet Relay Chat (IRC), or other web interfaces such as Facebook Bots and Twitter Bots. These chat bots may allow people to ask questions in plain English and then formulate a response. Such bots can often handle reporting weather, zip code information, sports scores, currency or other unit conversion, etc.

Malicious use of bots is the coordination and operation of an automated attack on networked computers, such as a denial-of-service attack by a botnet. Internet bots or web bots can also be used to commit click fraud and more recently have appeared around MMORPG games, as computer game bots. Another category is represented by Spam bots, internet bots that attempt to spam large amounts of content on the Internet, usually adding advertising links. More than 94.2% of websites have experienced a bot attack.[2]

There are malicious bots (and botnets) of the following types:

* Spambots that harvest email addresses from contact or guestbook pages
* Downloaded programs that suck bandwidth by downloading entire websites
* Website scrapers that grab the content of websites and re-use it without permission on automatically generated doorway pages
* Registration bots which sign up a specific email address to numerous services in order to have the confirmation messages flood the email inbox and distract from important messages indicating a security breach.[3]
* Viruses and worms
* DDoS attacks
* Botnets, zombie computers, etc.
* Spambots that try to redirect people onto a malicious website, sometimes found in comment sections or forums of various websites.
* Viewbots create fake views[4][5]

A botnet is a number of Internet-connected devices, each of which is running one or more bots. Botnets can be used to perform Distributed Denial-of-Service (DDoS) attacks, steal data,[6] send spam, and allow the attacker to access the device and its connection. The owner can control the botnet using command and control (C&C) software. The word "botnet" is a portmanteau of the words "robot" and "network". The term is usually used with a negative or malicious connotation. The controller of a botnet is able to direct the activities of these compromised computers through communication channels formed by standards-based network protocols, such as IRC and Hypertext Transfer Protocol (HTTP).[7] Botnet architecture has evolved over time in an effort to evade detection and disruption. Traditionally, bot programs are constructed as clients which communicate via existing servers. This allows the bot herder (the person controlling the botnet) to perform all control from a remote location, which obfuscates the traffic.[8] Many recent botnets now rely on existing peer-to-peer networks to communicate. These P2P bot programs perform the same actions as the client-server model, but they do not require a central server to communicate.

A denial-of-service attack (DoS attack) is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.[9] In a distributed denial-of-service attack (DDoS attack), the incoming traffic flooding the victim originates from many different sources. This effectively makes it impossible to stop the attack simply by blocking a single source. A DoS or DDoS attack is analogous to a group of people crowding the entry door of a shop, making it hard for legitimate customers to enter, thus disrupting trade. Criminal perpetrators of DoS attacks often target sites or services hosted on high-profile web servers such as banks or credit card payment gateways. Revenge, blackmail and activism can motivate these attacks.

Common DDoS attacks types:

Some of the most commonly used DDoS attack types include:

* UDP Flood

A UDP flood, by definition, is any DDoS attack that floods a target with User Datagram Protocol (UDP) packets.[10] The goal of the attack is to flood random ports on a remote host. This causes the host to repeatedly check for the application listening at that port, and (when no application is found) reply with an ICMP ‘Destination Unreachable’ packet. This process saps host resources, which can ultimately lead to inaccessibility.

* ICMP (Ping) Flood

Similar in principle to the UDP flood attack, an ICMP flood overwhelms the target resource with ICMP Echo Request (ping) packets, generally sending packets as fast as possible without waiting for replies. This type of attack can consume both outgoing and incoming bandwidth, since the victim’s servers will often attempt to respond with ICMP Echo Reply packets, resulting a significant overall system slowdown.

* SYN Flood

A SYN flood DDoS attack exploits a known weakness in the TCP connection sequence (the “three-way handshake”), wherein a SYN request to initiate a TCP connection with a host must be answered by a SYN-ACK response from that host, and then confirmed by an ACK response from the requester.[11] In a SYN flood scenario, the requester sends multiple SYN requests, but either does not respond to the host’s SYN-ACK response, or sends the SYN requests from a spoofed IP address. Either way, the host system continues to wait for acknowledgement for each of the requests, binding resources until no new connections can be made, and ultimately resulting in denial of service.

* Ping of Death

A ping of death (“POD”) attack involves the attacker sending multiple malformed or malicious pings to a computer. The maximum packet length of an IP packet (including header) is 65,535 bytes. However, the Data Link Layer usually poses limits to the maximum frame size – for example 1500 bytes over an Ethernet network. In this case, a large IP packet is split across multiple IP packets (known as fragments), and the recipient host reassembles the IP fragments into the complete packet. In a Ping of Death scenario, following malicious manipulation of fragment content, the recipient ends up with an IP packet which is larger than 65,535 bytes when reassembled. This can overflow memory buffers allocated for the packet, causing denial of service for legitimate packets.

* Slowloris

Slowloris is a highly-targeted attack, enabling one web server to take down another server, without affecting other services or ports on the target network. Slowloris does this by holding as many connections to the target web server open for as long as possible. It accomplishes this by creating connections to the target server, but sending only a partial request. Slowloris constantly sends more HTTP headers, but never completes a request. The targeted server keeps each of these false connections open. This eventually overflows the maximum concurrent connection pool, and leads to denial of additional connections from legitimate clients.[12]

* NTP Amplification

In NTP amplification attacks, the perpetrator exploits publically-accessible Network Time Protocol (NTP) servers to overwhelm a targeted server with UDP traffic. The attack is defined as an amplification assault because the query-to-response ratio in such scenarios is anywhere between 1:20 and 1:200 or more. This means that any attacker that obtains a list of open NTP servers (e.g., by a using tool like Metasploit or data from the Open NTP Project) can easily generate a devastating high-bandwidth, high-volume DDoS attack.

* HTTP Flood

In an HTTP flood DDoS attack, the attacker exploits seemingly-legitimate HTTP GET or POST requests to attack a web server or application. [13]HTTP floods do not use malformed packets, spoofing or reflection techniques, and require less bandwidth than other attacks to bring down the targeted site or server. The attack is most effective when it forces the server or application to allocate the maximum resources possible in response to every single request.

# CHAPTER 3:METHODOLOGY

## FEASIBILITY STUDY

Early studies have been made on this, as it is a rising topic. Some research have been done before setting goals or objectives for the project and initializing the project. It has been known from the research that there had been earlier attempts for creating such products and there are some products that meets certain portion of the public and market demands. But from deep research we have come to know that there has not been any kind of complete product relating to this topic so we decided to have an attempt on creating more facilitated and complete product in comparison to the earlier products that already exist in the market.

Thus, after long term research and discussion among the group members we have decided to make a proposal for the project defining the researches and objectives we have set for the project.

There are three tests of Feasibility Study:-

* Operational
* Technical
* Economical /financial

### Operational Feasibility

This test of feasibility asks if the system will work when it is developed and installed. This feasibility observes the all operations like finding products, gathering information, getting supports & software & more.

### Technical Feasibility

This involves the technological consideration like available technology to run the purposed system. The system has been created with the most common technology available. Hence the system is technically feasible.

### Economical Feasibility

Cost of the system will be affordable for the user. As it is a simple product financial consideration was not a big deal for our project.

## BEHAVIOUR MODELING

### FLOW CHART

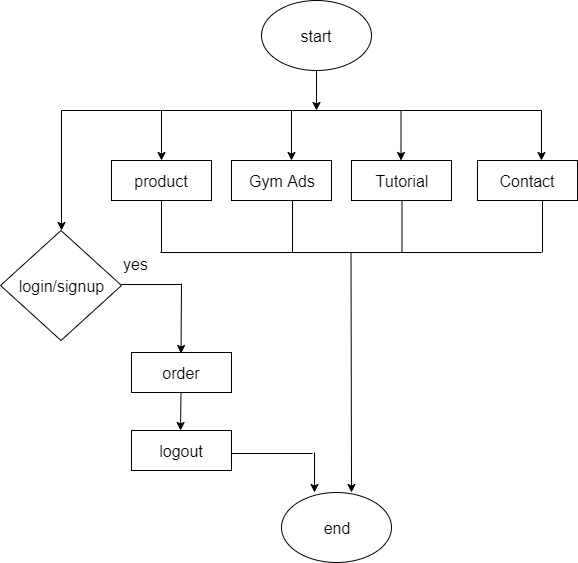


Figure 2:FLOW CHART

### DATA FLOW DIAGRAM

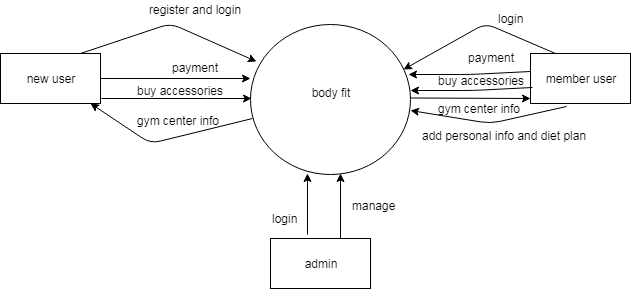


Figure 3:CONTEXT DIAGRAM

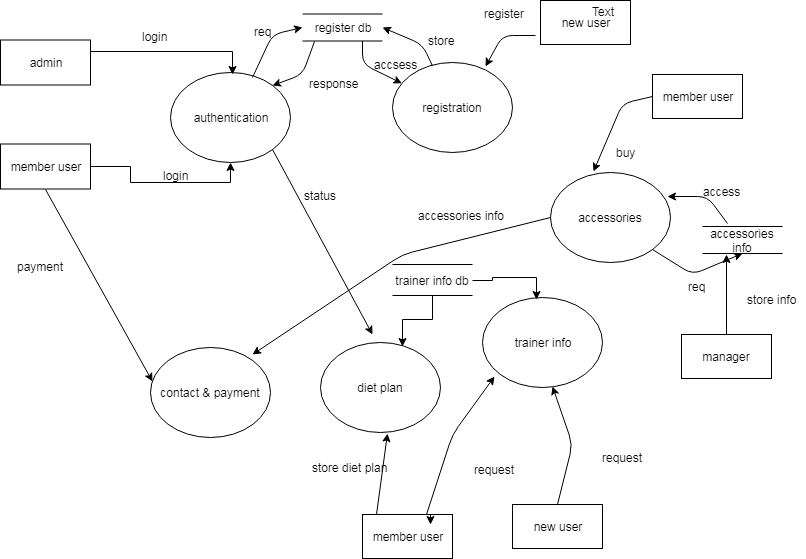


Figure 4:DFD LEVEL 1

### ER DIAGRAM

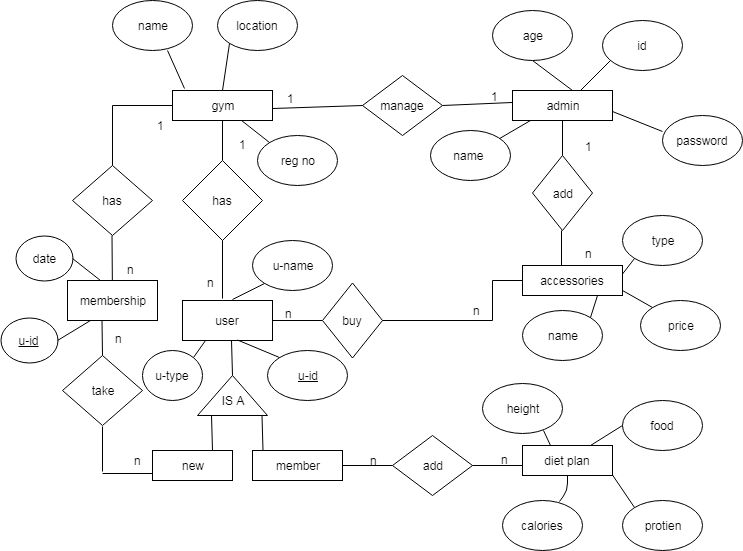


Figure 5:ER DIAGRAGM

### USE CASE DIAGRAM

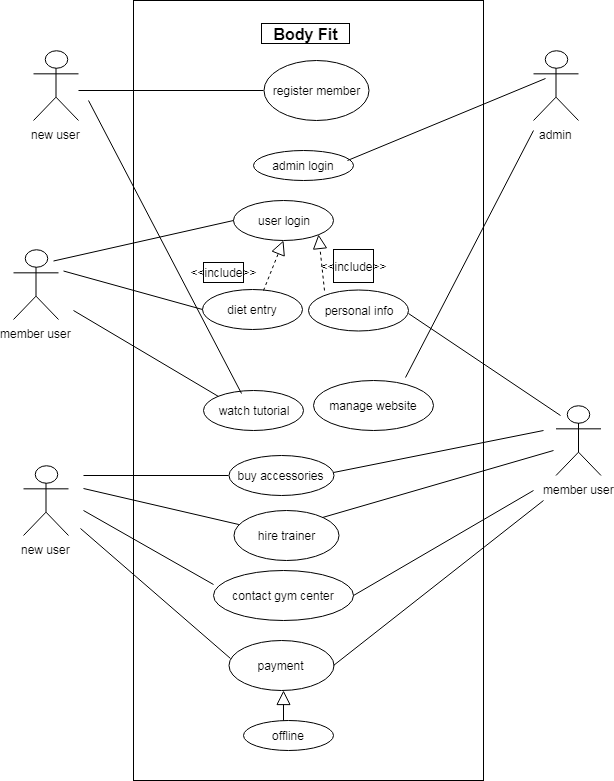


Figure 6:CLASS DIAGRAM

### STATE DIAGRAM

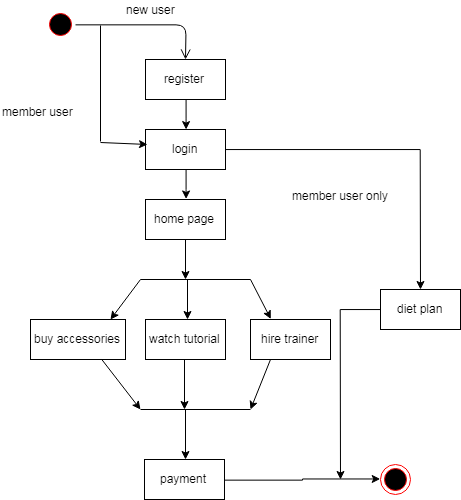


Figure 7:STATE DIAGRAM

# CHAPTER 4:SYSTEM TESTING

## COST ESTIMATION

These documents will provide the necessary data to feed into the budget line items. The accuracy of the budgets requires that attention be given both to individual expense per item. The budget goal, is to the extent possible, directly map the project expense to actual costs within 10 per-cent difference if there be any disparity.

|  |  |  |  |
| --- | --- | --- | --- |
| **ITEM** | **PRICE**  **(Rs)** | **UNITS** | **TOTAL**  **(Rs)** |
| **Backup device (Portable hard disk)** | 800 | 1 | **800** |
| **Modem** | 3,000 | 1 | **3,000** |
| **Antivirus software** | 3,000 | 1 | **3,000** |
| **Printing Costs** | 5 per page | Over 25 pages | **125** |
| **Miscellaneous** | - | - | **1,000** |
| **Total** | - | - | **7,225** |

Figure 8:COST ESTIMATION

## GANTT CHART

Jan Feb Mar May June

Figure 9:GANTT CHART

# CHAPTER 5:OUTPUT

## SCREEN SHOTS

# CHAPTER 6:CONCLUSION

## CONCLUSION

# REFRENCE

* [1] Dunham, Ken; Melnick, Jim (2009). [*Malicious Bots: An outside look of the Internet*](https://archive.org/details/malicio_dun_2009_00_4004) .
* [2] Zeifman, Igal. ["Bot Traffic Report 2016"](https://www.incapsula.com/blog/bot-traffic-report-2016.html). *Incapsula*. Retrieved 1 February 2017.
* [3] Dima Bekerman: [How Registration Bots Concealed the Hacking of My Amazon Account](https://www.imperva.com/blog/amazon-account-hack-registration-bots/), Application Security, Industry Perspective, December 1st 2016,
* [4] Carr, Sam (July 15, 2019). ["What Is Viewbotting: How Twitch Are Taking On The Ad Fraudsters"](https://ppcprotect.com/what-is-viewbotting/).
* [5] Lewis, Richard (March 17, 2015). ["Leading StarCraft streamer embroiled in viewbot controversy"](https://dotesports.com/general/news/starcraft-streamer-viewbot-winter-ballnick-1618).
* [6] ["Thingbots: The Future of Botnets in the Internet of Things"](https://securityintelligence.com/thingbots-the-future-of-botnets-in-the-internet-of-things/). *Security Intelligence*. 20 February 2016. Retrieved 28 July 2017.
* [7] Ramneek, Puri (8 August 2003). ["Bots &; Botnet: An Overview"](http://www.sans.org/reading-room/whitepapers/malicious/bots-botnet-overview-1299). [SANS Institute](https://en.wikipedia.org/wiki/SANS_Institute).
* [8] Schiller, Craig A.; Binkley, Jim; Harley, David; Evron, Gadi; Bradley, Tony; Willems, Carsten; Cross, Michael (1 January 2007). *Botnets*. Burlington: Syngress. pp. 29–75.
* [9] ["Understanding Denial-of-Service Attacks"](https://www.us-cert.gov/ncas/tips/ST04-015). US-CERT. 6 February 2013. Retrieved 26 May 2016.
* [10] ["CA-1996-01: UDP Port Denial-of-Service Attack"](https://resources.sei.cmu.edu/asset_files/WhitePaper/1996_019_001_496172.pdf#page=5) (PDF). *Carnegie Mellon University Software Engineering Institute*. [Archived](https://web.archive.org/web/20010124064800/http:/www.cert.org/advisories/CA-1996-01.html) from the original on 2001-01-24.
* [11] [New York's Panix Service Is Crippled by Hacker Attack](https://partners.nytimes.com/library/cyber/week/0914panix.html), New York Times, September 14, 1996
* [12] ["Slowloris HTTP DoS"](https://web.archive.org/web/20150426090206/http:/ha.ckers.org/slowloris). Archived from the original on 26 April 2015. Retrieved 26 June 2009.
* [13] Daniel (February 6, 2014). ["Layer 7 DDOS – Blocking HTTP Flood Attacks"](https://blog.sucuri.net/2014/02/layer-7-ddos-blocking-http-flood-attacks.html).