**Pictofy**

***Dissertation Submitted in Partial fulfillment of the***

***Requirement for the Award of the Degree of***

***Master of Computer Applications***

***Semester VI***

***Jan-May 2019***

**Under the guidance of: Submitted By:**

**Mrs. Poonam Mangwani Harsh Rathore (IC-2K16-20)**

**Neha Thakur(IC-2K16-25)**

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**DECLARATION**

I hereby declare that the project entitles “**Pictofy”,** submitted by us for the partial fulfillment of the requirement for the award of Master of Computer Application Technology (6 Years) Semester VI to International Institute of Professional Studies, Devi Ahilya Vishwavidyalaya, Indore, comprises of our own work that we learned in school and taking part in different external competitions and due acknowledgement has been made in the text.

Signature of Student: Harsh Rathore Neha Thakur

IC-2K16-20 IC-2K16-25

Date: May 9, 2019

Place: Indore (M.P)

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**CERTIFICATE FROM GUIDE**

It is to certify that dissertation on “Pictofy”, submitted by Ms. Neha Thakur and Mr. Harsh Rathore to the International Institute of Professional Studies, DAVV, Indore has been completed under my supervision and the work is carried out and presented in a manner required for its acceptance in partial fulfillment for the award of the degree of “Master of Computer Application (6 Years) Semester XII”.

**Project Supervisor: Mrs. Poonam Mangwani**

Signature:

**Name:**

**Date:**

**International Institute of Professional Studies**

**Devi Ahilya Vishwavidyalaya, Indore, M.P.**

**CERTIFICATE**

This is to certify that we have examined the dissertation on this project report for **PICTOFY** submitted by Harsh Rathore (IC-2K16-20) and Neha Thakur (IC-2K16-25) to the International Institute of Professional Studies, DAVV, Indore and hereby accord our approval of it as a study that they carried out and presented in a manner required for its acceptance in partial fulfillment for the award of the degree of “Masters of Computer Application (6yrs) Semester VI”.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**Signature: Signature:**

**Name:** **Name:**

**Date: Date:**

**Acknowledgement**

A lot of time and efforts have been put into the project. All these efforts have been a result of guidance and inspiration of many people around us. So we would like to take our time and thank some of them.

I sincerely thank IIPS, DAVV to provide us with the valuable support through the computer labs and the internet provided, which has helped us throughout and given us a place to work.

I would like to acknowledge the fact that we have been sincerely helped and guided throughout by our project incharge, Mrs. Poonam Mangwani. In her guidance it is that we have been able to work on this project.

I also thank Dr. A. K. Sapre, Director IIPS, DAVV, Dr. Ramesh Thakur, Program In-charge MCA course and, Mr. Jugendra Dongre, Program In-charge MCA course who helped very willingly. Our friends also deserve a mention for their constant review, praise and critical appraisal. I thank and acknowledge each and everyone's efforts that helped me in some or the other way for small and significant things.

Before ending I would like to thank my parents, and friends who have always stood by us whenever I was in need of them. I sincerely regret any inadvertent omissions.

Thanking all.

Harsh Rathore (IC-2K16-20)

Neha Thakur (IC-2K16-25)

**Table of Contents**

*Ch. Title Page No*

Declaration i

Certificate From Guide ii

Certificate iii

Acknowledgement vi

1. Introduction 1
2. Language and Tools to be Used 2

Chapter-1

Introduction

* 1. **Introduction**

This document provides an overview of the project, “**Pictofy**”. This document explains the architecture of the project and puts out the goals of **Pictofy.** The documentprovides a high –level description of goals of an image sharing platform – Pictofy. It also provides the list of hardware and software requirements.

* 1. **Purpose**

This project’s purpose is to create an image sharing platform with an easy to use interface and managed photo albums. It focuses on easy and efficient sharing of photographs with friends and closed ones.

* 1. **Scope**

Pictofy handles all the operations like creating profile, uploading images, adding captions, liking and commenting on the pictures.

Allows user to view profile of the people they are following.

1.4 **Project Objectives**

Pictofy allows a user to:

✓ Create Profile.

✓ Upload Profile Picture.

✓ Add bio to user’s profile.

✓ Upload Photos.

✓ Comment on the photos of the people you’re following.

✓ Like the pictures uploaded by the people you’re following.

✓ Add captions to your photos.

✓ Search for people.

* 1. **Features of Pictofy**

**Pictofy** can be used to upload, like and comment on a photo uploaded by another user and follow any user to view their future posts in their ‘feed’. This project aims towards creating a minimalistic photo-sharing platform with easy to use and easy to understand interface, with minimal fuss and with a dark theme to be easy on the eye of the user. It can help a user share their experiences with others and make and store memories with them.

* 1. **Tools**

When referring to software, **tools** are accessories that help developers develop software programs more effectively. When referring to hardware, **tools** are what enable a person to install, remove, or perform other actions on their computer or their computer hardware. Below is a listing of some of the tools we required for the development of this system.

Communication Tool: Discord

Version Control Host: Git

Development Environment: Windows

Browser: Chrome

Backend Language: Python

Frontend Language: HTML, CSS, Javascript

DBMS: Firebase

Chapter-3

Project Planning

The most critical phase of managing system projects is planning. Information systems have become increasingly important during the past decade. First, information is now recognized as a vital resource. Second, more and more financial resources are committed to information systems. Third, there is growing need for formal long-range planning with Information systems

* 1. **Development Plan**

The project plan is to carry the design and implementation of the project in a completely step-by-step manner. The entire project is divided in the following phases:-

Phase 1 – Study and Analysis Phase:-

• Understanding the concepts of image sharing platform.

• Understanding various development environments.

• Requirement Gathering from other resources.

• Understanding data storage and retrieval.

Phase 2 - Design Phase:-

• In this phase the design inspiration was taken from various existing image sharing platforms like Instagram and Facebook.

• Once taken, the System Requirement Specification comprising of the system hardware and software requirements was made.

• The coding language is decided.

• Tasks are being allotted to development team.

• Changes are made in individual branches and are merged into a single master branch.

• Data Flow Diagram and Use-Case Diagram are prepared in this phase.

Phase 3 – Coding and Implementation Phase:-

• The design of the system is implemented through actual code.

• Proper mapping of database is done.

• Work is divided into modules.

• Each person does individual code, test on personal machine that gets merged into master after the review is done.

Phase 4 –Testing

• Alpha Testing by the developers.

• Beta Testing by other users.

Chapter-4

System Design

* 1. **Introduction**

Design is the abstraction of a solution; it is a general description of the solution to a problem without the details. Design is view patterns seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required to create the implementation. In this chapter we introduce context diagram, models, system architecture, principal system object, design model and object interface.

* 1. **Context Diagram**

This diagram represents what are the bounders and scope of Pictofy. It describes the main objective of the system and its entities involved.The user can do the following:

✓ Create Profile.

✓ Upload Profile Picture.

✓ Add bio to user’s profile.

✓ Upload Photos.

✓ Comment on the photos of the people you’re following.

✓ Like the pictures uploaded by the people you’re following.

✓ Add captions to your photos.

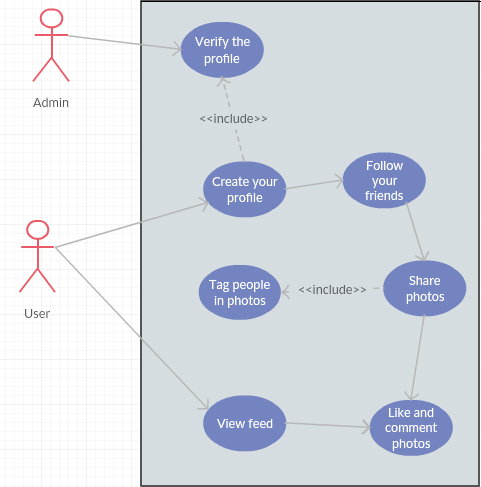
✓ Search for people.

* 1. **Models**
     1. **Interaction Model:-**

Is a dynamic model that shows how the system interacts with its environment. We use a data flow diagram, use case diagram for the depiction of activities.



* + 1. **Use Case Diagram:-**



Chapter-5

Methodology

* 1. **The Agile Model**

**What is Agile Model?**

Agile methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall model.

The agile process is broken into individual models that designers work on. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

In the Agile model, “fast failure” is a good thing. The approach produces ongoing release cycles, each featuring small, incremental changes from the previous release. At each iteration, the product is tested. The Agile model helps teams identify and address small issues on projects before they evolve into more significant problems, and engage business stakeholders and get their feedback throughout the development process.

* 1. **Why Agile?**

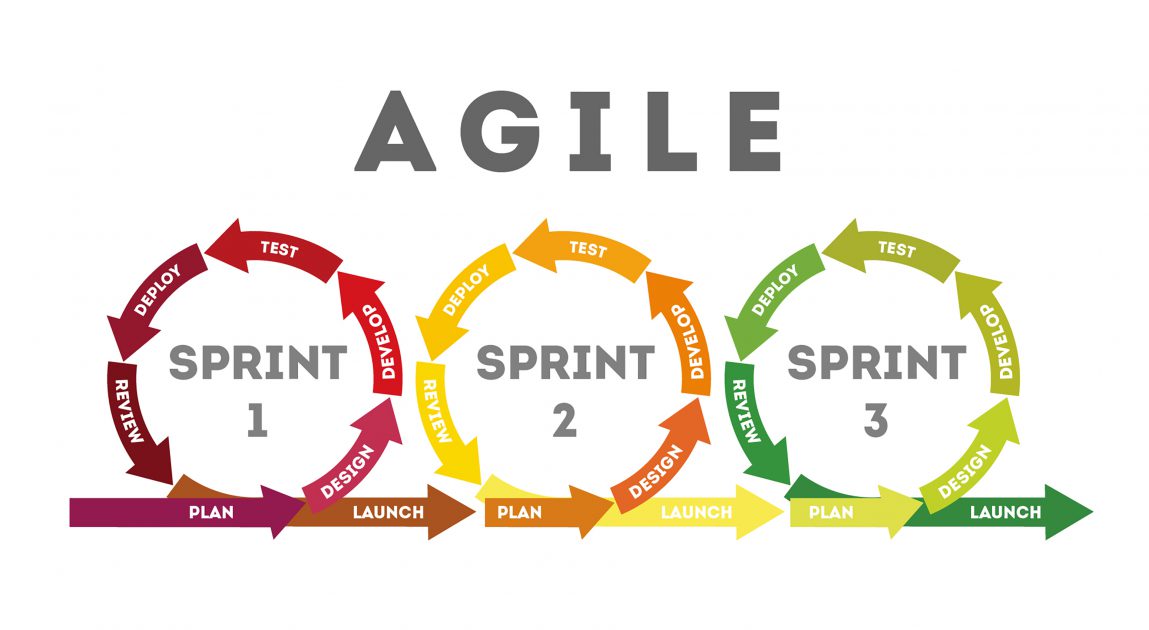
Agile methodology provides customer satisfaction by rapid, continuous delivery of useful software. People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other. Working software is delivered frequently (weeks rather than months).

Agile methodology encourages Face-to-face conversation, which is the best form of communication. It initiates close, daily cooperation between business people and developers. Continuous attention to technical excellence and good design.

This “inspect-and-adapt” approach to development reduces development costs and time to market. Because teams can develop software at the same time they’re gathering requirements, “analysis paralysis” is less likely to impede a team from making progress. And because a team’s work cycle is limited to two weeks, stakeholders have recurring opportunities to calibrate releases for success in the real world. Agile development helps companies build the right product. Instead of committing to market a piece of software that hasn’t been written yet, agile empowers teams to continuously replan their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Agile development preserves a product’s critical market relevance and ensures a team’s work doesn’t wind up on a shelf, never released. Regular adaptation to changing circumstances.

Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. Even late changes in requirements are welcomed.





Chapter-6

Technologies and Implementation

Different languages and tools were used to handle different aspects inside the project:

For handling the backend side of the project, the following languages and tools have been used:

* + 1. **Python**

It is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library

Version used: Python 3.7.2

* + - 1. **Flask**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

* + - 1. **Pyrebase**

Pyrebase is a Python library that is used to establish connection with Google’s firebase.

* + 1. **Firebase**

Firebase is a Google managed real-time database service which stores data and files in a tree. It has its own built in authentication system, as well as a binary file storage system.

For handling the frontend side of the project, the following languages and tools have been used:

* + 1. **Jinja**

Jinja is a web template engine for the Python programming language created by Armin Ronacher. It is similar to the Django template engine but provides Python-like expressions while ensuring that the templates are evaluated in a sandbox. It is a text-based template language and thus can be used to generate any markup as well as sourcecode. The Jinja template engine allows customization of tags, filters, tests, and globals. Also, unlike the Django template engine, Jinja allows the template designer to call functions with arguments on objects. Jinja is Flask's default template engine.

* + 1. **HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

* + 1. **CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

* + 1. **Vanilla JS**

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it. The terms Vanilla JavaScript and Vanilla JS refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.

Technology used for version control:

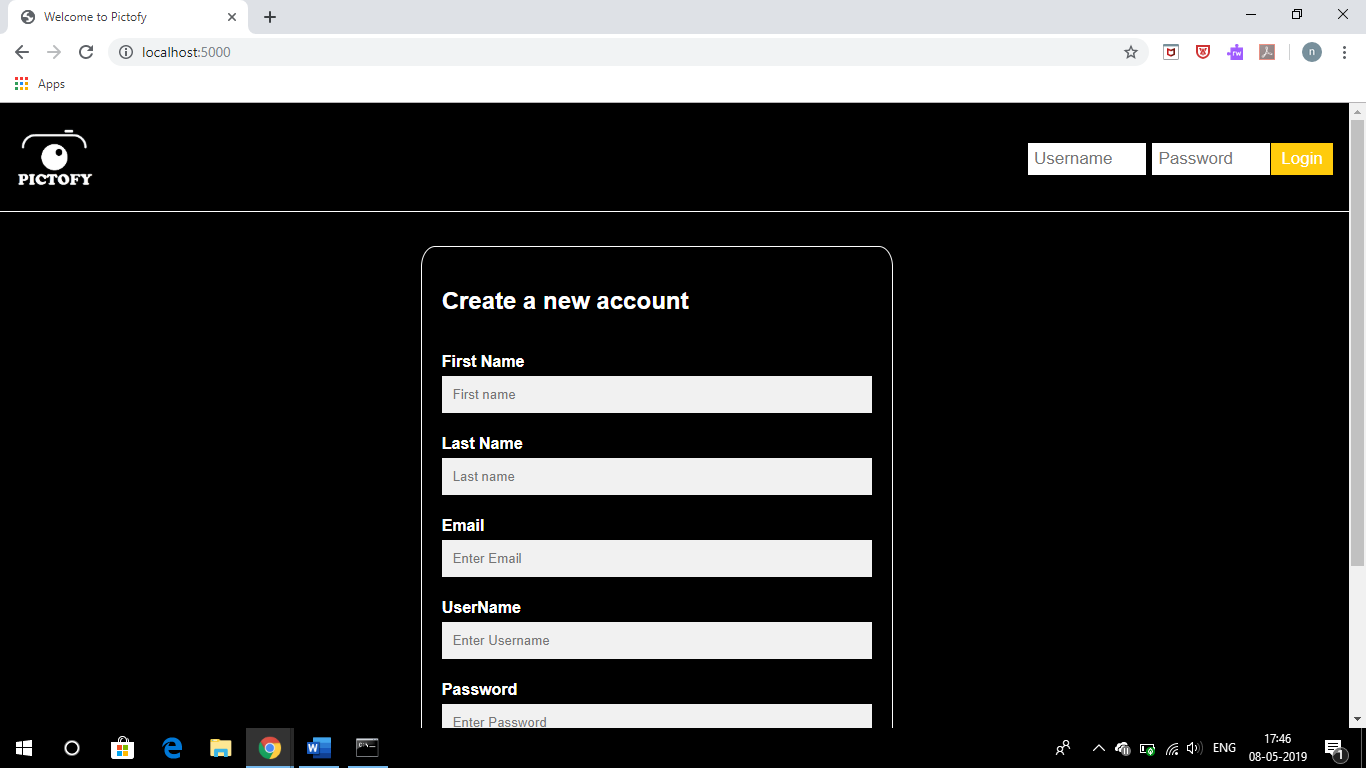
* 1. **Git**

Git is a version control system (VCS) for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for software development, but it can be used to keep track of changes in any files. As a distributed revision control system it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

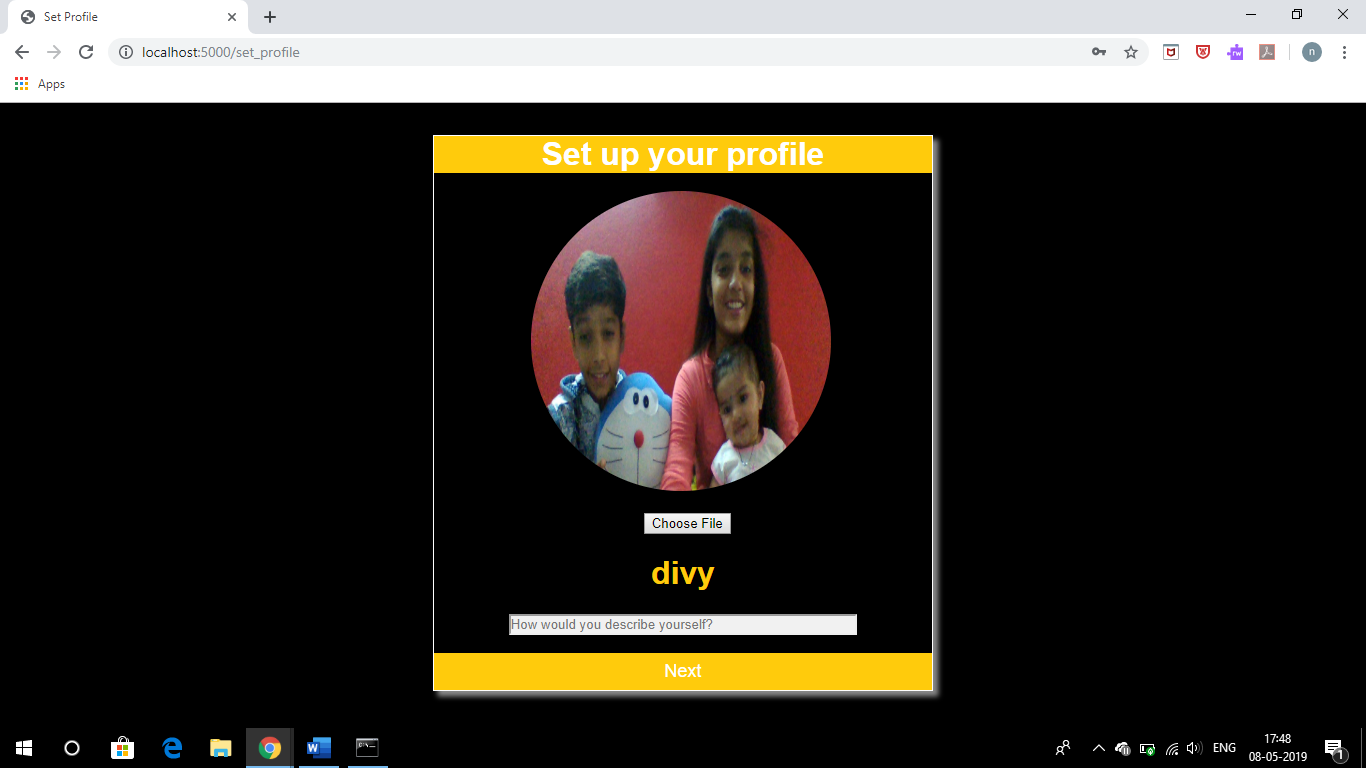
Chapter-6

Technologies and Implementation

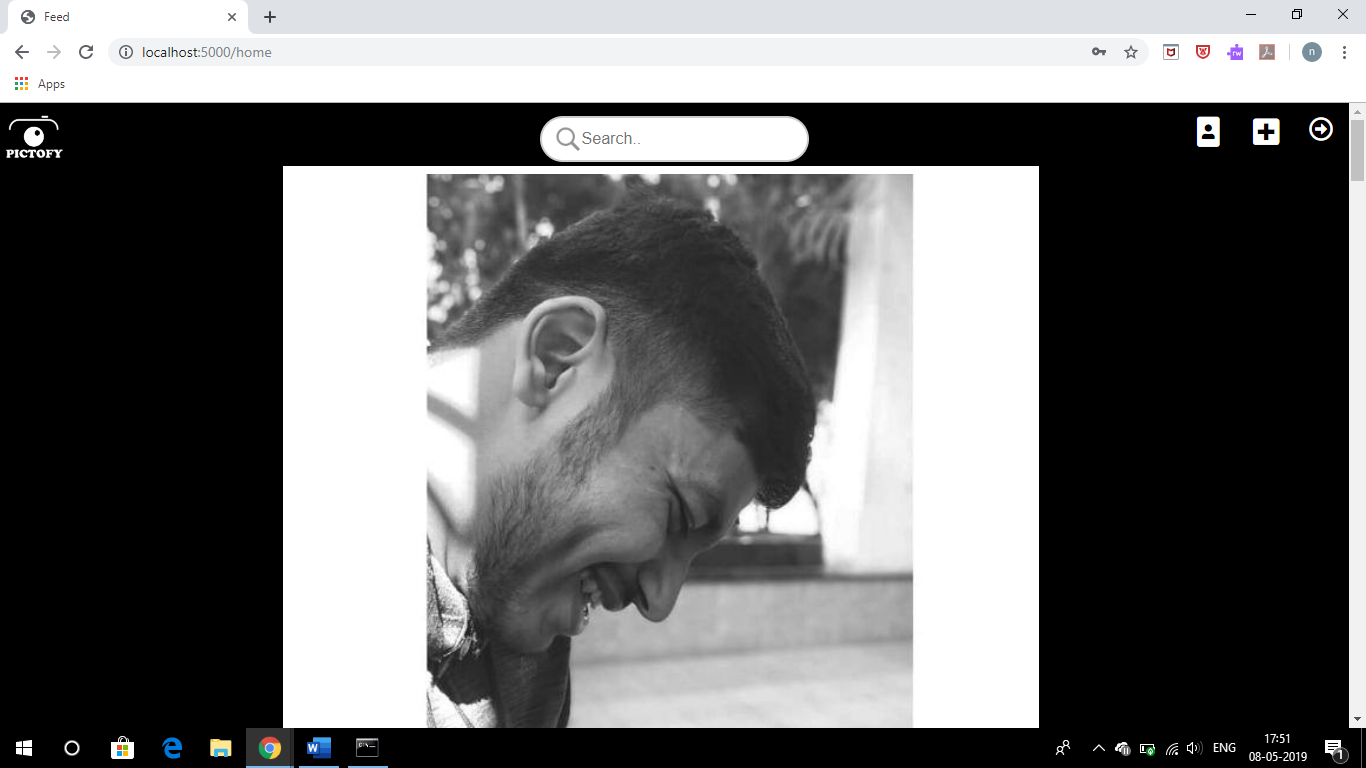
**8.1 Login and Signup Page:**

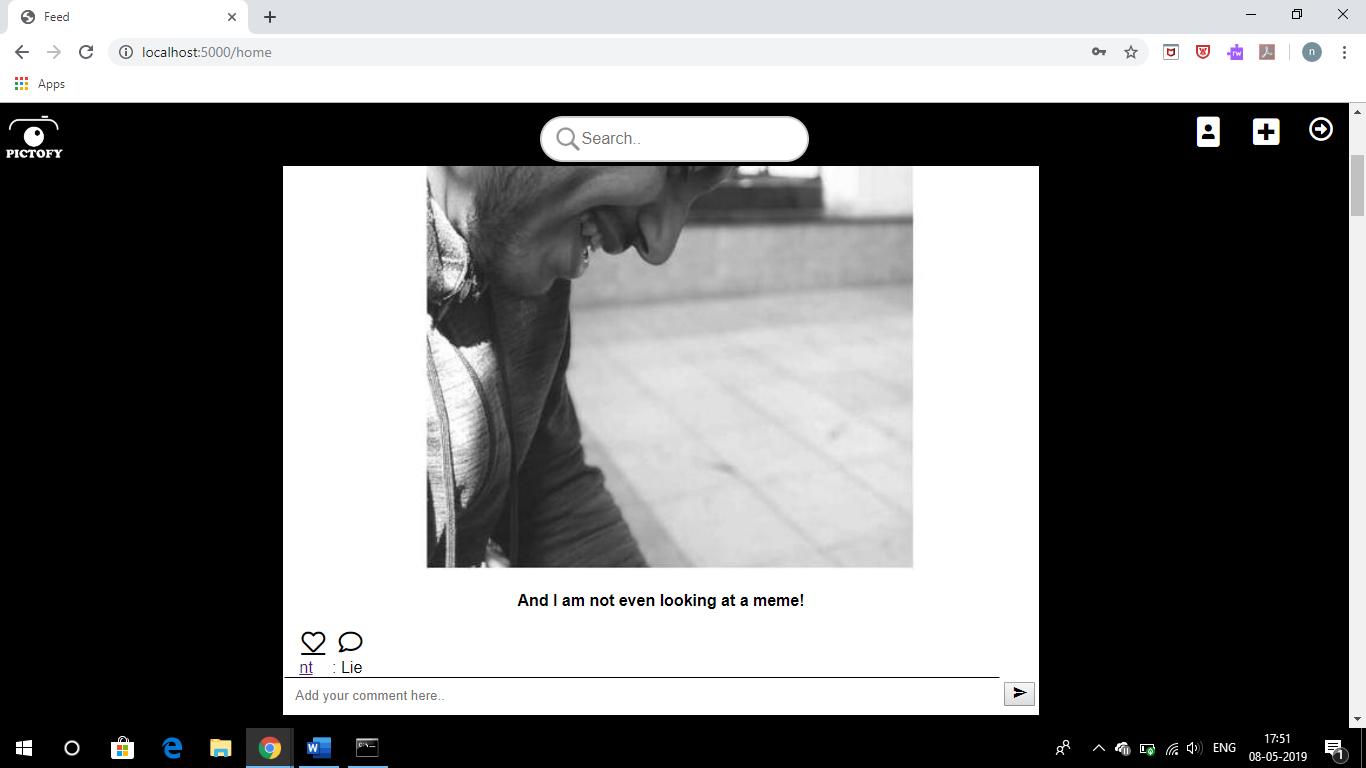


**8.2 Profile Setup:**

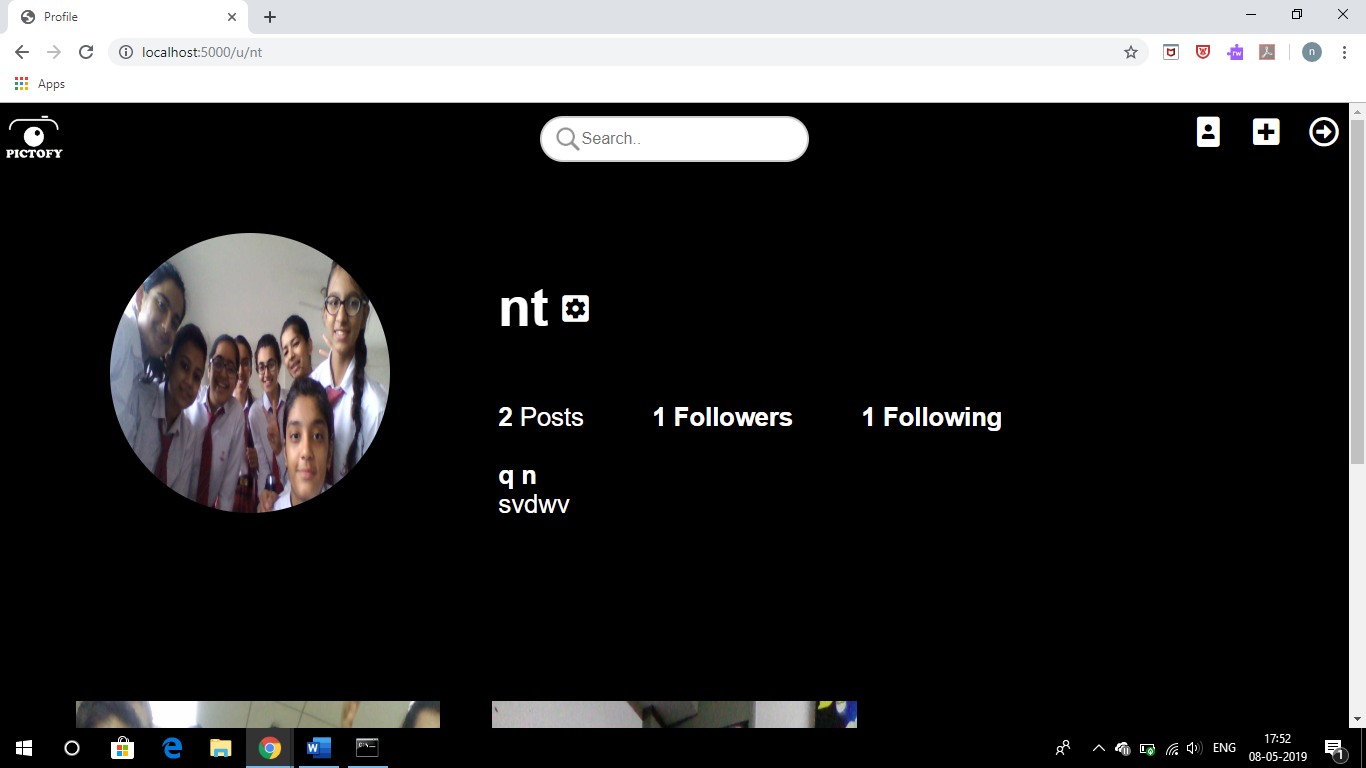


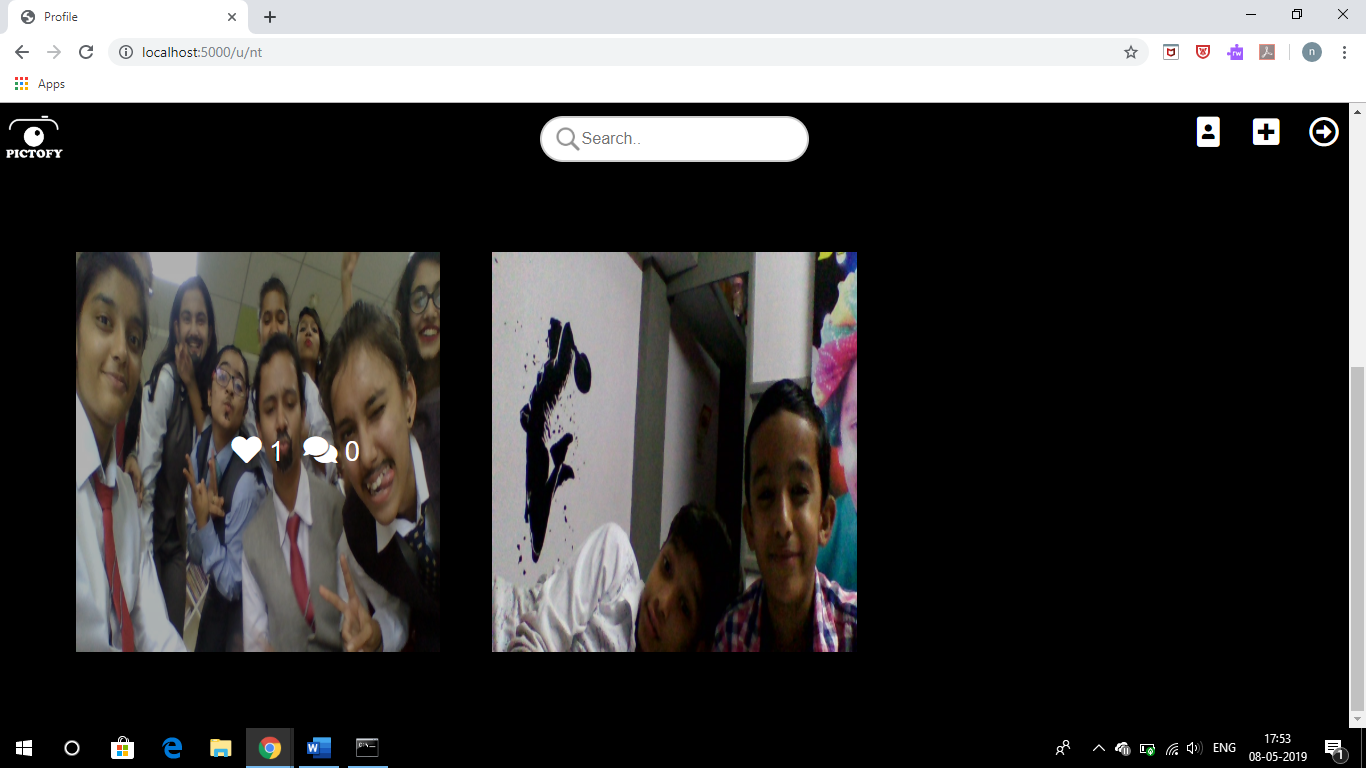
**8.3 Feed:**



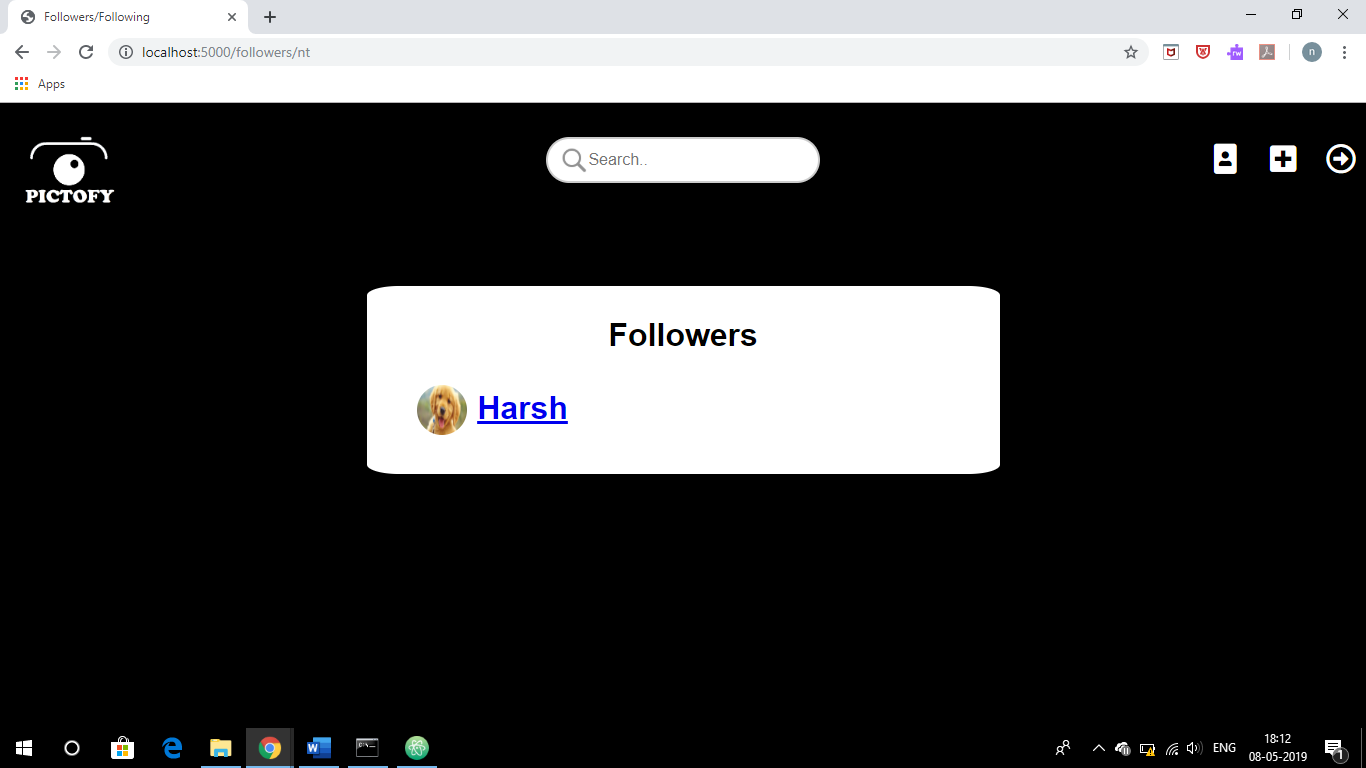


**8.4 Profile:**





**8.5 Followers and Following List:**



**8.6: Image Uploading:**