A

MINI PROJECT REPORT

ON

REVIEW OF THE USE OF AI TECHNIQUES IN SERIOUS GAMES DECISION MAKING AND MACHINE LEARNING

A project report submitted to the
Jawaharlal Nehru Technological University
In partial fulfillment for the award

Bachelor of Technology
In
COMPUTER SCIENCE AND ENGINEERING

Submitted by

B SHYAM RAVI TEJ – 17RJ1A0522
ABDUS SALAM LASKAR– 17RJ1A0504
DHUDUKA RAGHUVEER PRASAD– 17RJ1A0555

Under the esteemed guidance of

Dr. Sateesh Nagavarapu

MALLA REDDY INSTITUTE OF TECHNOLOGY

(Affiliated to JNTU, Hyderabad | Approved by AICTE, New Delhi)
Accredited by NBA, Certificated by ISO 9001:2015
Maisammaguda, Dhullapally, Via: Kompally, Hyderabad - 500100
2017 - 2021

MALLA REDDY INSTITUTE OF TECHNOLOGY

(Sponsored by Malla Reddy Educational Society)
Accredited by NBA, Certificated by ISO 9001:2015
Approved by AICTE & Affiliated to JNTU, Hyderabad
Maisammaguda, Dhullapally Post, (Via: Kompally), Secunderabad - 500100.

CERTIFICATE

This is to certify that mini project work entitled **“Review of the Use of AI Techniques in Serious Games Decision Making and Machine Learning”** is a bonafide work carried by **B SHYAM RAVI TEJ (17RJ1A0522), ABDUS SALAM LASKAR (17RJ1A0504), DHUDUKA RAGHUVEER PRASAD (17RJ1A0555)** of **COMPUTER SCIENCE AND ENGINEERING DEPARTMENT** in **MALLA REDDY INSTITUTE OF TECHNOLOGY** and submitted to **JNT UNIVERSITY, Hyderabad** in the partial fulfillment of the requirement for the award of **BACHELOR OF TECHNOLOGY**.

Project Guide	**Project Coordinator**	**Head of Department**
Dr. Sateesh Nagavarapu	**Dr. Kalimuthu**	**Dr. Sateesh Nagavarapu**

External Examiner

DECLARATION

We hereby declare that the project entitles **“Review of the Use of AI Techniques in Serious Games Decision Making and Machine Learning”** submitted to **Malla Reddy Institute of Technology**, affiliated to Jawaharlal Nehru Technological University Hyderabad (**JNTUH**) for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is a result of work done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

B SHYAM RAVI TEJ - 17RJ1A0522
ABDUS SALAM LASKAR - 17RJ1A0504
DHUDUKA RAGHUVEER PRASAD - 17RJ1A0555

ACKNOWLEGEMENT

We are happy to express our deep sense of gratitude to the **Principal** of the college **Dr. K. Srinivas Rao, Professor, Malla Reddy Institute of Technology** for having provided us the adequate facilities to pursue our project.

We would like to thank **Dr. Sateesh Nagavarapu, Associate Professor and Head, Department of Computer Science and Engineering, Malla Reddy Institute of Technology,** for having provided the freedom to use all the facilities available in the department, especially the laboratories and library.

We also grateful to our project coordinator **Dr. Kalimuthu, Associate professor**, **Department of Computer Science and Engineering**, **Malla Reddy institute of Technology**, for extending his support and assisting us throughout our project work.

We very grateful to our project guide, **Dr. Sateesh Nagavarapu**, **Associate Professor and Head of the Department, Computer Science and Engineering**, **Malla Reddy institute of Technology**, for his extensive patience and guidance throughout our project work.

We sincerely thank all the **Teaching and Non-teaching staff** of the department of **Computer Science and Engineering**, also like to thank our **Classmates** for their timely suggestions, healthy criticism and motivation during the course of our project work. We thank our **parents** who were the backbone behind our deeds.

Finally, we express our immense gratitude with pleasure to all **individuals** who have either directly or indirectly contributed to our need at right time for the development and success of our project work.

ABSTRACT

The video-games market has become an established and ever-growing global industry. The health of the video and computer games industry, together with the variety of genres and technologies available, mean that videogame concepts and programs are being applied in numerous different disciplines. One of these is the field known as serious games. The main goal of this article is to collect all the relevant articles published during the last decade and create a trend analysis about the use of certain artificial intelligence algorithms related to decision making and learning in the field of serious games. A categorization framework was designed and outlined to classify the 129 papers that met the inclusion criteria. The authors made use of this categorization framework for drawing some conclusions regarding the actual use of intelligent serious games. The authors consider that over recent years enough knowledge has been gathered to create new intelligent serious games to consider not only the final aim but also the technologies and techniques used to provide players with a nearly real experience. However, researchers may need to improve their testing methodology for developed serious games, so as to ensure they meet their final purposes.

INDEX

TITLES **PAGE NO**

CONTENTS

LIST OF FIGURES PAGE NO

LIST OF SCREENSHOTS PAGE NO

CHAPTER - 1
INTRODUCTION

1. INDRODUCTION

Since the beginning of the twenty-first century, the video-games market has become an established and ever-growing global industry. The health of the video and computer games industry, together with the variety of genres and technologies available, mean that videogame concepts and programmes are being applied in numerous different disciplines. One of these is the field known as serious games. The term serious games were coined by North American researcher Clark Abt in his book “Serious Games” in the 70s. Although the spirit of this trend has been maintained over the last decades, the technologies, applications and scope have changed significantly. Serious games represent a genre designed to be more than “just” fun. Moreover, the educational value associated to serious games goes beyond academic purposes, locating their focus on skill practice and entertainment value during exposure. The main purpose of a serious game is both to be fun and entertaining, and educational. A serious game is thus designed both to be attractive and appealing to a broad target audience, and to meet specific educational goals. They are designed to foster knowledge, skills or routine habits in the player. Serious games span a broad range of fields and areas of expertise. In the literature, serious games were divided into several categories based on different classification schemes. These models can be divided into two main categories: market-based and purpose-based classifications. Several authors established different categories of market-based classification. This segmentation is based primarily on the different “markets” or fields the serious games are developed for. The different segments identified in the literature are: − Military games, government games, educational games, corporate games, health-care games, political, religious and art games. − Health, public policy, strategic communication, human performance engineering, training and simulation, education, game evaluation. − Educational, social change, military, occupation and marketing. − Defence, teaching and training, advertising, information and communications, health, culture and activism. − K-12 edutainment, higher education, health-care, corporate, military, non-government and other. Different authors also provide different categories for purpose-based classifications, or intention they were intended to satisfy. − Advergames, activism games, training and simulation games, edugames, newsgames and edumarket games. − Business games, health and medicine, news, activism, advergames and political games. Other classification approaches use alternatives to the market/purpose distinction, proposing labels or tags as means of classification. Meanwhile, the G/P/S classification model considers a gameplay, purpose and scope trio. Finally, classification can also be conducted according to learning principles, target age group or game platform. The main goal of this article is to collect all the relevant articles published during recent years and create a trend analysis about the use of certain artificial intelligence algorithms related to decision making and machine learning in the field of serious games. A categorization framework was designed and outlined to classify available articles in the literature. The authors made use of this categorization framework for performing an analysis of the actual use of intelligent serious games.

CHAPTER – 2

LITERATURE SURVEY

2. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

2.1 SOFTWARE ENVIRONMENT

PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library

DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.

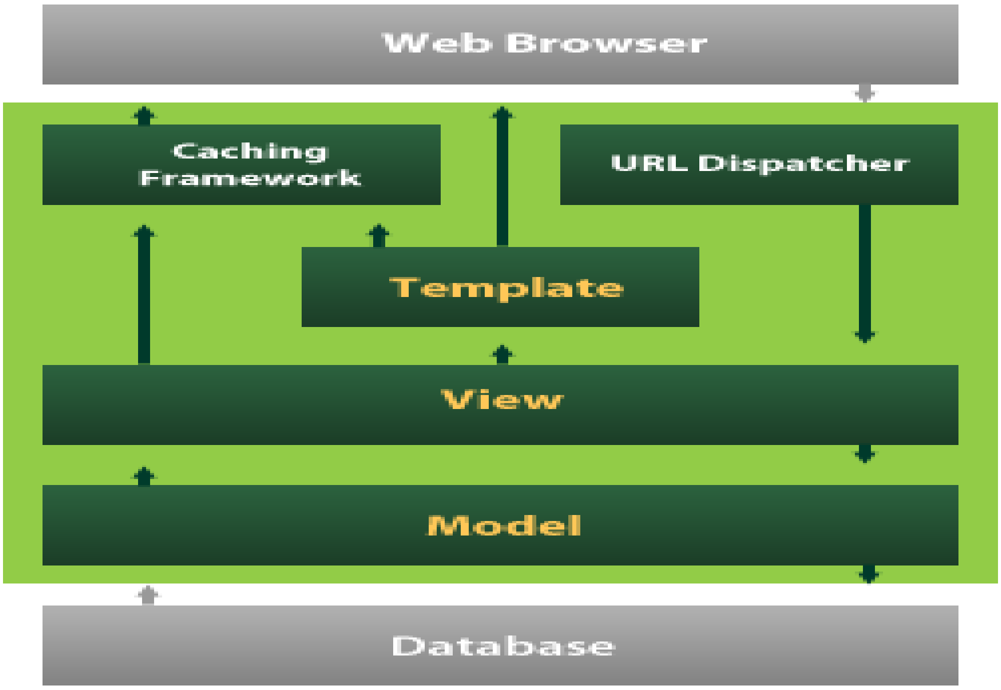


Fig. 2.1 Django Framework

Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

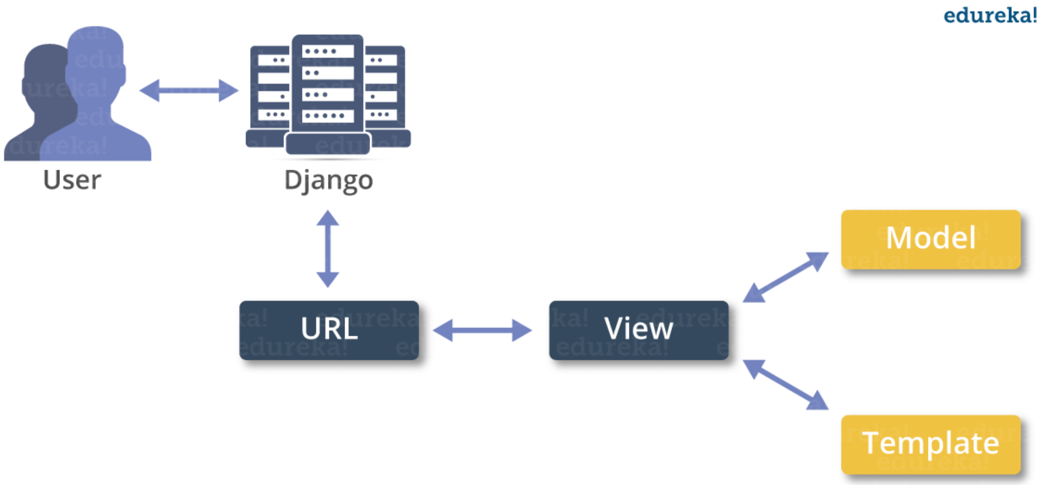


Fig. 2.2 Django Working Model

2.2 WHY CHOOSE PYTHON

If you’re going to write programs, there are literally dozens of commonly used languages to choose from. Why chose Python? Here are some of the features that make Python an appealing choice.

Python is Popular

Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day.

According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the Popularity of Programming Language Index.

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you’d like to dig deeper into Python salary statistics and job opportunities, you can do so here.

Python is interpreted

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It’s Not That Simple for all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

Conclusion

This section gave an overview of the Python programming language, including:

- A brief history of the development of Python
- Some reasons why you might select Python as your language of choice

Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer.

Python is an open source programming language that was made to be easy-to-read and powerful. A Dutch programmer named Guido van Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp. Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

Python use [change / change source]

Python is used by hundreds of thousands of programmers and is used in many
places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.
Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.
Some things that Python is often used for are:

- Web development
- Scientific programming
- Desktop GUIs
- Network programming
- Game programming

CHAPTER – 3

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

There are numerous studies about Finding and Setting the Trend in Use of the AI Techniques in serious games. The main purpose of a serious game is both to be fun and entertaining, and educational. A serious game is thus designed both to be attractive and appealing to a broad target audience, and to meet specific educational goals. They are designed to foster knowledge, skills or routine habits in the player. The existing systems are not setting the current trends. This will find and predict the current trend in the gaming world. This article consists of an analysis of serious games, offering a literary review of their use combined with certain artificial intelligence techniques in the area of decision making and machine learning. Other areas such as pathfinding were initially considered for study but were not sized enough for further analysis, so they were excluded from this review. The article is divided into several sections. First of all, a complete methodology is introduced presenting the form and function of data collected from the literature review. Then, each of the following subsections presents a contextualization and classification of available articles. Finally, the article ends with the discussion and conclusion section.

DISADVANTAGES:

Numerical measure of how different are two data objects

– Lower when objects are more alike

– Minimum dissimilarity is often 0

–Upper limit varies

1. Does not work well with large dataset

2. Does not work well with high dimensions

3.2 PROPOSED SYSTEM:

The Best Decision Making and machine learning algorithms are found to use in the serious games design which supports the developer to do it effective way. AI techniques were applied with a wide variety of final. The most common implementations were for altering the game flow or for assessing/classifying users' state and behavior while playing. The productions of intelligent serious games that dynamically adapt themselves to users' needs and performance have been proved to be efficient in terms of improvement comparisons.

ADVANTAGES:

- More accurate results
- It is robust
- It is efficient

3.3 ARCHITECTURE:

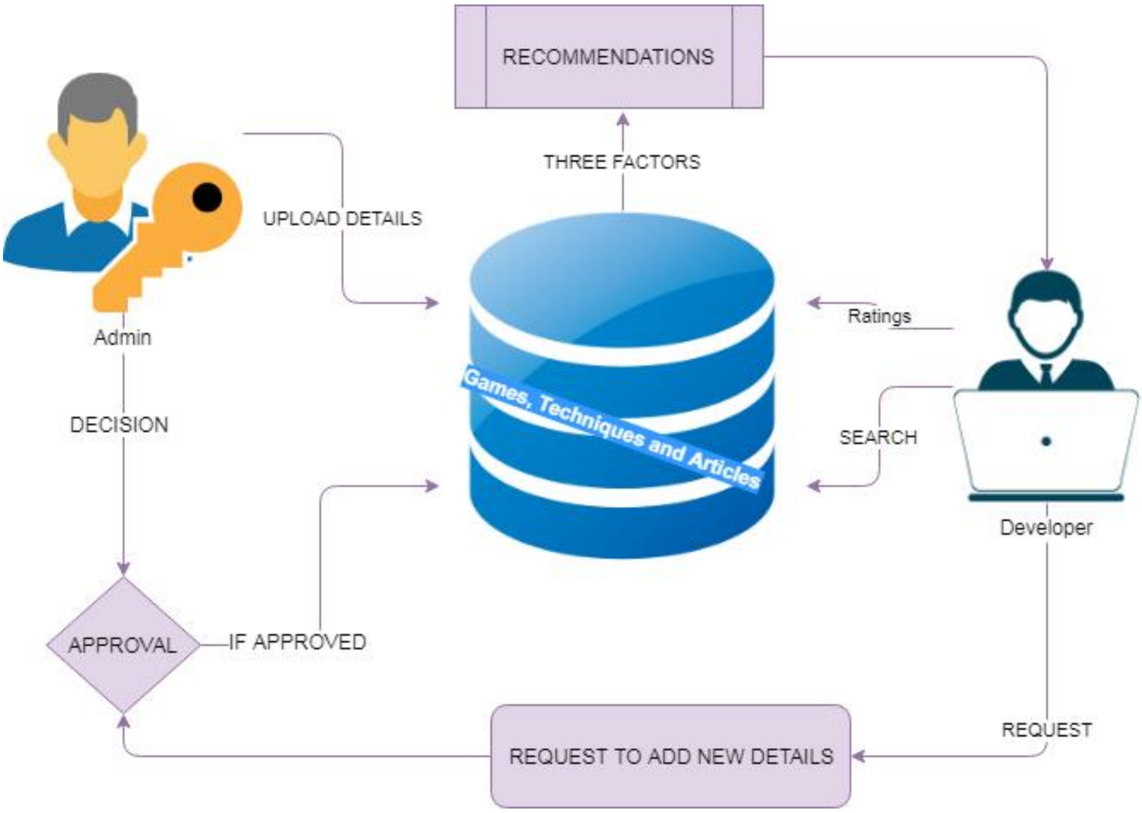


Fig. 3.1 System Architecture

CHAPTER – 4

FEASIBILITY STUDY

4. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ♦ ECONOMICAL FEASIBILITY
- ♦ TECHNICAL FEASIBILITY
- ♦ SOCIAL FEASIBILITY

4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER – 5

SYSTEM SPECIFICATION

5. SYSTEM SPECIFICATION:

5.1 HARDWARE REQUIREMENTS:

❖ **System**	**:**	Pentium IV 2.4 GHz.
❖ **Hard Disk**	**:**	40 GB.
❖ **Floppy Drive**	**:**	1.44 Mb.
❖ **Monitor**	**:**	14’ Colour Monitor.
❖ **Mouse**	**:**	Optical Mouse.
❖ **Ram**	**:**	512 Mb.

5.2 SOFTWARE REQUIREMENTS:

❖ **Operating system**	**:**	Windows 7 Ultimate.
❖ **Coding Language**	**:**	Python 3.
❖ **Front-End**	**:**	Python.
❖ **Back-End**	**:**	MySQL **(**phpMyadmin, Django ORM)
❖ **Designing**	**:**	Html, CSS, JavaScript.
❖ **Data Base**	**:**	MySQL (WAMP Server).

CHAPTER – 6

SYSTEM DESIGN

6. SYSTEM DESIGN

Data Flow Diagram / Use Case Diagram / Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

6.1 SEQUENCE DIAGRAM:

AUTHORITY:

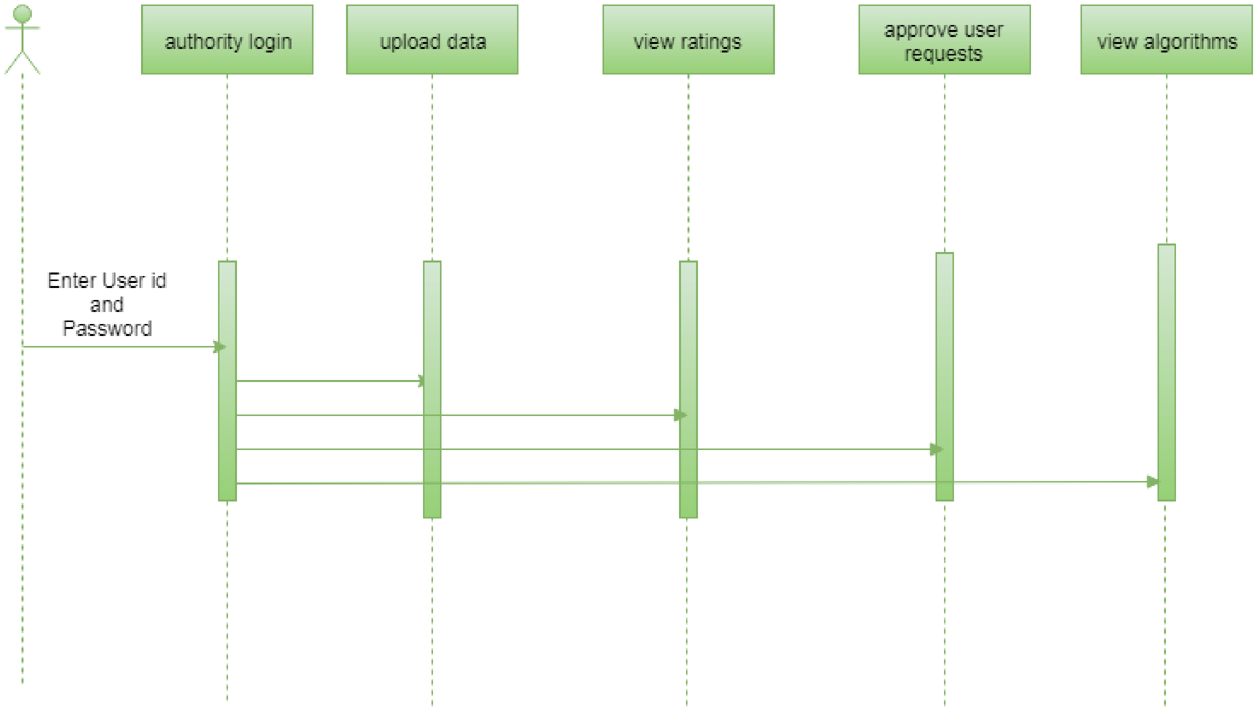


Fig. 6.1 (a) Sequence Diagram (Authority)

DEVELOPER:

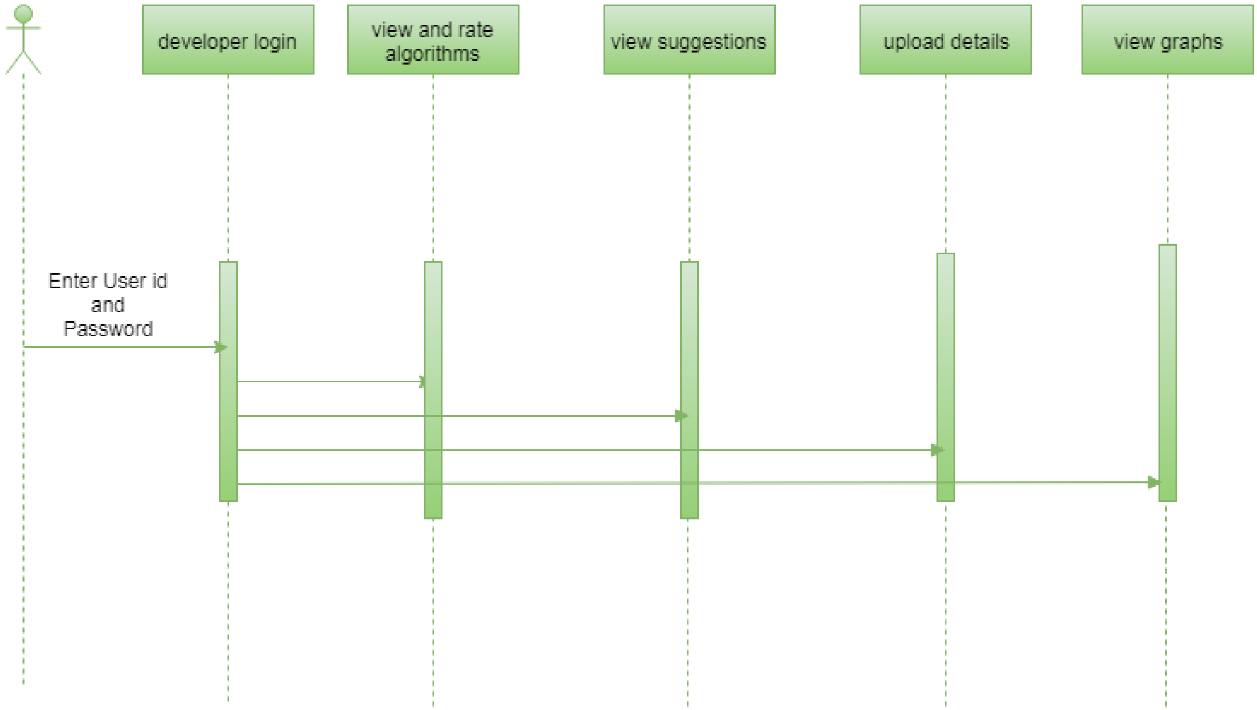


Fig. 6.1 (b) Sequence Diagram (Developer)

6.2 ACTIVITY DIAGRAM:

AUTHORITY:

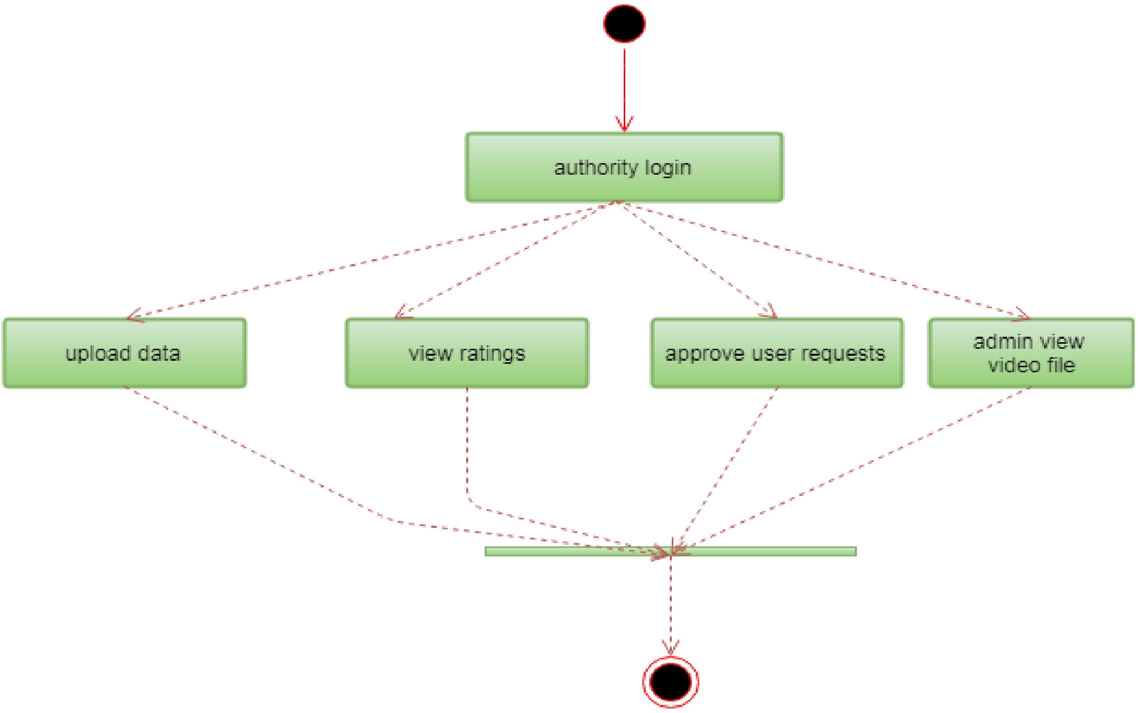


Fig. 6.2 (a) Activity Diagram (Authority)

DEVELOPER:

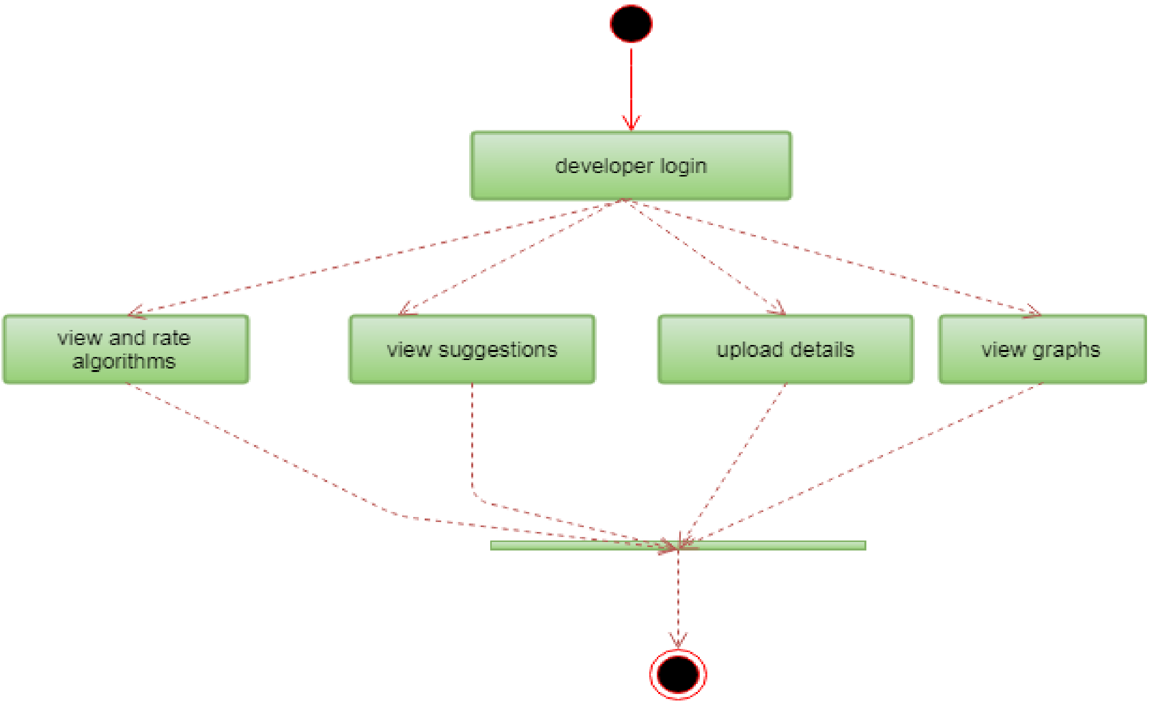


Fig. 6.2 (b) Activity Diagram (Developer)

6.3 USECASE DIAGRAM:

AUTHORITY:

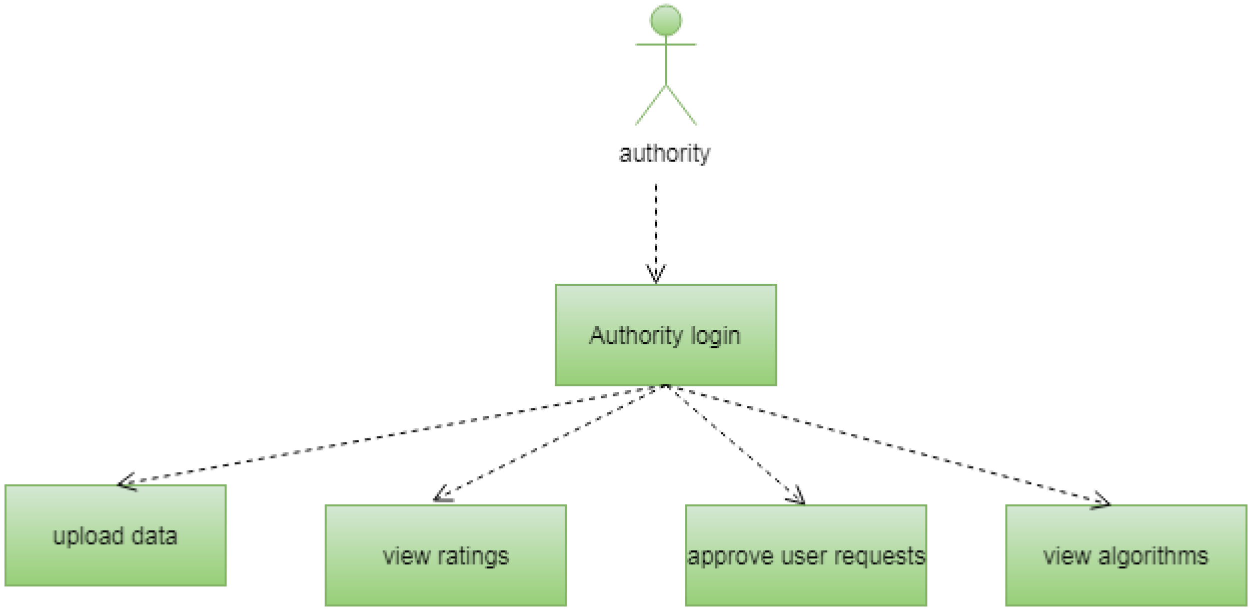


Fig. 6.3 (a) Usecase Diagram (Authority)

DEVELOPER:

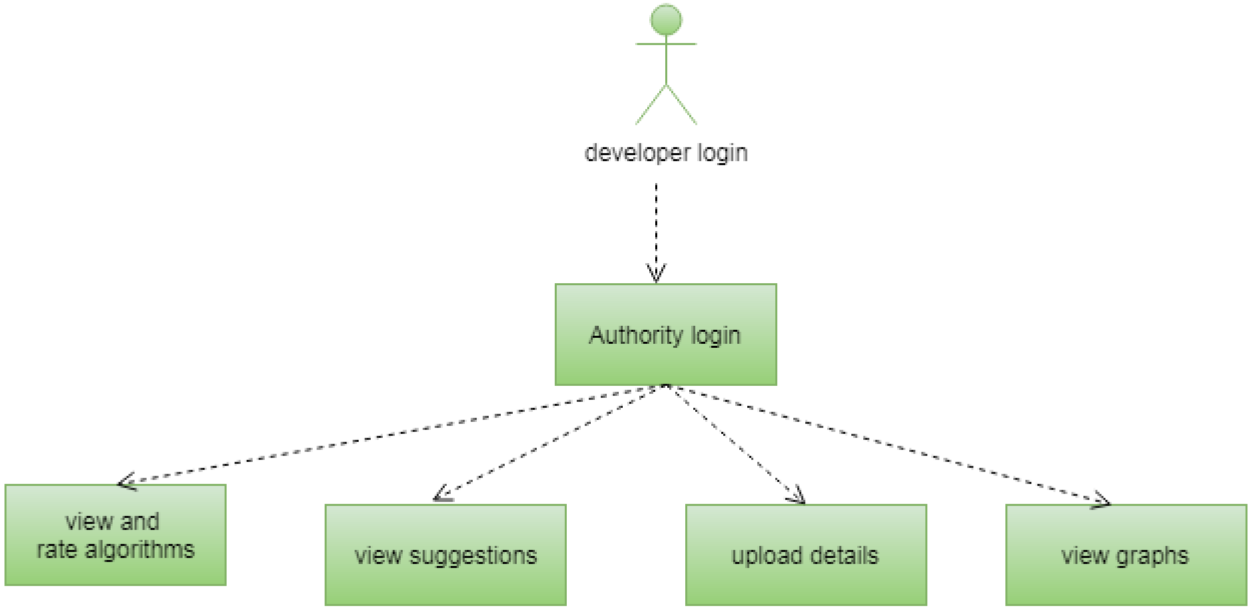


Fig. 6.3 (b) Usecase Diagram (Developer)

6.4 COMPONENT DIAGRAM:

AUTHORITY:

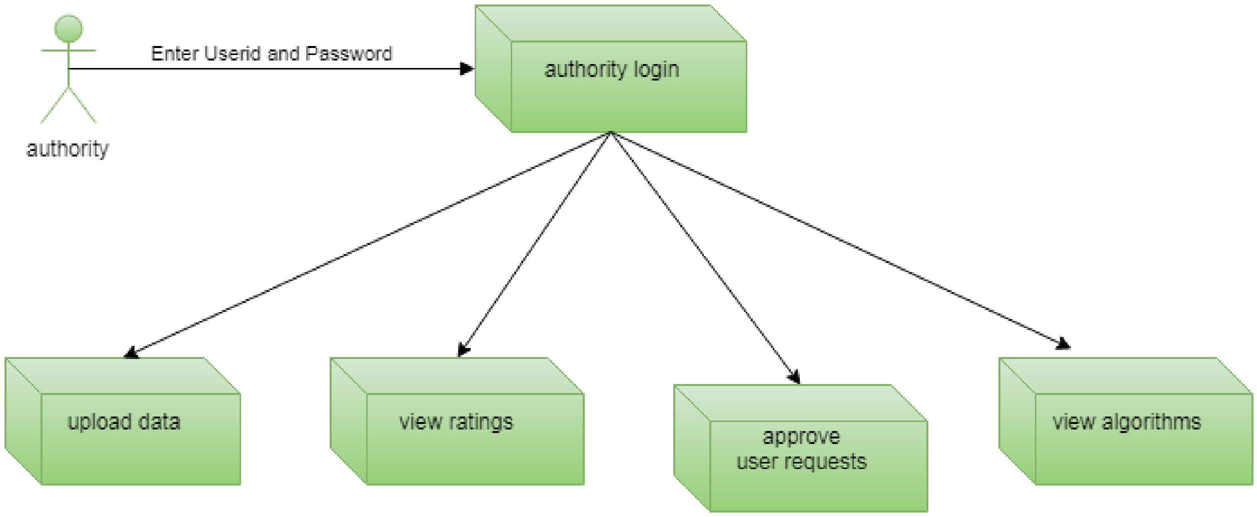


Fig. 6.4 (a) Component Diagram (Authority)

DEVELOPER:

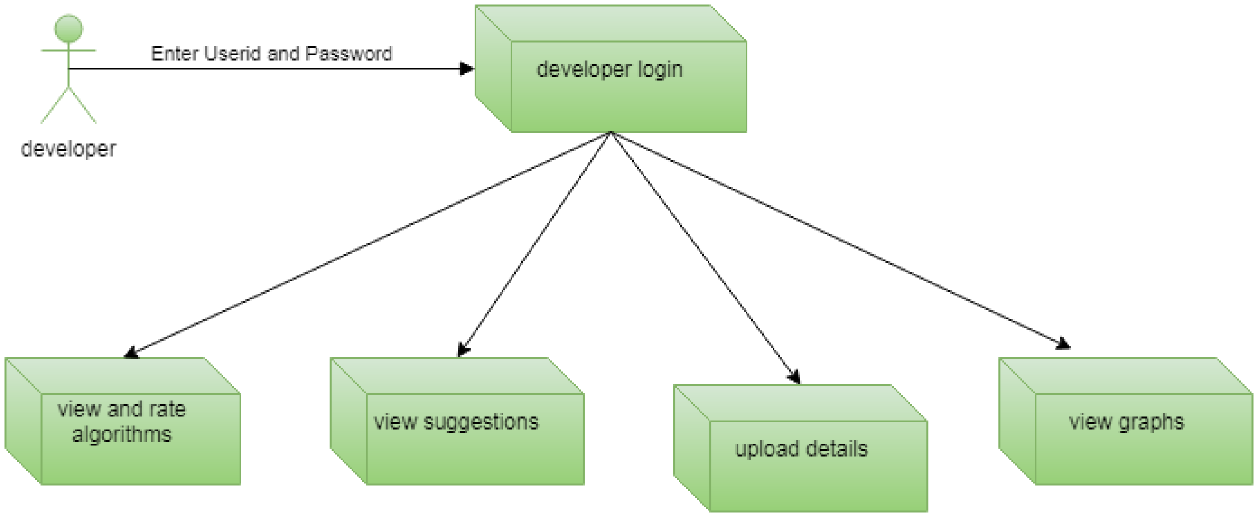


Fig. 6.4 (b) Component Diagram (Developer)

6.5 CLASS DIAGRAM:

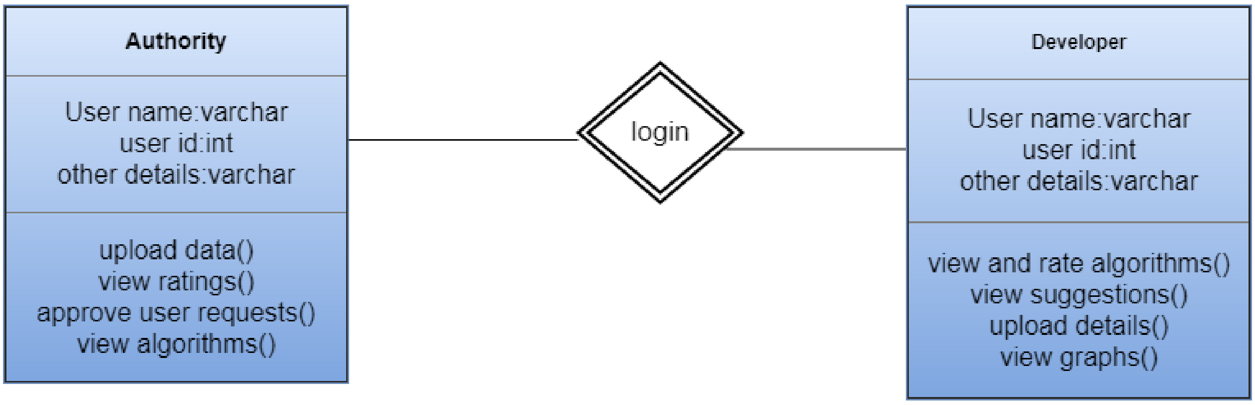


Fig. 6.5 Class Diagram

6.6 ER DIAGRAM:

AUTHORITY:

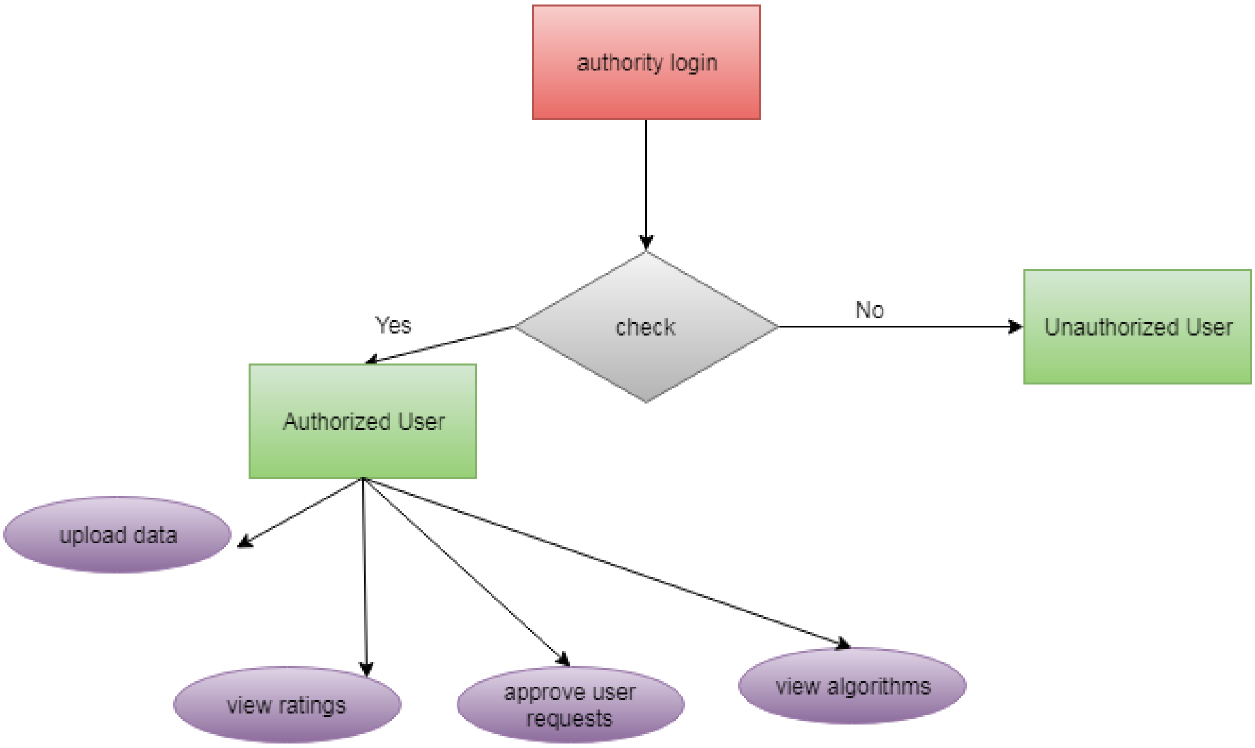


Fig. 6.6 (a) ER Diagram (Authority)

DEVELOPER:

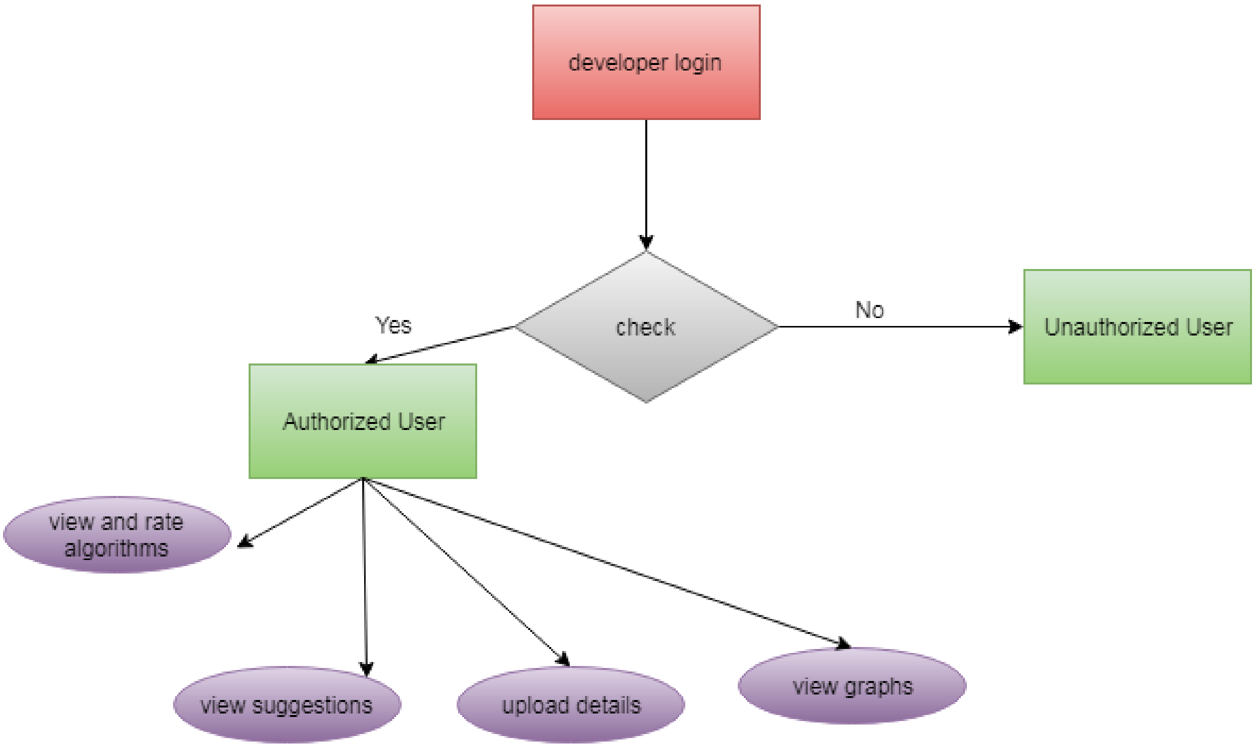
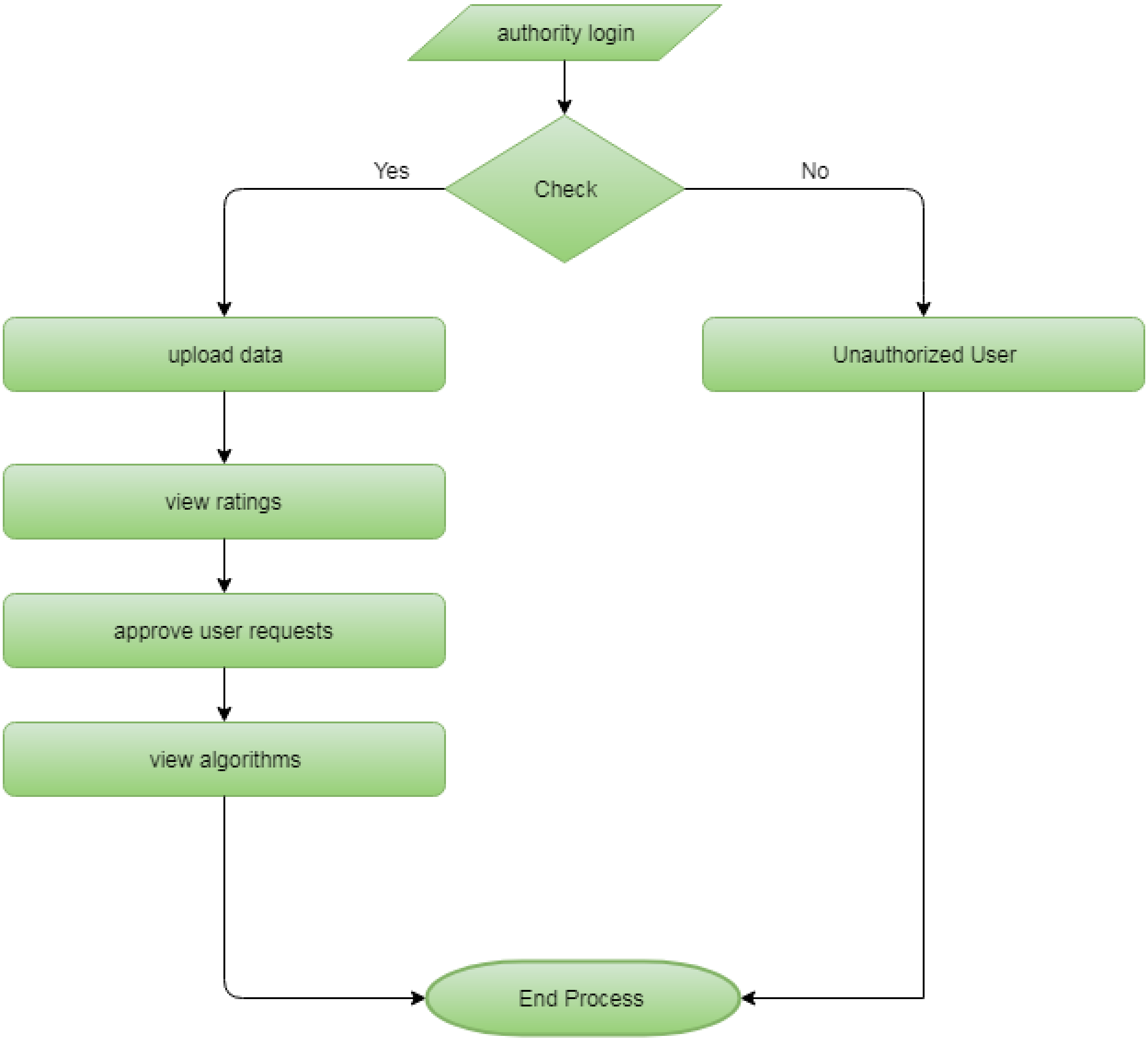


Fig. 6.6 (b) ER Diagram (Developer)

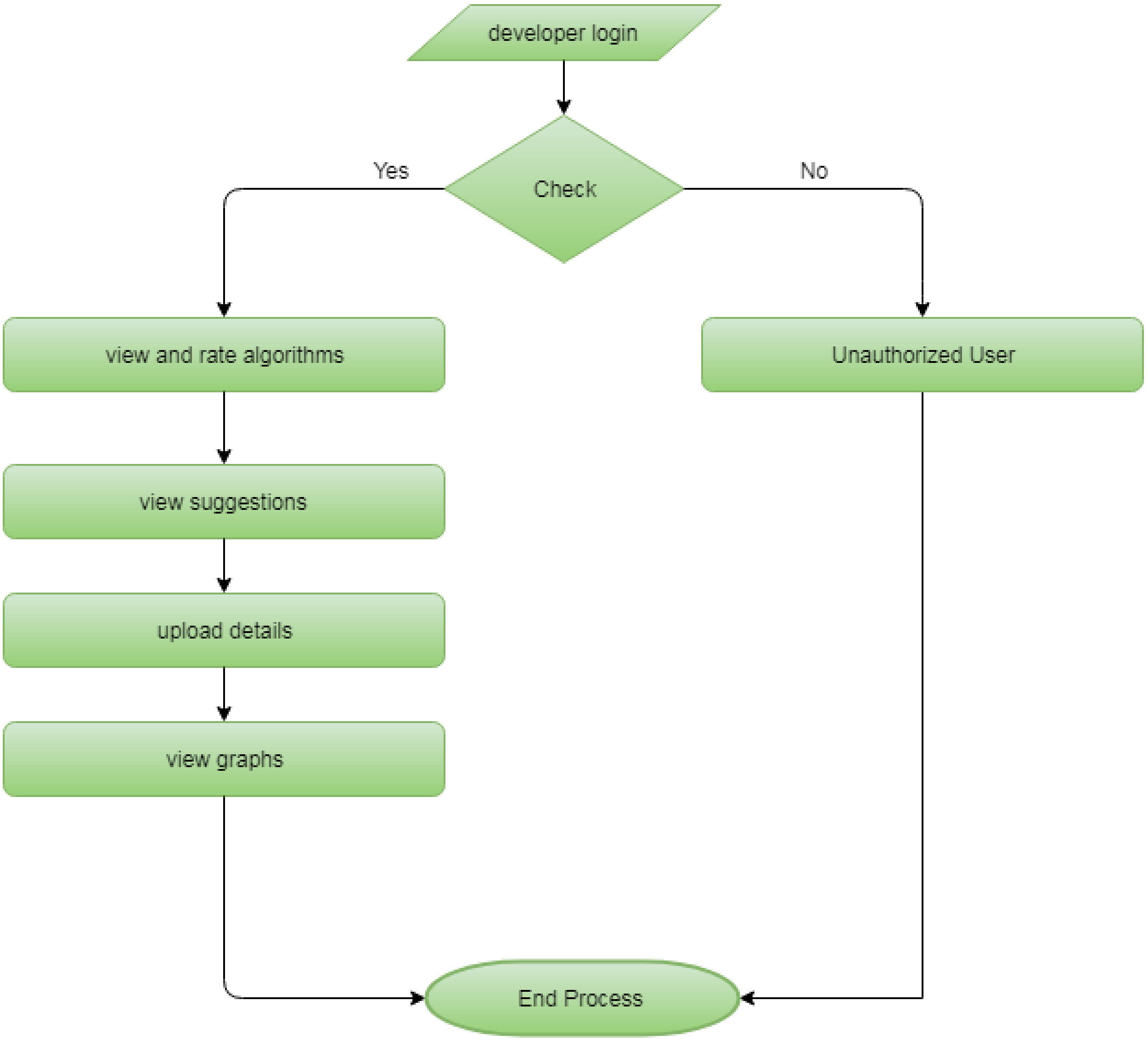
6.7 DATA FLOW DIAGRAM:

AUTHORITY:



6.7 (a) Data Flow Diagram (Authority)

DEVELOPER:



6.7 (b) Data Flow Diagram (Developer)

CHAPTER – 7

IMPLEMENTATION

7. IMPLEMENTATION

7.1 MODULES

- ❖ UPLOAD AND SEARCH GAMES AND TECHNIQUES
- ❖ RECOMMENDATIONS
- ❖ REQUEST FOR ADD
- ❖ RATINGS AND REVIEWS

MODULE DESCRIPTION:

UPLOAD AND SEARCH GAMES AND TECHNIQUES

This is admin side module. Admin Uploads the Games, Algorithm techniques and Technical Articles. The uploaded details were maintained in database. Developer can view the details by searching all over the details. The technical details are listed in the Developers window which is very useful for them to get to know about the current technologies and details used in other developers of same knowledge.

RECOMMENDATIONS

The Recommendation is set the trend in the gaming world of development. The Recommendations are based on three factors which is listed follows. Based on usage of algorithms in industry, based on the Technical Articles Presented and Ratings and Reviews of the Developer in the systems. These recommendations are given to developer to set the industry trends.

REQUEST FOR ADD

In this module add the developer technical details based on their wish. Once the developer is request admin to add their details to database. The admin can view the details of the users and they can take the decision on the developer’s request weather the details should add to the database. The Details will be added and take account of the recommendation details.

RATINGS AND REVIEWS

The Ratings and reviews are given to the recommend or used techniques. The Details were taken into the consideration to add the recommendation system and get the feedback of how technology is used and effective and user-friendly.

7.2 SAMPLE CODE

BASE.HTML

```
{% load staticfiles %}
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Simple PureCss dropdown menu with following subnav</title>
  <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,700'  rel='stylesheet'
type='text/css'>
  <link rel="stylesheet"
            href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.cs        ">
  <link rel="stylesheet" href="{% static 'nav/css/style.css' %}">
  <style>
      body{
            background:rgb(94,156,156);
      }
      </style>
</head>


<body>
<img src="{% static 'images/algorithm2.png' %}" style="width:15%;float:left;">
  <!-- not responsive yet -->
<nav>
  <ul id="main">
    <li>
        <a href="{% url 'developer:home' %}">Home</a>
    </li>
    <li>
        <a href="{% url 'developer:uploads' %}">Upload</a>
        <ul class="drop">
            <div>
                <li><a href="{% url 'developer:uploads' %}">Upload New File</a></li>

                <li><a href="{% url 'developer:myuploads' %}">My Uploads</a></li>

                <li><a href="{% url 'developer:uploadstatus' %}">Upload Request
            Status</a></li>

            </div>

        </ul>

      </li>

    <li>

        <a href="{% url 'developer:ratingdetails' %}">Rating Details</a>

        <ul class="drop">

            <div>

                <li><a href="{% url 'developer:ratingdetails' %}">Ratings</a></li>

                <li><a href="{% url 'developer:yourratings' %}">Your Ratings</a></li>

            </div>

        </ul>

    </li>

    <li>

        <a href="{% url 'developer:graphs' 'bar' %}">Graphs</a>

        <!--<ul class="drop">

            <div>

                <li><a href="{% url 'developer:graphs' 'bar' %}">Bar Chart</a></li>

                <li><a href="{% url 'developer:graphs' 'pie' %}">Pie Chart</a></li>

                <li><a href="{% url 'developer:graphs' 'spline' %}">Spline Chart</a></li>

                <li><a href="{% url 'developer:graphs' 'line' %}">Line Chart</a></li>

                <li><a href="{% url 'developer:graphs' 'column' %}">Column
            Chart</a></li>

            </div>

        </ul>-->

    </li>

    <li>

        <a href="{% url 'developer:logout' %}">Logout</a>

    </li>

    <div id="marker">
```

```
    </div>
  </ul>
</nav>
{% block mainblock %}
{% endblock %}

</body>
</html>
```

CHAPTER – 8

SYSTEM TESTING

8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

8.1 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Chapter – 9

INPUT AND OUTPUT DESIGN

9. INPUT AND OUTPUT DESIGN

9.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

9.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

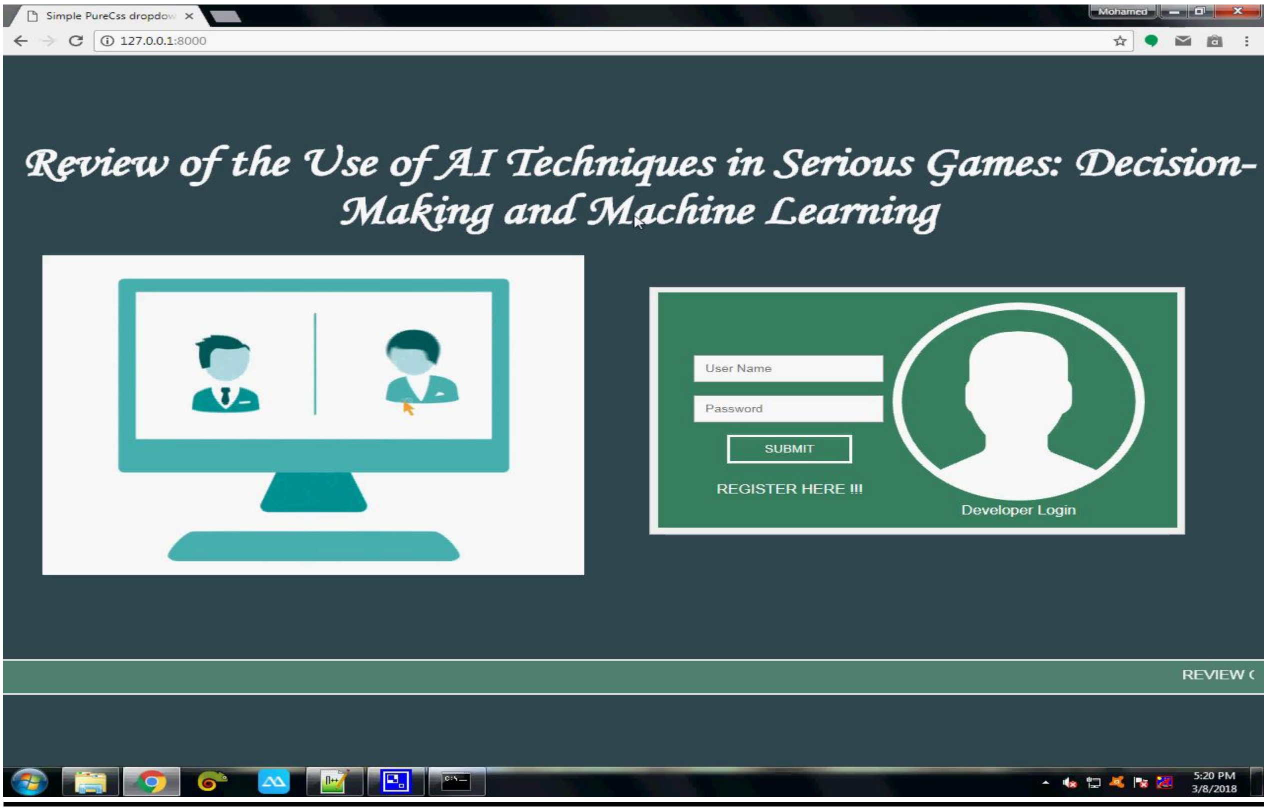
The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER – 10

SCREENSHOTS

10. SCREENSHOTS



Review of the Use of AI Techniques in Serious Games: Decision-Making and Machine Learning

Fig:10.1 Developer Login

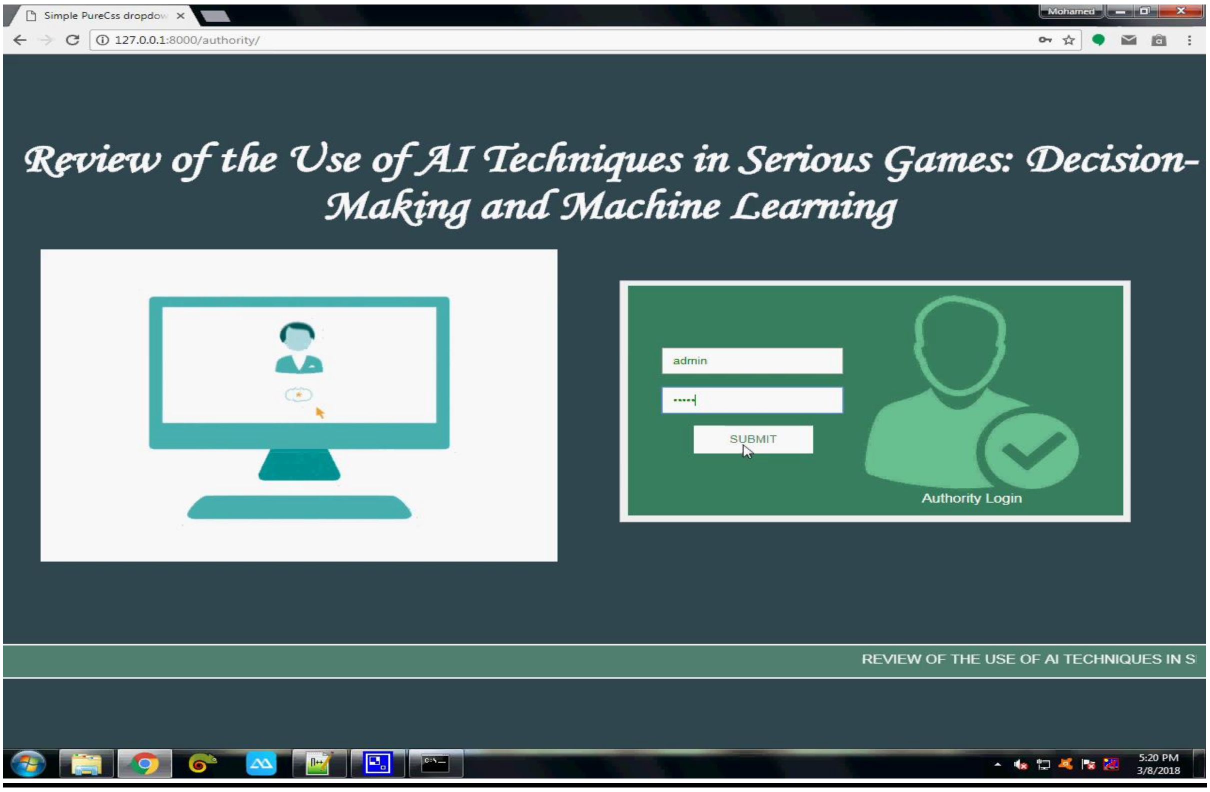
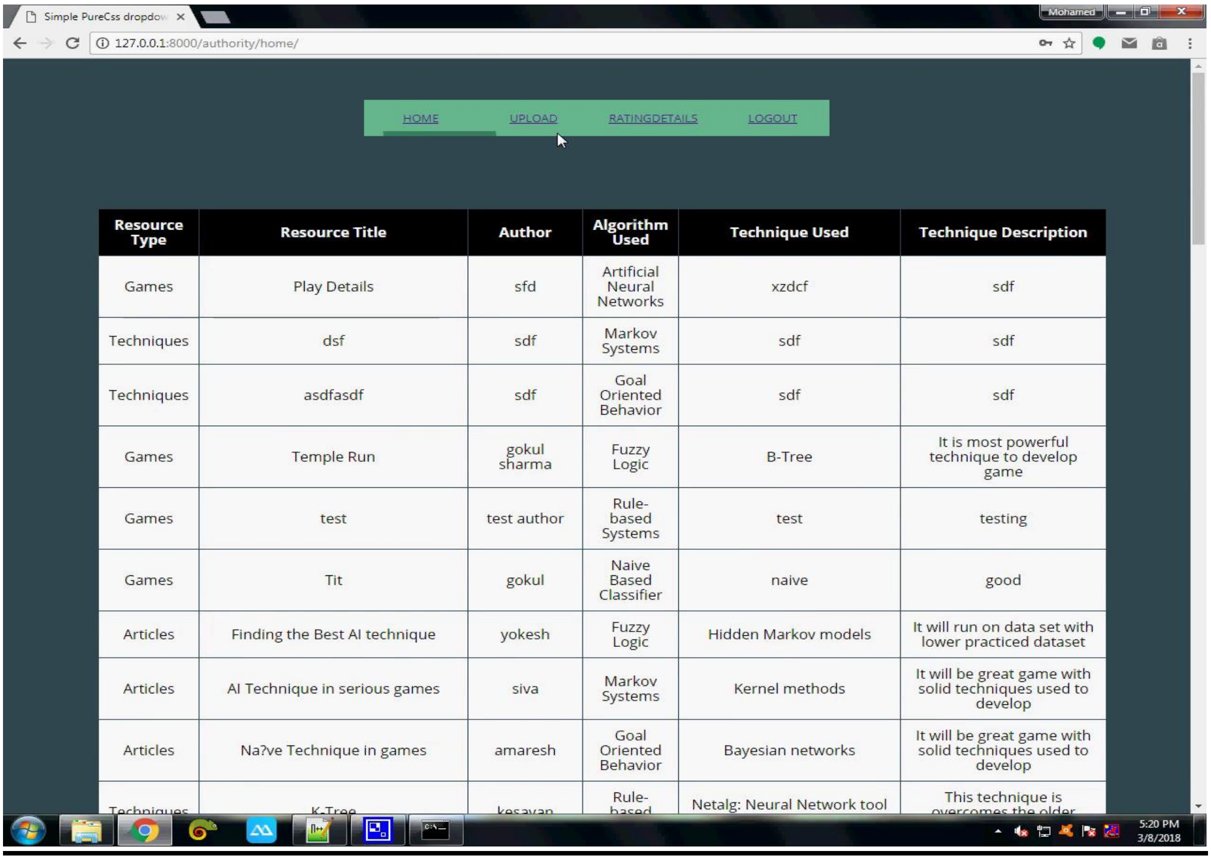
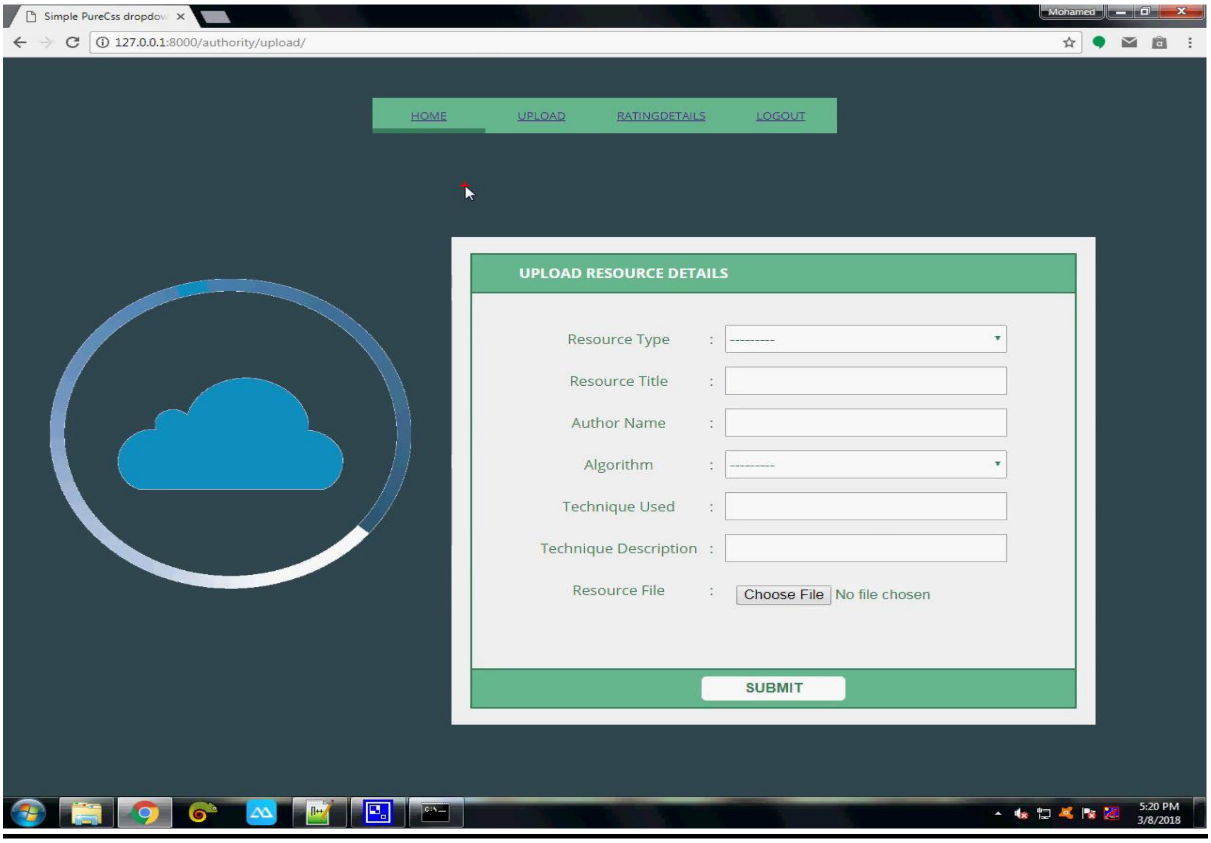

Simple PureCss dropdow
127.0.0.1:8000/authority/
Mohamed
Review of the Use of AI Techniques in Serious Games: Decision-Making and Machine Learning
admin
•••••
SUBMIT
Authority Login
REVIEW OF THE USE OF AI TECHNIQUES IN S
5:20 PM
3/8/2018

Fig:10.2 Authority Login



Resource Type	Resource Title	Author	Algorithm Used	Technique Used	Technique Description
Games	Play Details	sfd	Artificial Neural Networks	xzdcf	sdf
Techniques	dsf	sdf	Markov Systems	sdf	sdf
Techniques	asdfasdf	sdf	Goal Oriented Behavior	sdf	sdf
Games	Temple Run	gokul sharma	Fuzzy Logic	B-Tree	It is most powerful technique to develop game
Games	test	test author	Rule-based Systems	test	testing
Games	Tit	gokul	Naive Based Classifier	naive	good
Articles	Finding the Best AI technique	yokesh	Fuzzy Logic	Hidden Markov models	It will run on data set with lower practiced dataset
Articles	AI Technique in serious games	siva	Markov Systems	Kernel methods	It will be great game with solid techniques used to develop
Articles	Na?ve Technique in games	amaresh	Goal Oriented Behavior	Bayesian networks	It will be great game with solid techniques used to develop
Techniques	K-Tree	kesavan	Rule-based	Netalg: Neural Network tool	This technique is overcomes the older

Fig:10.3 Authority Home Page



UPLOAD RESOURCE DETAILS

Resource Type : ---------

Resource Title :

Author Name :

Algorithm : ---------

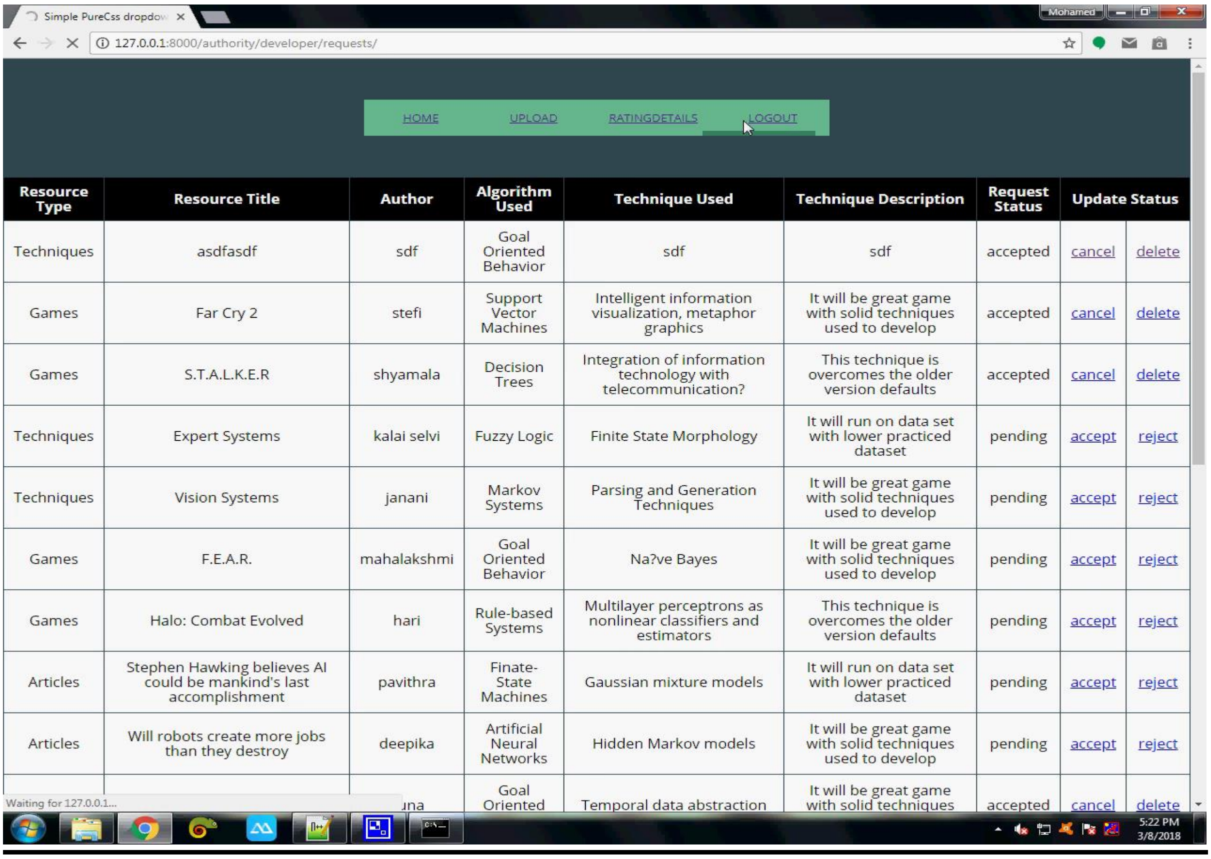
Technique Used :

Technique Description :

Resource File : Choose File No file chosen

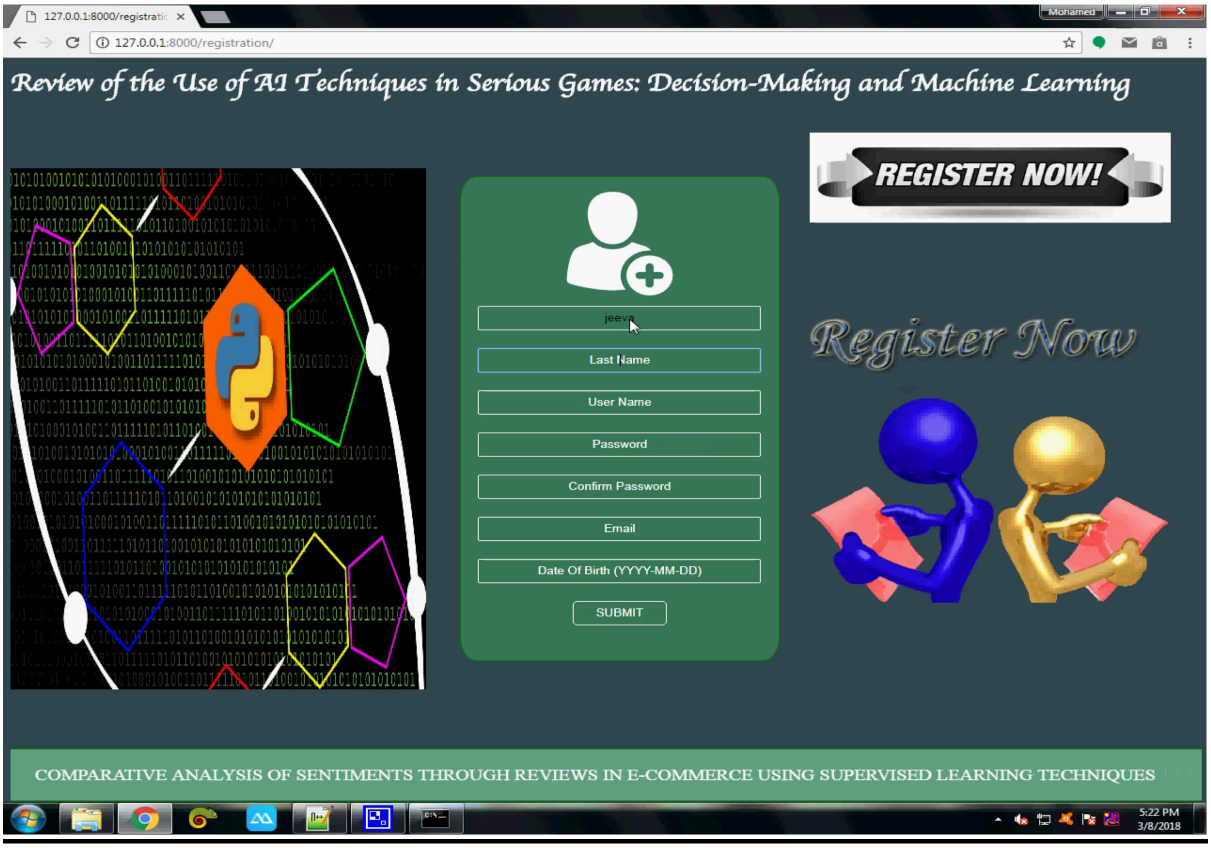
SUBMIT

Fig:10.4 Upload Details



Resource Type	Resource Title	Author	Algorithm Used	Technique Used	Technique Description	Request Status	Update Status	
Techniques	asdfasdf	sdf	Goal Oriented Behavior	sdf	sdf	accepted	cancel	delete
Games	Far Cry 2	stefi	Support Vector Machines	Intelligent information visualization, metaphor graphics	It will be great game with solid techniques used to develop	accepted	cancel	delete
Games	S.T.A.L.K.E.R	shyamala	Decision Trees	Integration of information technology with telecommunication?	This technique is overcomes the older version defaults	accepted	cancel	delete
Techniques	Expert Systems	kalai selvi	Fuzzy Logic	Finite State Morphology	It will run on data set with lower practiced dataset	pending	accept	reject
Techniques	Vision Systems	janani	Markov Systems	Parsing and Generation Techniques	It will be great game with solid techniques used to develop	pending	accept	reject
Games	F.E.A.R.	mahalakshmi	Goal Oriented Behavior	Na?ve Bayes	It will be great game with solid techniques used to develop	pending	accept	reject
Games	Halo: Combat Evolved	hari	Rule-based Systems	Multilayer perceptrons as nonlinear classifiers and estimators	This technique is overcomes the older version defaults	pending	accept	reject
Articles	Stephen Hawking believes AI could be mankind's last accomplishment	pavithra	Finate-State Machines	Gaussian mixture models	It will run on data set with lower practiced dataset	pending	accept	reject
Articles	Will robots create more jobs than they destroy	deepika	Artificial Neural Networks	Hidden Markov models	It will be great game with solid techniques used to develop	pending	accept	reject
		ına	Goal Oriented	Temporal data abstraction	It will be great game with solid techniques	accepted	cancel	delete

Fig:10.5 Developer Request Page



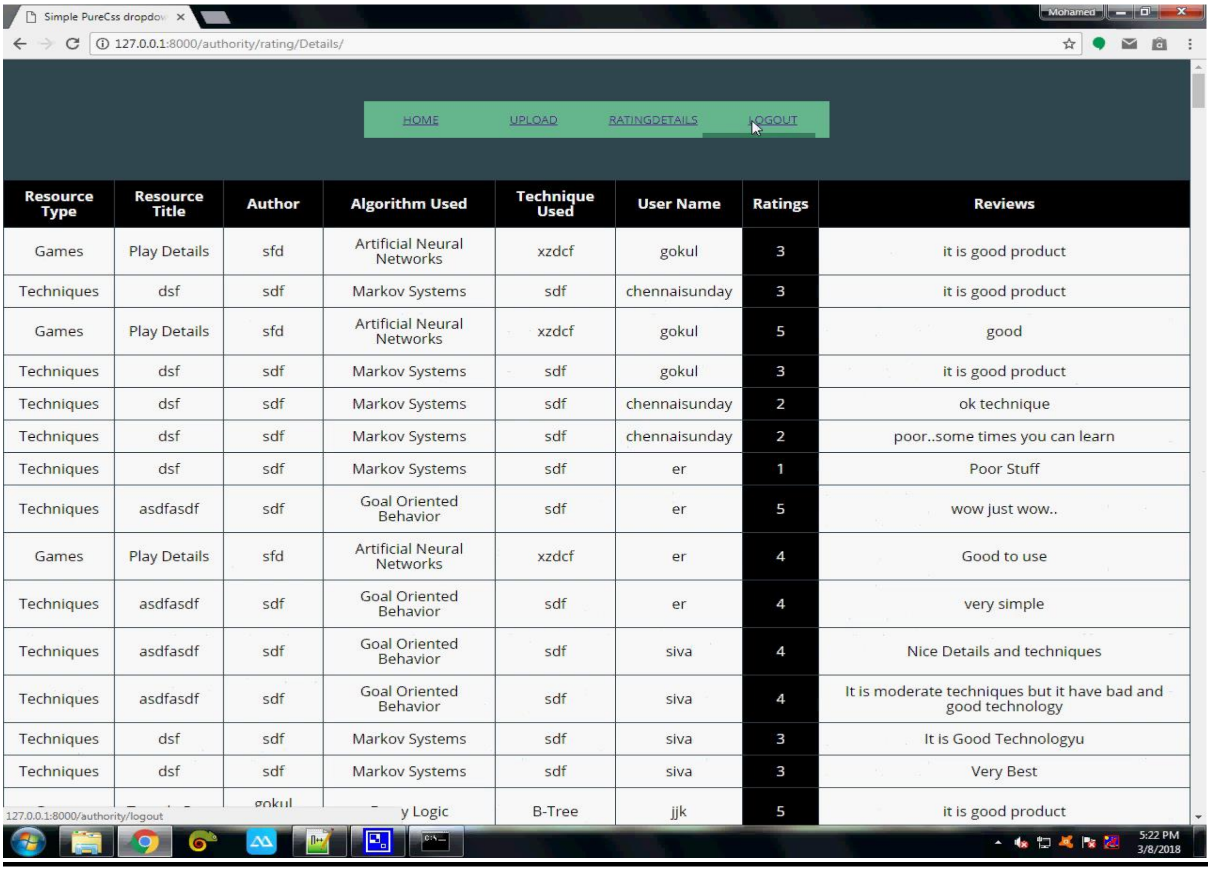
127.0.0.1:8000/registration/

Review of the Use of AI Techniques in Serious Games: Decision-Making and Machine Learning

jeeva

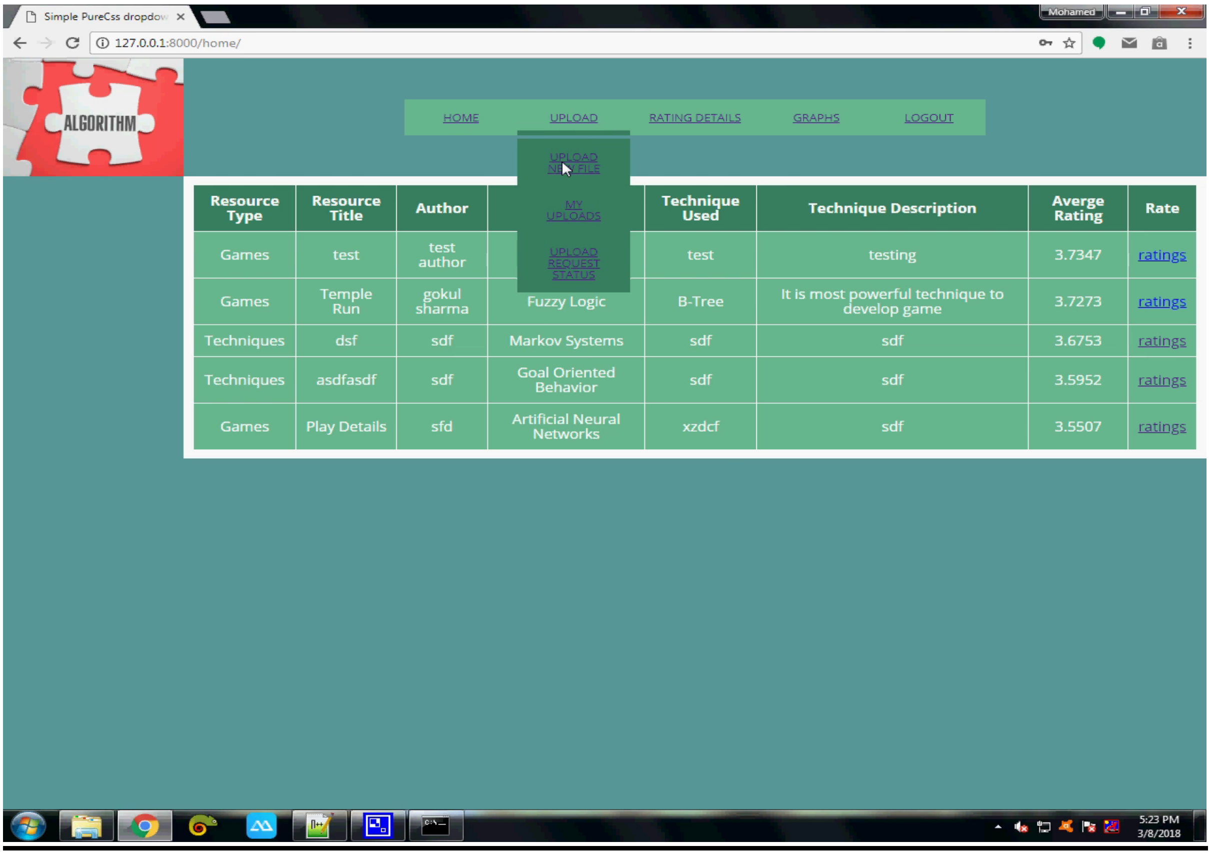
COMPARATIVE ANALYSIS OF SENTIMENTS THROUGH REVIEWS IN E-COMMERCE USING SUPERVISED LEARNING TECHNIQUES

Fig:10.6 Developer Registration



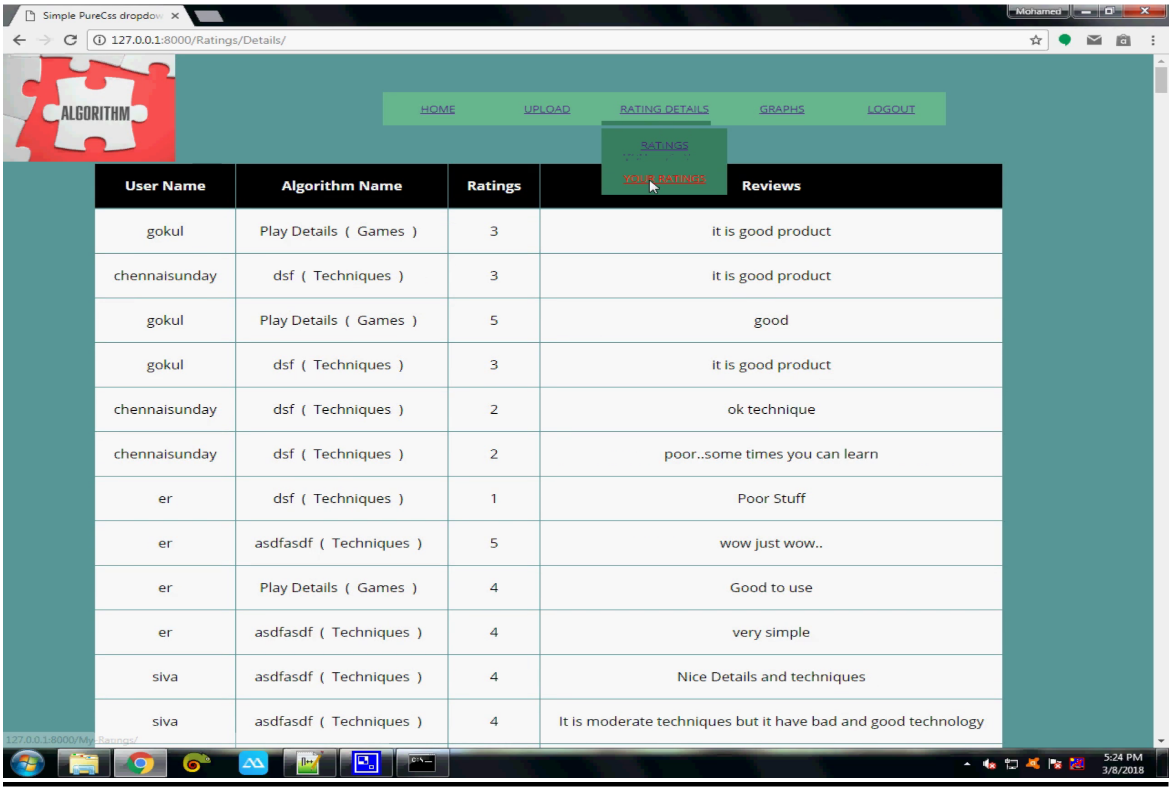
Resource Type	Resource Title	Author	Algorithm Used	Technique Used	User Name	Ratings	Reviews
Games	Play Details	sfd	Artificial Neural Networks	xzdcf	gokul	3	it is good product
Techniques	dsf	sdf	Markov Systems	sdf	chennaisunday	3	it is good product
Games	Play Details	sfd	Artificial Neural Networks	xzdcf	gokul	5	good
Techniques	dsf	sdf	Markov Systems	sdf	gokul	3	it is good product
Techniques	dsf	sdf	Markov Systems	sdf	chennaisunday	2	ok technique
Techniques	dsf	sdf	Markov Systems	sdf	chennaisunday	2	poor..some times you can learn
Techniques	dsf	sdf	Markov Systems	sdf	er	1	Poor Stuff
Techniques	asdfasdf	sdf	Goal Oriented Behavior	sdf	er	5	wow just wow..
Games	Play Details	sfd	Artificial Neural Networks	xzdcf	er	4	Good to use
Techniques	asdfasdf	sdf	Goal Oriented Behavior	sdf	er	4	very simple
Techniques	asdfasdf	sdf	Goal Oriented Behavior	sdf	siva	4	Nice Details and techniques
Techniques	asdfasdf	sdf	Goal Oriented Behavior	sdf	siva	4	It is moderate techniques but it have bad and good technology
Techniques	dsf	sdf	Markov Systems	sdf	siva	3	It is Good Technologyu
Techniques	dsf	sdf	Markov Systems	sdf	siva	3	Very Best
[illegible]	[illegible]	gokul	[illegible] y Logic	B-Tree	jjk	5	it is good product

Fig:10.7 Authority Rating Details



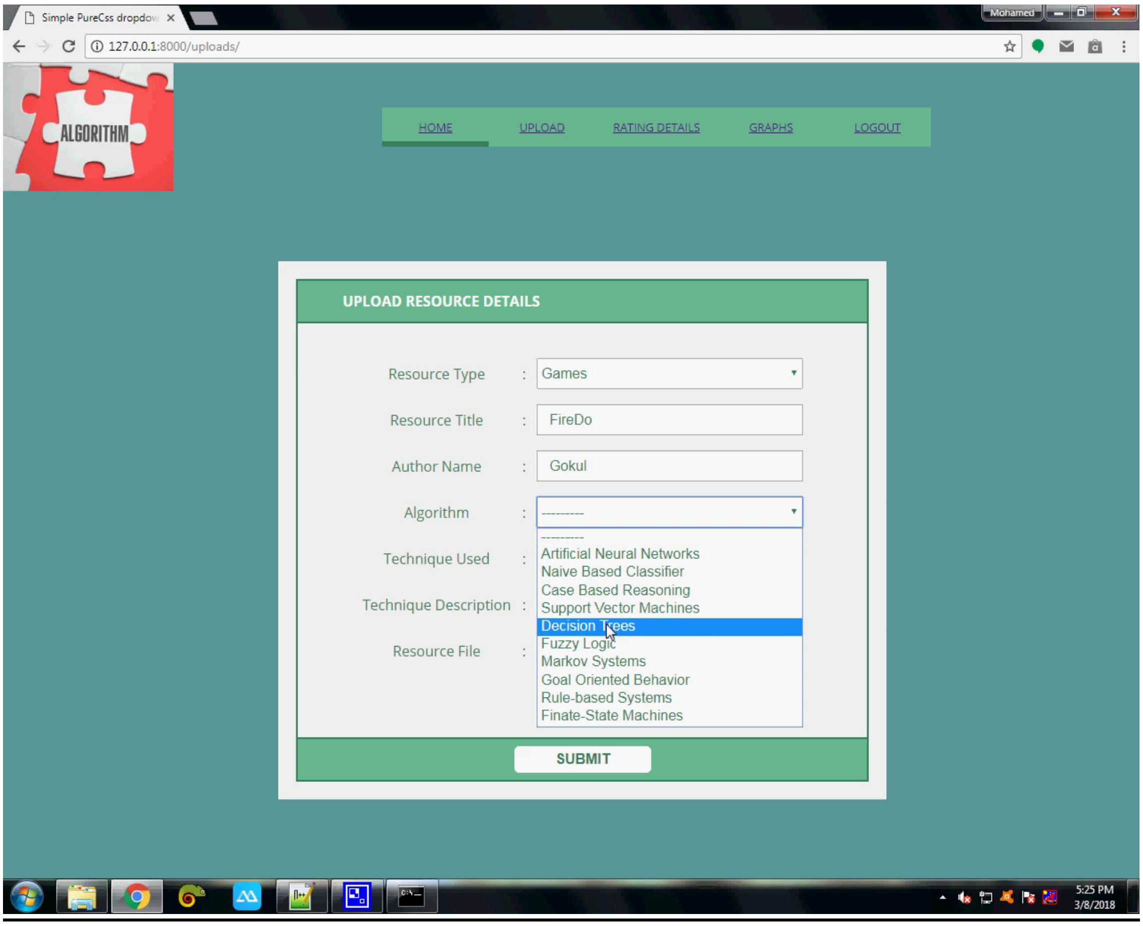
Resource Type	Resource Title	Author		Technique Used	Technique Description	Averge Rating	Rate
Games	test	test author		test	testing	3.7347	ratings
Games	Temple Run	gokul sharma	Fuzzy Logic	B-Tree	It is most powerful technique to develop game	3.7273	ratings
Techniques	dsf	sdf	Markov Systems	sdf	sdf	3.6753	ratings
Techniques	asdfasdf	sdf	Goal Oriented Behavior	sdf	sdf	3.5952	ratings
Games	Play Details	sfd	Artificial Neural Networks	xzdcf	sdf	3.5507	ratings

Fig:10.8 Home Page



User Name	Algorithm Name	Ratings	Reviews
gokul	Play Details (Games)	3	it is good product
chennaisunday	dsf (Techniques)	3	it is good product
gokul	Play Details (Games)	5	good
gokul	dsf (Techniques)	3	it is good product
chennaisunday	dsf (Techniques)	2	ok technique
chennaisunday	dsf (Techniques)	2	poor..some times you can learn
er	dsf (Techniques)	1	Poor Stuff
er	asdfasdf (Techniques)	5	wow just wow..
er	Play Details (Games)	4	Good to use
er	asdfasdf (Techniques)	4	very simple
siva	asdfasdf (Techniques)	4	Nice Details and techniques
siva	asdfasdf (Techniques)	4	It is moderate techniques but it have bad and good technology

Fig:10.9 Rating Details



UPLOAD RESOURCE DETAILS

Resource Type : Games

Resource Title : FireDo

Author Name : Gokul

Algorithm : ---------

Technique Used :

Technique Description :

Resource File :

Artificial Neural Networks
Naive Based Classifier
Case Based Reasoning
Support Vector Machines
Decision Trees
Fuzzy Logic
Markov Systems
Goal Oriented Behavior
Rule-based Systems
Finate-State Machines

Fig:10.10 Upload Resource Details

CHAPTER – 11

CONCLUSION

11. CONCLUSION

Finally, the authors consider that over recent years enough knowledge has been gathered to create new intelligent serious games to consider not only the final aim but also the technologies and techniques used to provide players with a nearly real experience. This new age of serious games is very close to the world of video-games, and they generate new solutions completely adapted to their target audience. However, researchers may need to improve their methodology for testing developed serious games, so as to ensure they meet their final purposes. Moreover, the authors would like to encourage other researchers to extend this article to other AI specific techniques and/or addressing new AI-related features, to extend this state of the art in the field of serious games, creating a knowledge hub for researchers in the area.

CHAPTER – 12
FUTURE ENHANCEMENT

12. FUTURE ENHANCEMENT

The developer interface through the website can be made easier to access through developer profiling.

This can also be developed as a standalone application (engine) which can be used by small game vendors to acquire and attach to their sites.

Sentiment analysis can also be applied to “comments” information to identify the emotion behind the comments (positive, negative or neutral) to recommend games appropriately.

CHAPTER – 13

BIBLIOGRAPHY

13. BIBLIOGRAPHY

1. C. C. Abt, Serious Games, Lanham, MD, USA:Univ. Press Amer., 1987.

2. T. Dumbleton and J. Kirriemuir, "13 digital games and education", *Understanding Digital Games*, pp. 223, 2006.

3. U. Ritterfeld, M. Cody and P. Vorderer, "Serious games: Mechanisms and effects", *Routledge*, 2009.

4. C. Girard, J. Ecalle and A. Magnan, "Serious games as new educational tools: How effective are they? A meta-analysis of recent studies", *J. Comput. Assisted Learn.*, vol. 29, no. 3, pp. 207-219, 2013.

5. J. P. Gee, "What video games have to teach us about learning and literacy", *Comput. Entertain.*, vol. 1, no. 1, pp. 20-20, 2003.

6. F. L. Greitzer, O. A. Kuchar and K. Huston, "Cognitive science implications for enhancing training effectiveness in a serious gaming context", *J. Educ. Resources Comput.*, vol. 7, no. 3, pp. 2, 2007.

7. F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger and R. Berta, "Assessment in and of serious games: An overview", *Adv. Human-Comput. Interaction*, vol. 2013, pp. 1, 2013.

8. B. Sawyer and P. Smith, "Serious games taxonomy", 2008.

9. D. R. Michael and S. L. Chen, Serious Games: Games That Educate Train and Inform, USA:Muska & Lipman/Premier-Trade, 2005.

10. M. Zyda, "From visual simulation to virtual reality to games", *Computer*, vol. 38, no. 9, pp. 25-32, 2005.

11. J. Alvarez and L. Michaud, "Serious games. Advergaming edugaming training and more", *Proc. IDATE*, 2008.

12. R. Ratan and U. Ritterfeld, "Classifying serious games", *Serious Games: Mechanisms and Effects*, pp. 10-24, 2009.

13. A. J. Stapleton, "Serious games: Serious opportunities", *Proc. Austral. Game Develop. Conf.*, 2004.