A PROJECT REPORT ON

EXERCISE RECOMMENDATION METHOD BASED ON MACHINE LEARNING

Project report submitted in partial fulfillment of the requirement for the award of the degree of

DIPLOMA

By

SK KASIM SAIDA	22333-CM-054
CHINNAPOGU GOPI	22333-CM-011
THOTA LAKSHMI CHARAN	22333-CM-062
KOTCHIRI KARTHIK	22333-CM-032
GARIPATI SATIEESH	22333-CM-019
CH SRINIVASA KUMAR	22333-CM-012
M CHIRANJIVI	22333-CM-42

Under the Esteemed guidance of

MR.V.V.B CHARI MTECH
Asst.Proffesor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A.M REDDY MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY
(Affiliated to Jawaharlal Nehru Technological University, Kakinada)
NARASARAOPET

2022-2025

A.M REDDY MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada)
Vinukonda Road, Petlurivaripalem(V), Palnadu District – 522 601, A.P, India

\

CERTIFICATE

This is to certify that the project report entitles **"EXERCISE RECOMMENDATION METHOD BASED ON MACHINE LEARNING "** being submitted by

CHINNAPOGU GOPI	22333-CM-011
THOTA LAKSHMI CHARAN	22333-CM-062
KOTCHIRI KARTHIK	22333-CM-032
GARIPATI SATIEESH	22333-CM-019
CH SRINIVASA KUMAR	22333-CM-012
M CHIRANJIVI	22333-CM-42

In partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University is a record of Bonafede work carried out by him under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

GUIDANCE

MR.V.V.B CHARI MTECH
Asst.Proffesor

HEAD OF DEPARTMENT

Mr. V.V.B CHARI M.Tech
Asst.Proffesor

EXTERNAL EXAMINE

Viva voice held on: _____

PRINCIPAL

Mr. KRISHNA REDDY M. TECH, Ph.D.,

A.M REDDY MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY

Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada, Vinukonda Road,
Petlurivaripalem(V), Palnadu District – 522 601, A.P, India

DECLARATION

We hereby declare that the Project report titled **EXERCISE RECOMMENDATION METHOD BASED ON MACHINE LEARNING**

SK KASIM SAIDA	22333-CM-054
CHINNAPOGU GOPI	22333-CM-011
THOTA LAKSHMI CHARAN	22333-CM-062
KOTCHIRI KARTHIK	22333-CM-032
GARIPATI SATIEESH	22333-CM-019
CH SRINIVASA KUMAR	22333-CM-012
M CHIRANJIVI	22333-CM-42

Is bona fide work done by us, under the guidance of **Mr. V.V.B CHARI, Asst. Professor, HOD, A.M. REDDY MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY.** This report is submitted in partial fulfillment of the requirements for the award of the degree DIPLOMA in **CSE.** We also declare that this project is a result of our own effort and that has not been copied from anyone and we have taken only citations from the sources which are mentioned in the references.

This work was not submitted earlier at any other University or Institute for the award of any degree.

Place: Narasaraopet.

Date**:**

ACKNOWLEDGEMENT

We are indebted to the Management for providing us an opportunity with excellent academic, infrastructural Lab facilities to carry out our U.G program successfully.

We would like to express our deepest sense to providing of gratitude towards my Internal Guide, **MR.V.V.B CHARI** **Asst. Professor** in **Department of Computer** of **Computer Science and Engineering** for her encouragement and valuable guidance in bringing shape to this dissertation. They are grateful **to** Doing a Project **, Mr. V.V.B CHARI M.Tech., HOD Department of Computer Science** & Engineering, **A.M. REDDY MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY**, **Narasaraopet** his encouragement and motivation.

We express our deep gratitude regard to our beloved principal **Dr. KRISHNA REDDY, M. TECH, Ph.D.**, who have given support for team members to doing the project or in other aspects of our studies at college, **AM REDDY MEMORIAL COLLEGE OF ENGINEERING &TECHNOLOGY,** Petlurivaripalem, Narasaraopet, and affiliated JNTUK Kakinada.

We are thankful to all the Faculty Members in the department for their teachings and academic support and thanks to Technical Staff and Non-teaching staff in the department.

At last, but not least we thank all my well-wishers for rendering necessary support during the execution of this work.

CHINNAPOGU GOPI	22333-CM-011
THOTA LAKSHMI CHARAN	**22333-CM-062**
KOTCHIRI KARTHIK	**22333-CM-032**
GARIPATI SATIEESH	**22333-CM-019**
M CHIRANJIVI	**22333-CM-42**
SK KASIM SAIDA	22333-CM-054

INDEX

CONTENTS

LIST OF FIGURES

ABSTRACT

It proposes a recommender system (RS) to support the fitness assistance system (FAS) with artificial intelligence. The RS is applied to make these suggestions for the beginners and existing users. The goal of the paper aims to develop an RS that has an ability to learn, analyze, predict, and make these suggestions as well as communicate to human through AI. Artificial Neural Network and Logistic Regression have been employed to predict the suitable workout for each beginner. In addition, the agent developed with reinforcement learning capability of Soar architecture help the members select their workout based on their condition. Through the experimental result, the effectiveness of utility application is validated.

INTRODUCTION

INTRODUCTION

The RS is known as a part of information filtering system which helps the users seek the prediction of rating or preference that users would give to an item or service recommendations. Currently, the RS has been upgraded with the several machine learning algorithms to provide users with the suggestion for their purposes in or build the framework for RS as shown in. In the fitness field, recent studies have focused on developing the RS to user with a wearable device and recording data in real-time. A fitness assistant framework is developed to smartly track and identify user's activity based on contextual interpretation in Moreover, RS has been approached for a runner, which is described in . The purpose of this study is to design the RS that will suggest personalized workout to the users and predict the plan for doing exercise in future. In the proposed RS, we use machine learning algorithms on activity data to build a predictive module in the basic training layer (BTL) that classify the user's activity in their workout. In addition, we also build the trainer agent (TA) with Soar architecture and machine learning algorithm to reflect the prediction of BTL for suggesting the several workouts to help users select the suitable workout fitting well with their exercise plan.

.

CHAPTER-1
LITERATURE SURVEY

LITERATURE SURVEY

1.Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions

AUTHORS: G. Adomavicius, A.

It presents an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. This paper also describes various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multicriteria ratings, and a provision of more flexible and less intrusive types of recommendations.

2. A general framework for intelligent recommender systems

AUTHORS: J. Aguilar, P. Valdiviezo-Díaz, and G. Rio Frio

It presents an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. This paper also describes various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multicriteria ratings, and a provision of more flexible and less intrusive types of recommendations.

3. Adaptive ε-Greedy Exploration in Reinforcement Learning Based on Value Differences

AUTHORS: **Yuriy V. Lonchakov** , **M. Tokic**

It presents “Value-Difference Based Exp loration” (VDBE), a method for balanceing the exploration/exploitation dilemma inherent to reinforcement learning. The proposed method adapts the exploration parameter of ε-greedy in dependence of the *temporal-difference error* observed from value-function backups, which is considered as a measure of the agent’s uncertainty about the environment. VDBE is evaluated on a multi-armed bandit task, which allows for insight into the behavior of the method. Preliminary results indicate that VDBE seems to be more parameter robust than commonly used ad hoc approaches such as ε-greedy or SoftMax.

4 .Personalization of wellness recommendation using contextual interpertation

AUTHORS: M. Afzal, S. I. Ali, R. Ali, M. Hussain, T. Ali, W. A. Khan, M. B. Amin, B. H. Kang, and S. Y. Le

A huge array of personalized healthcare and wellness systems are introduced into the portfolio of digital health and quantified-self movement in recent years. These systems share common capabilities including self-tracking/monitoring and self-quantifications, based on the raw sensory data. These capabilities provide solid ground for the users to be more aware of their health; however, such measures are inefficient for changing the unhealthy habits of the users. In order to induce healthy habits in the users, a system must be capable of generating context-aware personalized recommendations. The main obstacle in this regard is the contextual interpretation of recommendations based on user's current context and contextual preferences. To resolve these issues, we propose a methodology of cross-context interpretation of recommendations (CCIR) for personalized health and wellness services. The CCIR method adds additional capabilities to the traditional reasoning methods and builds advanced form of the reasoning with the incorporation of contextual factors in the process of interpretations of the recommendations.

CHAPTER-2
SYSTEM ANALYSIS

SYSTEM ANALYSIS

EXISTING SYSTEM:

In the existing recommender systems and describes recommendation methods that are usually classified into the following three main categories content-based, collaborative, and hybrid recommendation approaches. The various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multicriteria ratings, and a provision of more flexible and less intrusive types of recommendations.

DISADVANTAGES OF EXISTING SYSTEM:

- Each element of the user space C can be defined with a profile that includes various user characteristics, such as age, gender, income, marital status, etc. In the simplest case, the profile can contain only a single (unique) element, such as User ID
- The content-based approach to recommendation has its roots in information retrieval, and information filtering research. Because of the significant and early advancements made by the information retrieval and filtering communities and because of the importance of several text-based applications.

PROPOSED SYSTEM

The RS is known as a part of information filtering system which helps the users seek the prediction of rating or preference that users would give to an item or service recommendations. Currently, the RS has been upgraded with the several machine learning algorithms to provide users with the suggestion for their purposes in or build the framework. We use machine learning algorithms on activity data to build a predictive module in the basic training layer (BTL) that classify the user's activity in their workout. In addition, we also build the trainer agent (TA) with Soar architecture and machine learning algorithm to reflect the prediction of BTL for suggesting the several workouts to help users select the suitable workout fitting well with their exercise plan.

ADVANTAGES OF PROPOSED SYSTEM

- The FAS is the system designed to support users doing exercise with fitness assistance equipment (FAE) used to support lifting the weight of exercise instead of the traditional method.
- The proposed RS used in FAS is a system combined with Artificial Intelligence (AI) packages, which plays a role as a professional trainer to give the training instructions of workout for users based on predictability and data analysis to provide the appropriate suggestions according to user's conditio

CHAPTER-3
SYSTEM DESIGN

3.1.ARCHITECTURE

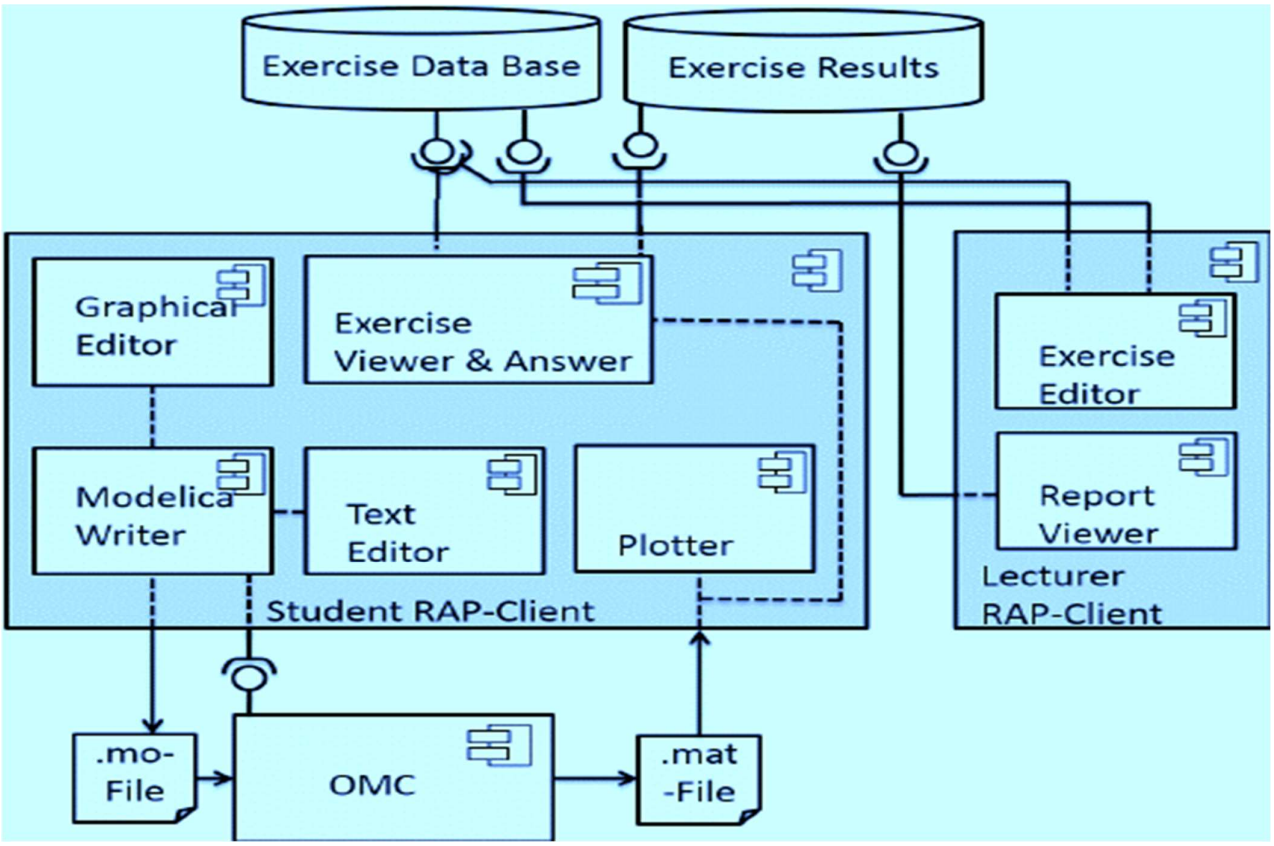


Figure: 3.1. The Block Diagram

3.2. USE CASE DIAGRAM

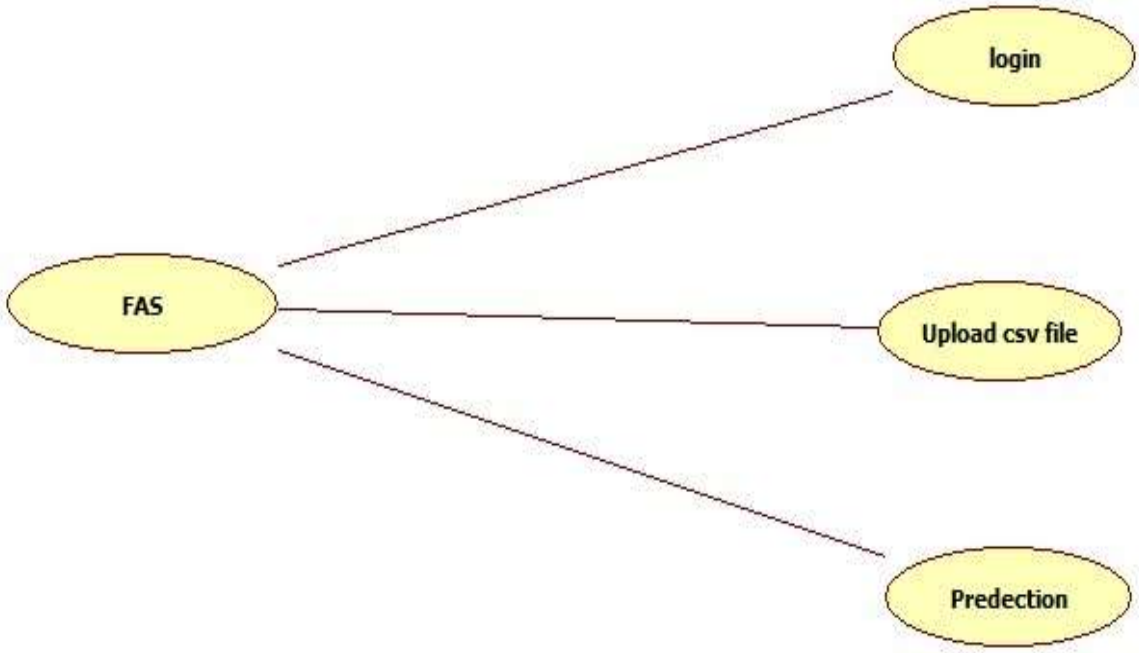


Figure 3.2.1: The Use Case Diagram for FAS

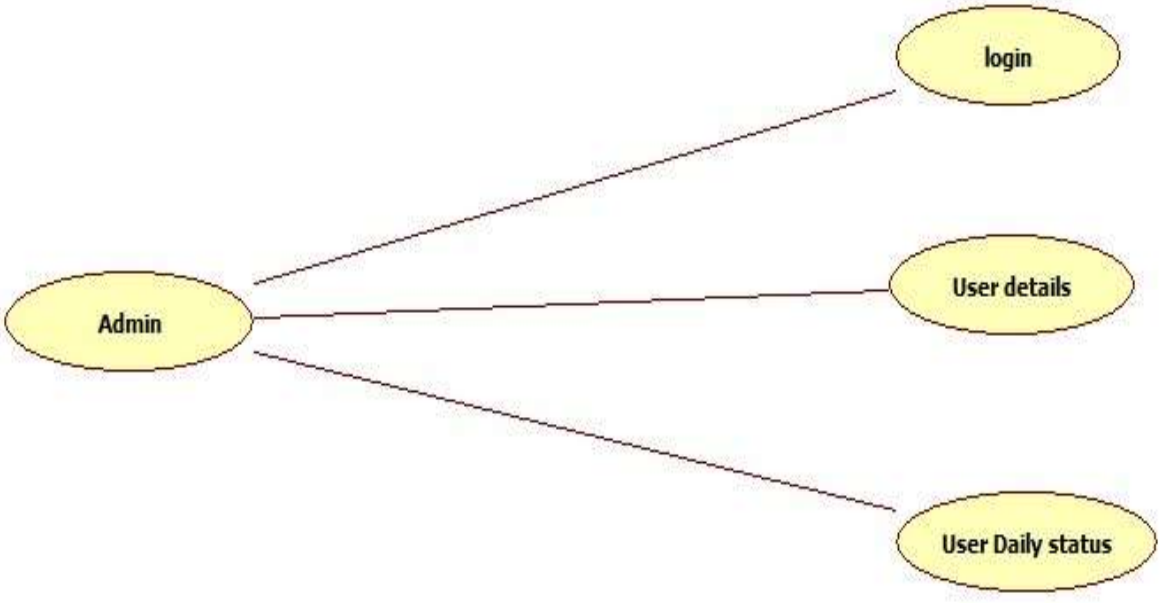


Figure 3.2.2: Use Case Diagram for ADMIN

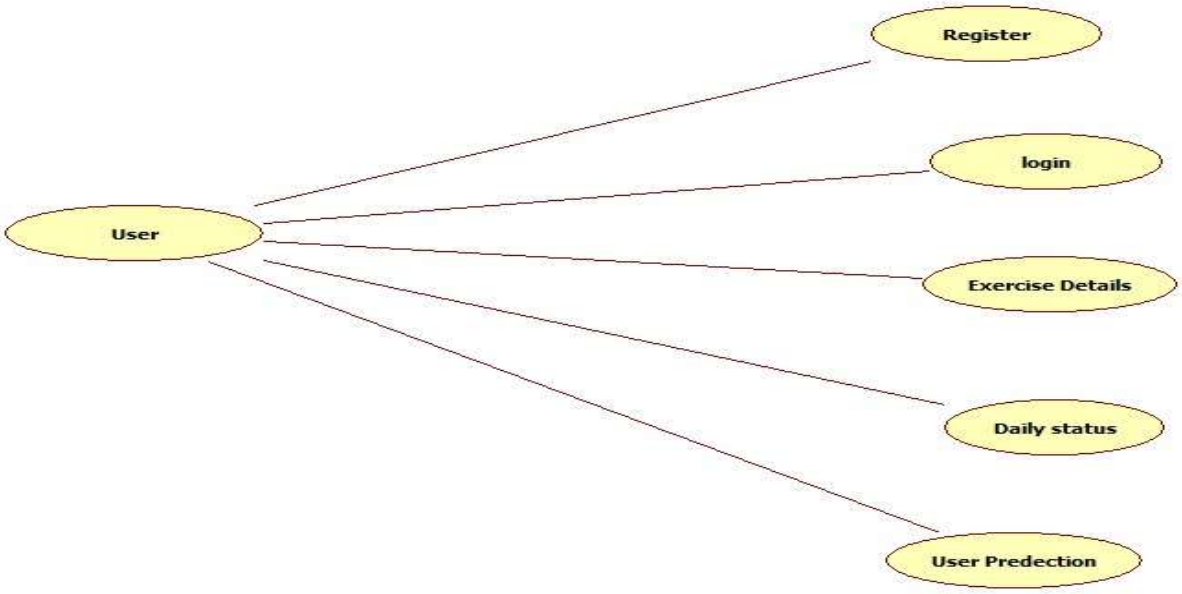


Figure 3.2.3. The Use Case Diagram for User

3.3 CLASS DIAGRAM

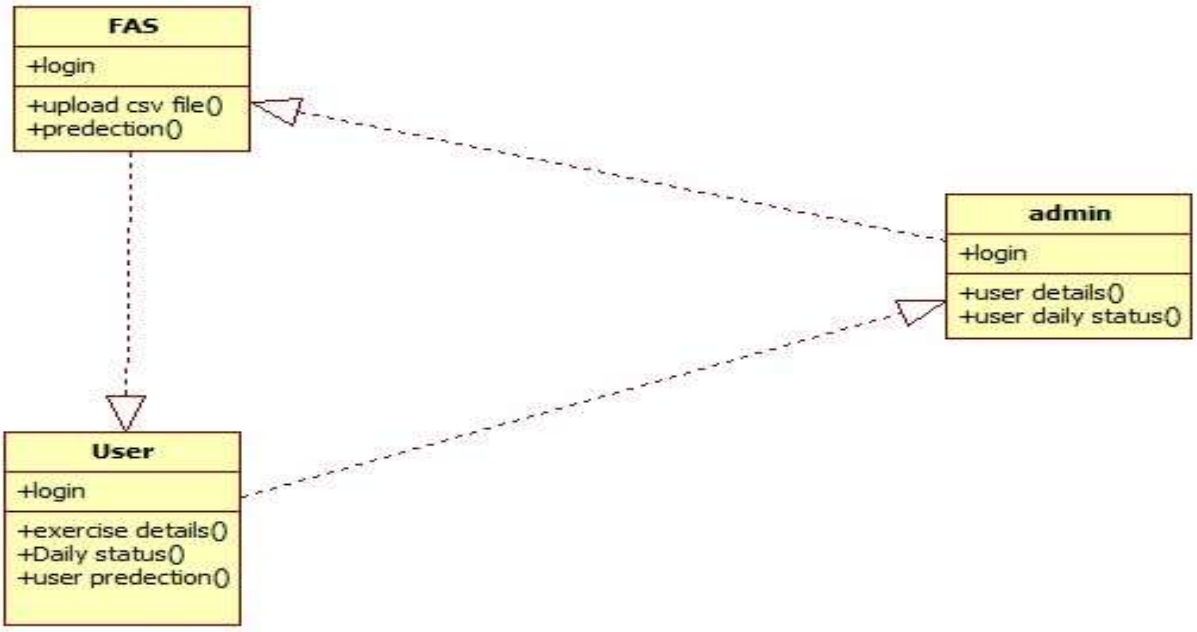


Figure 3.3.1: The Class Diagram

3.4. SEQUENCE DIAGRAM

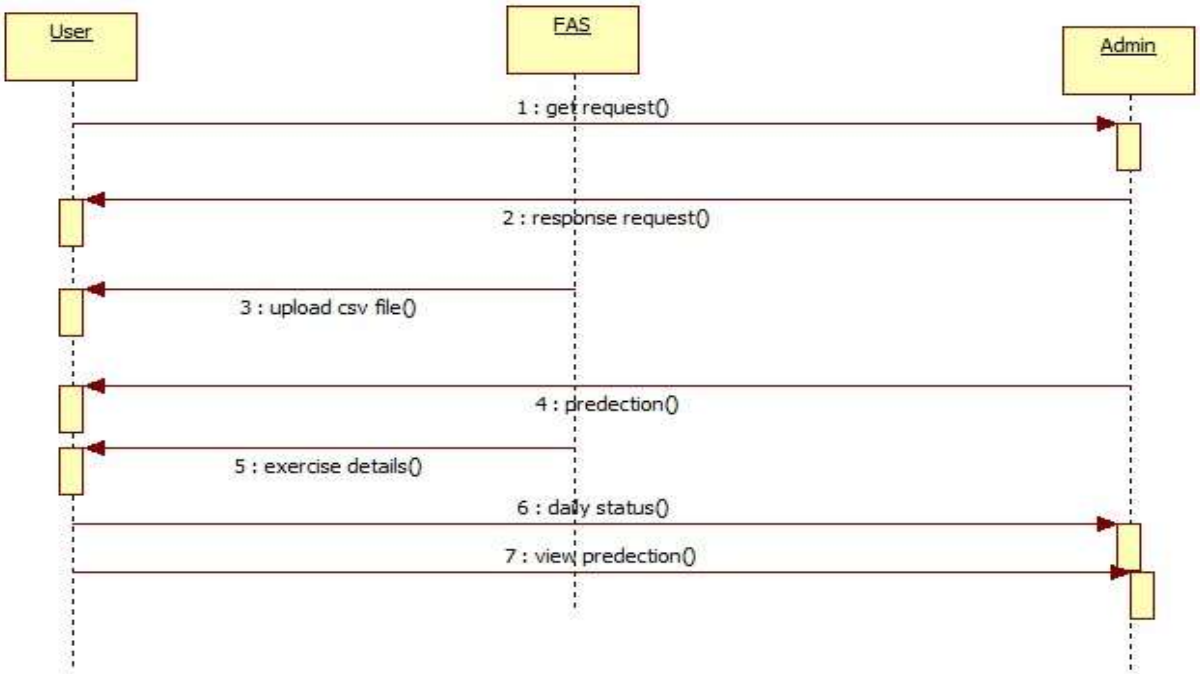


Figure 3.4.1: The Sequence Diagram

3.5. ACTIVITY DIAGRAM

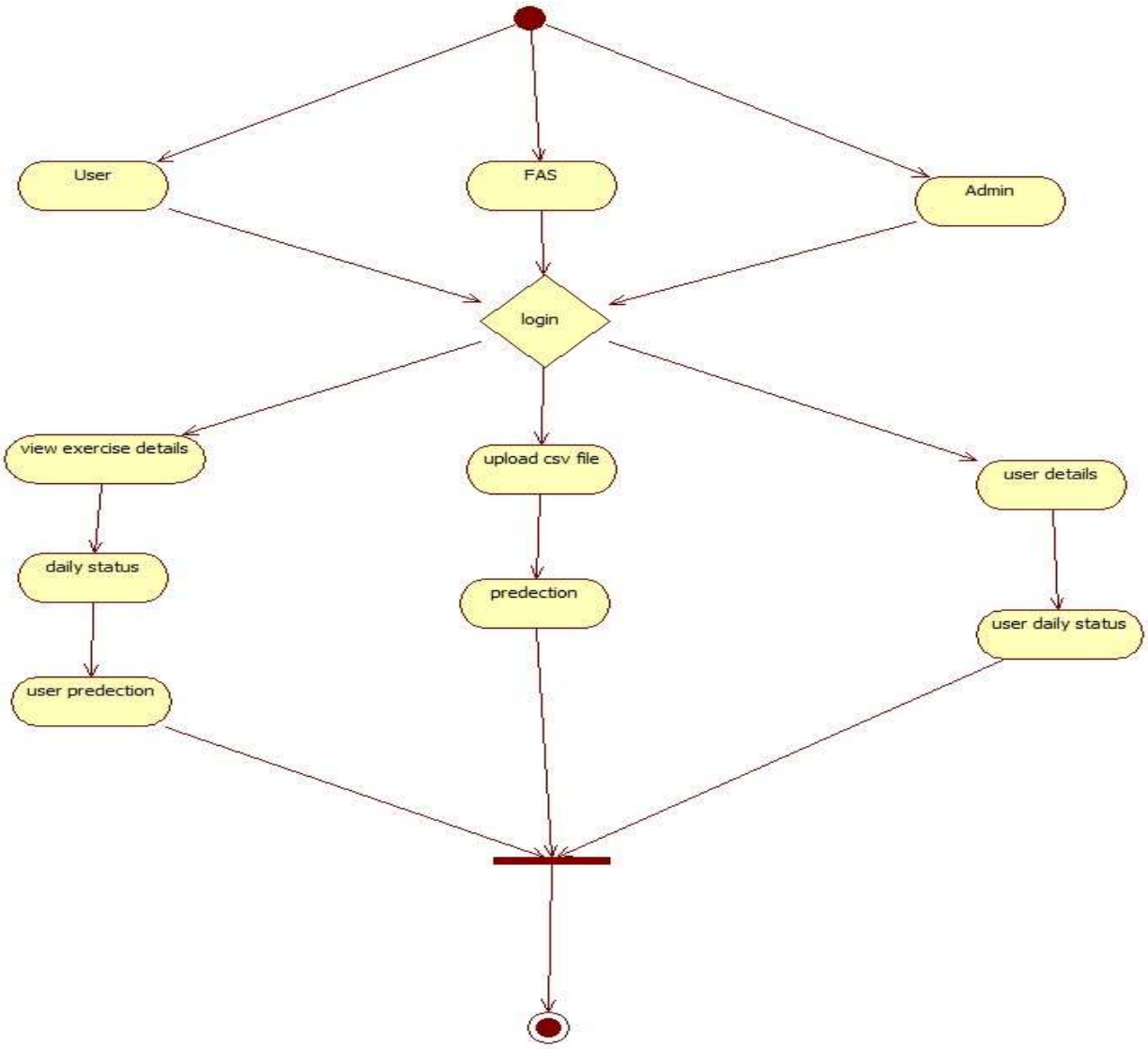


Figure 3.5.1: The Activity Diagram

CHAPTER-4
MODULES

MODULES:

- FAS
- User
- Admin

FAS

The FAS is the system designed to support users doing exercise with two motors (called fitness assistance equipment, FAE) used to support lifting the weight of exercise instead of the traditional method. In FAS, the proposed RS is added top redict appropriate suggestions for users and transfer a control command to embedded controller conducting the FAE. The proposed RS used in FAS is a system combined with artificial intelligence (AI) packages, which plays a role.

Professional trainer to give the training instructions of work out for users based on predictability and data analysis to provide the appropriate suggestions according to user's condition. Machine learning algorithms help RS improve the ability of learning, identifying and acquiring knowledge from the real workout data. Particularly, it supports FAS to perform the simulation of exercise for each user's requirements.

User

The RS is known as a part of information filtering system which helps the users seek the prediction of rating or preference that users would give to an item or service recommendations. Currently, the RS has been upgraded with the several machine learning algorithms to provide users with the suggestion for their purposes in or build the framework for RS as shown in. In the fitness field, recent

Studies have focused on developing the RS to user with a wearable device and recording data in real-time. A fitness assistant framework is developed to smartly track and identify user's activity based on contextual interpretation in. Moreover, RS has been approached for a runner, which is described in the purpose of this study is to design the RS that will suggest personalized workout to the users and predict the plan for doing exercise in future.

In the proposed RS, we use machine learning algorithms on activity data to build a predictive module in the basic training layer (BTL) that classify the user's activity in their workout. In addition, we also build the trainer agent (TA) with Soar architecture and machine learning algorithm to reflect the prediction of BTL for suggesting the several workouts to help users select the suitable workout fitting well with their exercise plan.

Admin

The aim of admin is to approve the users. the entire data must be gathered to admin. Admin maintain the all registered user information's and admin should maintain the users daily status reports.

The purpose of this is to design the RS that will suggest personalized workout to the users and predict the plan for doing exercise in future. In the proposed RS, we use machine learning algorithms on activity data to build a predictive module in the basic training layer (BTL) that classify the user's activity in their workout. In addition, we also build the trainer agent (TA) with Soar architecture and machine learning algorithm to reflect the prediction of BTL for suggesting the several workouts to help users select the suitable workout fitting well with their exercise plan.

CHAPTER-5
ALGORITHM

ALGORITHMS

a. ARTIFICIAL NEURAL NETWORKS

b. LOGISTIC REGRESSION

c. REINFORCEMENT LEARNING

5. 1. ARTIFICIAL NEURAL NETWORKS ALGORITHM

Artificial Neural Network Tutorial provides basic and advanced concepts of ANNs. Our Artificial Neural Network tutorial is developed for beginners as well as professions.

The term "Artificial neural network" refers to a biologically inspired sub-field of artificial intelligence modelled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.

Artificial neural network tutorial covers all the aspects related to the artificial neural network. In this tutorial, we will discuss ANNs, Adaptive resonance theory, Kohonen self-organizing map, Building blocks, unsupervised learning, Genetic algorithm, etc

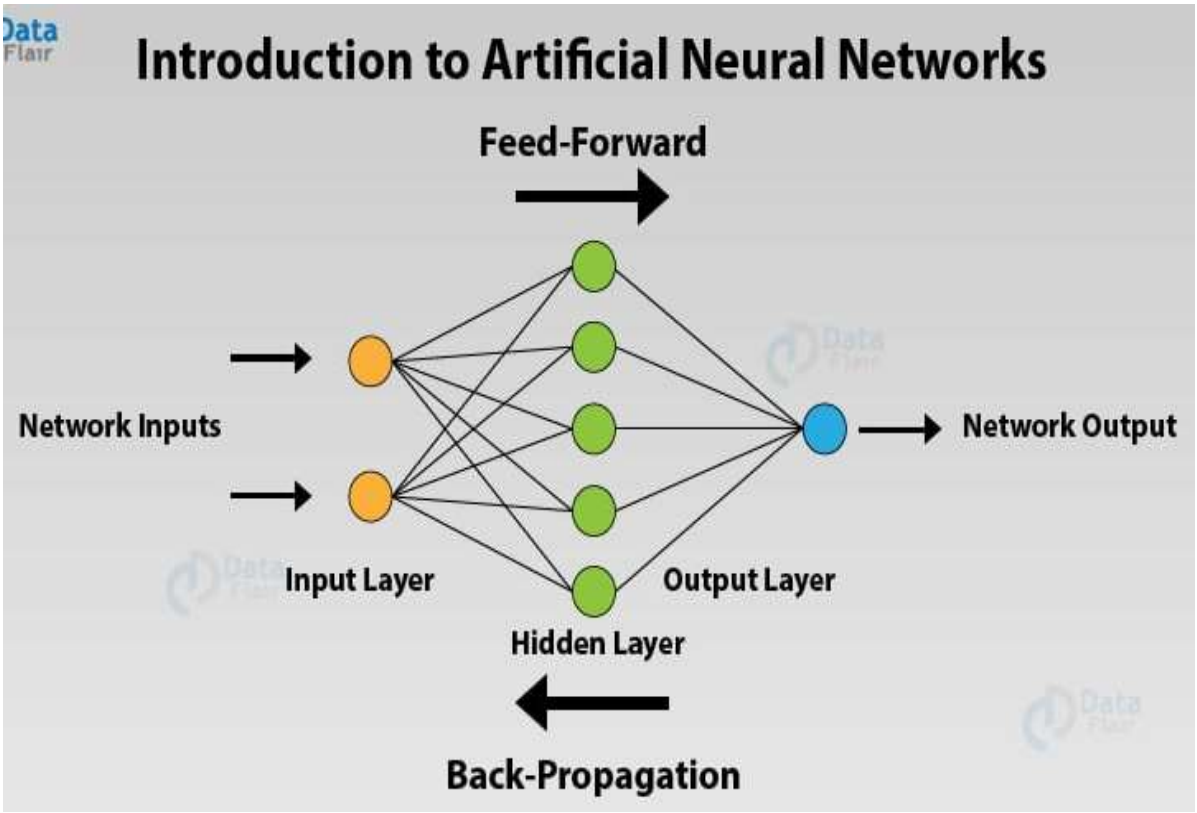


Figure 5.1.1. The Artificial Neural Networks Algorithm

5.2. LOGISTIC REGRESSION ALGORITHM

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

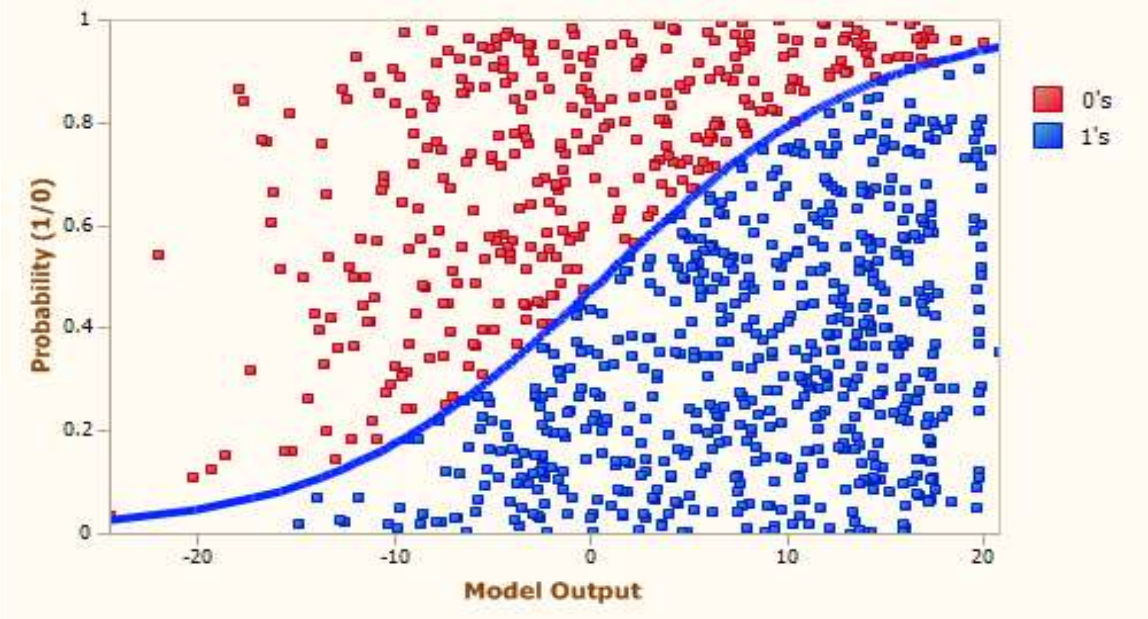


Figure 5.2.1. The Regression Graph

5.3. REINFORCEMENT LEARNING ALGORITHM

- Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.
- In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike **supervised learning**.
- Since there is no labeled data, so the agent is bound to learn by its experience only.
- RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as **game-playing, robotics**, etc.
- The agent interacts with the environment and explores it by itself. The primary goal of an agent in reinforcement learning is to improve the performance by getting the maximum positive rewards.
- The agent learns with the process of hit and trial, and based on the experience, it learns to perform the task in a better way. Hence, we can say that **"Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that."** How a Robotic dog learns the movement of his arms is an example of Reinforcement learning.

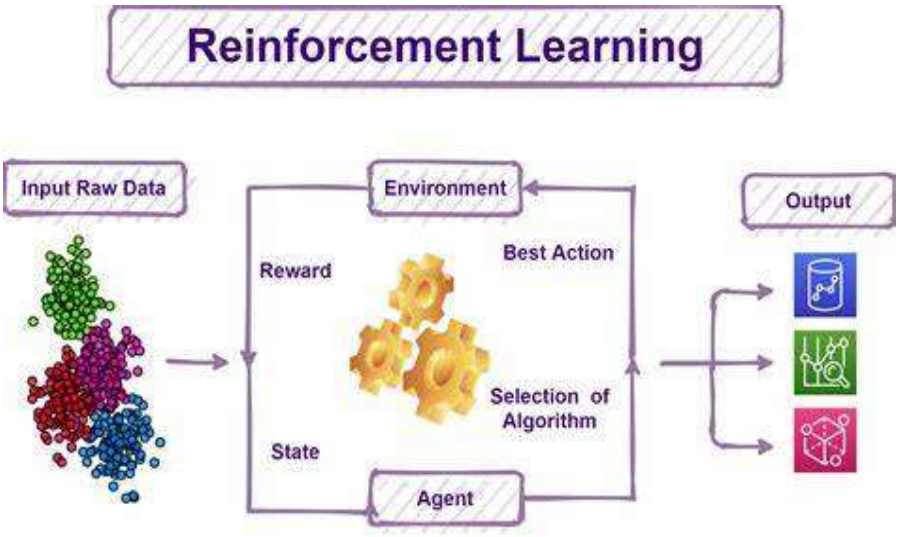


Figure 5.3.1. The Reinforcement learning.

CHAPTER-6
SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE

The structure of RS employed in FAS is illustrated. In order to build the RS with AI, some machine learning Recommender System with Artificial Intelligence for Fitness Assistance System* IEEE 2018 15th International Conference on Ubiquitous Robots (UR) Hawaii Convention Center, Hawai'i, USA, algorithms have been applied to predict and give the workout recommendation. the structure of RS is composed of two modules: basic training layer (BTL) and trainer agent (TA), where BTL is built with Artificial Neural Network (ANN) and Logistic Regression (LR). Data classification is the core component of this module. In the current implementation, the main task of this module aims to predict and give the suggestions of workout for beginners based on their initial information.

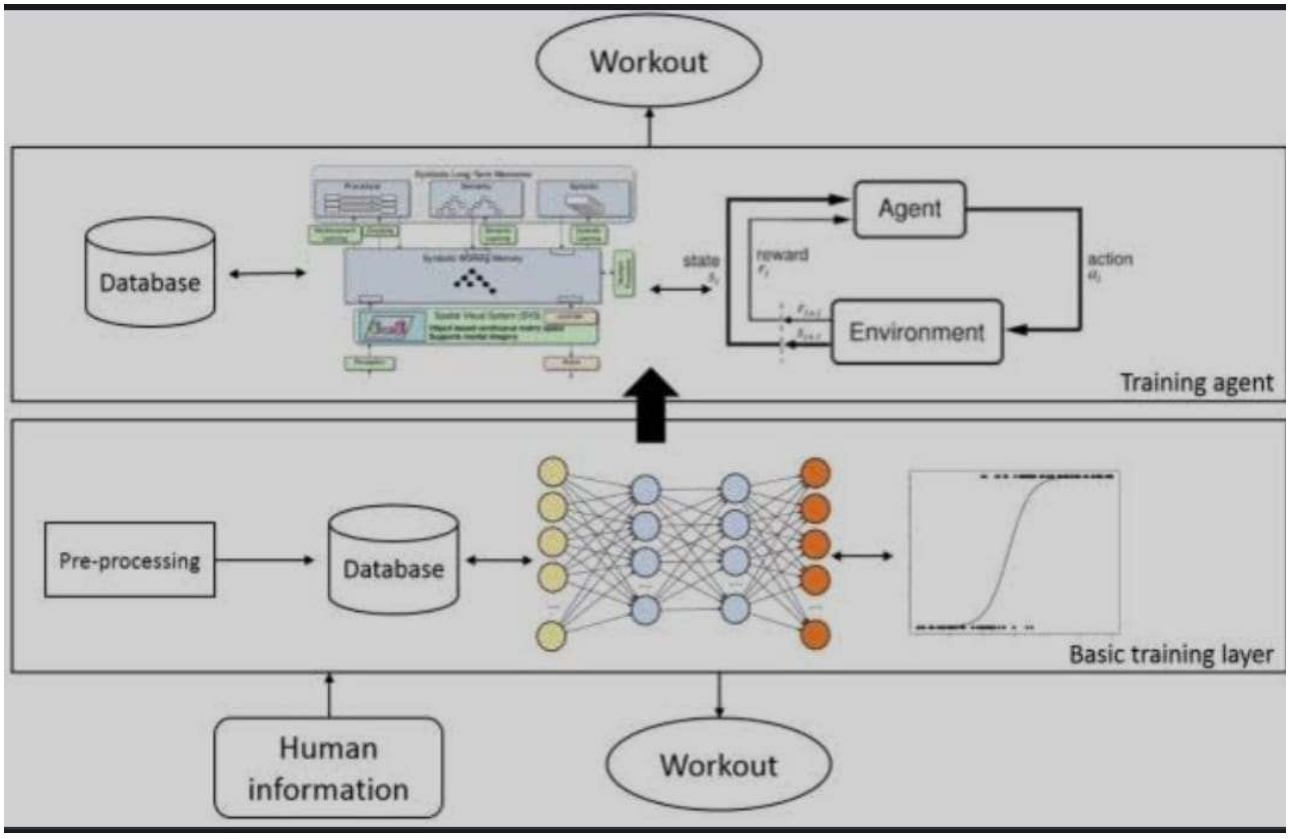


Figure 6.1. The System Architecture

CHAPTER-7
IMPLEMENTATION

IMPLEMENTATION

REQUIREMENT ANALYSIS:

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION

Functional Requirements

- Graphical User interface with the User.

Software Requirements

For developing the application, the following are the Software Requirements:

1. Python
2. Django

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application, the following are the Hardware Requirements:

- Processor: Pentium IV or higher
- RAM: 256 MB
- Space on Hard Disk: minimum 512MB

SOFTWARE REQUIREMENTS:

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. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python 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.

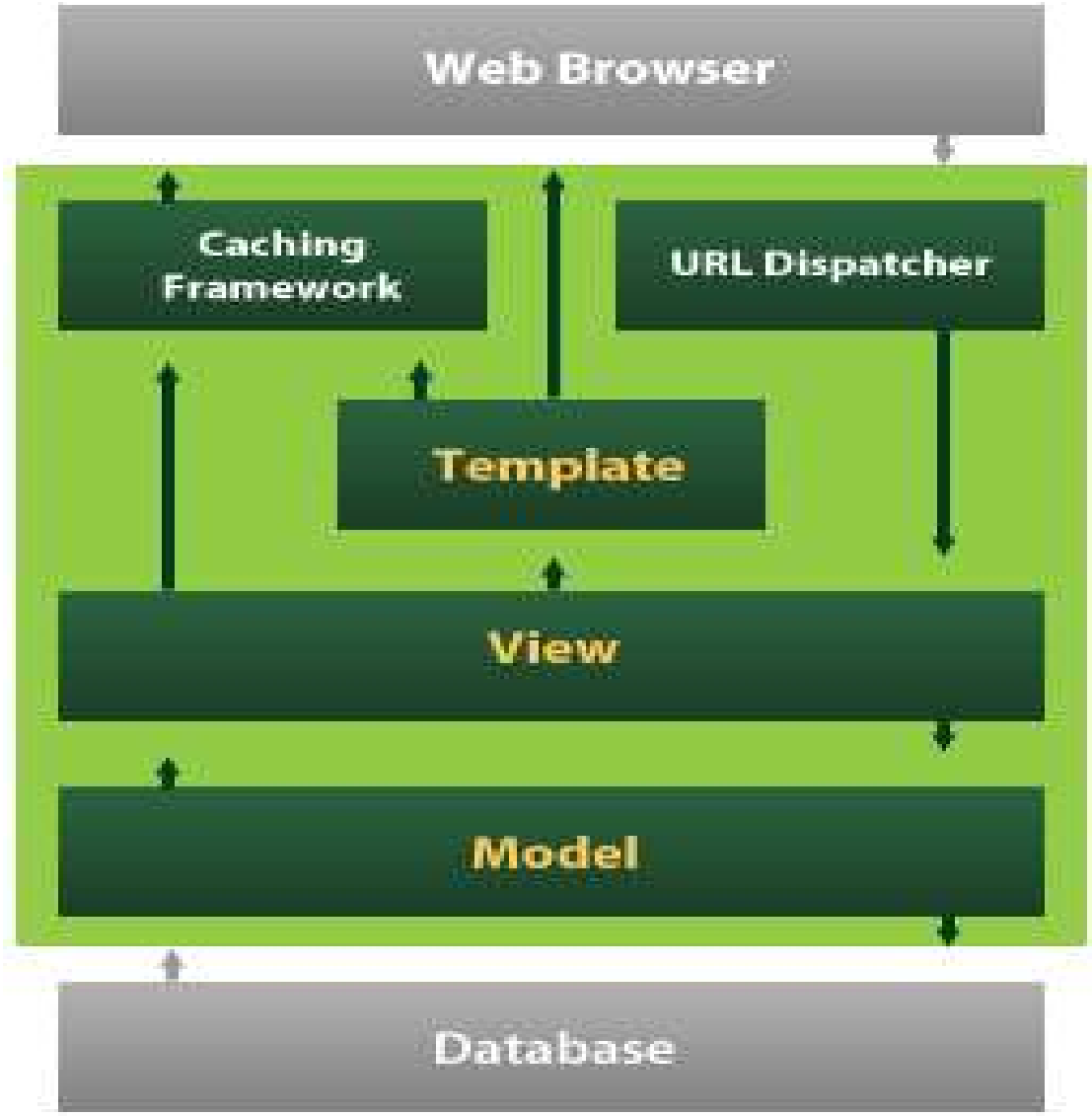


Figure 7.1. The files and data models.

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

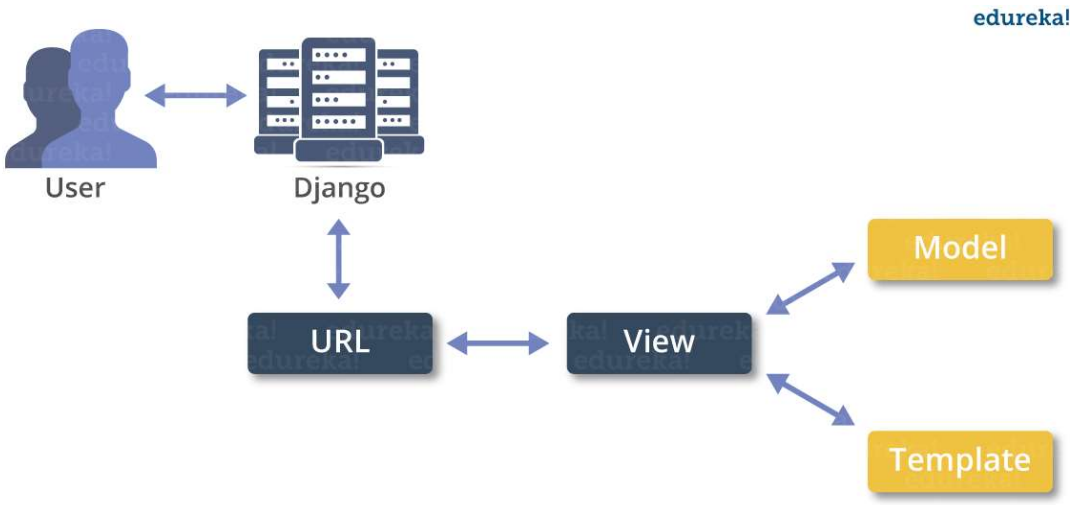


Figure 8.2. A dynamically through introspection and configured via admin models

SYSTEM REQUIREMENT:

HARDWARE REQUIREMENTS:

- **System** : Pentium IV 2.4 GHz.
- **Hard Disk** : 40 GB.
- **Ram** : 512 Mb.

SOFTWARE REQUIREMENTS

- **predating system** : Windows 7 Ultimate.
- **Coding Language** : Python.
- **Web Framework** : Django.
- **Designing** : Html, CSS, JavaScript.
- **Data Base** : MySQL.

SAMPLE CODE:

Urls.py

```
"""recommendersystem URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.2/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.conf.urls import url
from django.conf.urls.static import static
from django.contrib import admin
from django.urls import path

from recommendersystem import settings
from simple.views import index, user, userregistration, adminlogin, adminloginaction,
adminhome, adminbase, \
    viewadminuserpage, activateusers, logout, uploadfile, storecsvdata, faslogin,
fasloginaction, fashome,faspredection
from user.views import userlogincheck, userhome, userbase, userfitness, status,
viewuserdailystatus,UserPredections

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', index, name="index"),
    url(r'^index/', index, name="index"),
    url(r'^user/',user,name="user"),

    url(r'^userregistration/',userregistration,name="userregistration"),
    url(r'^userlogincheck/', userlogincheck, name="userlogincheck"),
    url(r'^userhome/', userhome, name="userhome"),
    url(r'^userbase/', userbase, name="userbase"),
    url(r'^uploadfile/',uploadfile,name="uploadfile"),
    url(r'^storecsvdata/',storecsvdata,name="storecsvdata"),
    url(r'^UserPredections/',UserPredections,name='UserPredections'),

    url(r'^adminlogin/', adminlogin, name="adminlogin"),
    url(r'^adminhome/', adminhome, name="adminhome"),
    url(r'^adminbase/',adminbase,name="adminbase"),
```

```
    url(r'^adminloginaction/', adminloginaction, name="adminloginaction"),
    url(r'^viewadminuserpage/',viewadminuserpage, name="viewadminuserpage"),
    url(r'^activateusers/$', activateusers, name="activateusers"),

    url(r'^faslogin/', faslogin, name="faslogin"),
    url(r'^fasloginaction/', fasloginaction, name="fasloginaction"),
    url(r'^fashome/', fashome, name="fashome"),
    url(r'^userfitness/', userfitness, name="userfitness"),
    url(r'^status/', status, name="status"),
    url(r'^viewuserdailystatus/',viewuserdailystatus,name="viewuserdailystatus"),
    url(r'^faspredection/',faspredection,name='faspredection'),

    url(r'^logout/',logout,name="logout"),
]
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

simple.forms.py

```
from django import forms
from django.core import validators
import datetime
from django.utils import timezone

from simple.models import userregistrationmodel, upload, dailystatus


class userregistrationform(forms.ModelForm):
    name = forms.CharField(widget=forms.TextInput(), required=True, max_length=100)
    loginid = forms.CharField(widget=forms.TextInput(), required=True, max_length=100)
    password = forms.CharField(widget=forms.PasswordInput(), required=True,
max_length=100)
    gender = forms.ChoiceField(choices=[('female','female'),('male','male')])
    age = forms.CharField(widget=forms.NumberInput(), required=True, max_length=100)
    height = forms.FloatField(required=False, max_value=250, min_value=0,
widget=forms.NumberInput(attrs={'id': 'form_homework', 'step': "0.01"}))
    weight = forms.CharField(widget=forms.NumberInput(), required=True,
max_length=100)
    email = forms.EmailField(widget=forms.TextInput(), required=True)
    contact = forms.CharField(widget=forms.NumberInput(), required=True,
max_length=100,validators=[validators.MaxLengthValidator(10),validators.MinLengthVa
lidator(10)])
    authkey = forms.CharField(widget=forms.HiddenInput(), initial='waiting',
max_length=100)
    status = forms.CharField(widget=forms.HiddenInput(), initial='waiting',
max_length=100)

    class Meta:
        model = userregistrationmodel
        fields =
['name','loginid','password','gender','age','height','weight','email','contact','authkey','status']
```

```
class UploadfileForm(forms.ModelForm):

    class Meta:
        model = upload
        fields = ('filename','description','file')




class dailystatusform(forms.ModelForm):
    name = forms.CharField(widget=forms.TextInput(), required=True, max_length=100)
    excercise =
forms.CharField(widget=forms.TextInput(),required=True,max_length=100)
    email = forms.CharField(widget=forms.TextInput(),required=True,max_length=100)
    contact =
forms.CharField(widget=forms.NumberInput(),required=True,max_length=100)
    date = forms.DateField(widget=forms.DateInput())
    #duration =
forms.ChoiceField(choices=[('1hour','1hour'),('2hours','2hours'),('3hours','3hours'),('4hours'
,'4hours'),('5hours','5hours'),('6hours','6hours'),('7hours','7hours'),('8hours','8hours'),('9hours
','9hours'),('10hours','10hours'),('11hours','11hours'),('12hours','12hours'),('13hours','13hour
s'),('14hours','14hours'),('15hours','15hours'),('16hours','16hours'),('17hours','17hours'),('18h
ours','18hours'),('19hours','19hours'),('20hours','20hours')])
    duration =
forms.TimeField(widget=forms.TimeInput(format='%H:%M'),required=True)
    caloriesburned =
forms.CharField(widget=forms.NumberInput(),required=True,max_length=100)
    class Meta:
        model = dailystatus
        fields = ('name','excercise','email','contact','date','duration','caloriesburned')
```

simple.models.py

```
from django.db import models


class userregistrationmodel(models.Model):
    name = models.CharField(max_length=100)
    loginid = models.CharField(max_length=100)
    password = models.CharField(max_length=100)
    gender = models.CharField(max_length=100)
    age = models.CharField(max_length=100)
    height = models.FloatField()
    weight = models.CharField(max_length=100)
    email = models.EmailField()
    contact = models.CharField(max_length=100)
    authkey = models.CharField(max_length=100)
    status = models.CharField(max_length=100)
```

```
def __str__(self):
    return self.email


class upload(models.Model):
    #uuid = models.CharField(max_length=30)
    filename = models.CharField(max_length=100)
    description = models.CharField(max_length=100,blank=True)
    file = models.FileField(upload_to='files/pdfs/')

    def __str__(self):
        return self.filename
    def delete(self, *args, **kwargs):
        self.file.delete()
        super().delete(*args,**kwargs)


class excercisesdata(models.Model):
    age = models.CharField(max_length=500)
    weight = models.CharField(max_length=300)
    excercise1 = models.CharField(max_length=300)
    excercise2 = models.CharField(max_length=300)
    diet = models.CharField(max_length=255)


    def __str__(self):
        return self.excercise1


class dailystatus(models.Model):
    name = models.CharField(max_length=100)
    excercise = models.CharField(max_length=100)
    email = models.CharField(max_length=100)
    contact = models.FloatField(max_length=100)
    date = models.DateField(max_length=50)
    duration = models.CharField(max_length=50)
    caloriesburned = models.CharField(max_length=100)

    class Meta:
        unique_together = ('name', 'date',)


    def __str__(self):
        return self.email
```

Simple.views.py

```
import csv
from collections import defaultdict
from io import TextIOWrapper
from random import randint

from django.contrib import messages
from django.http import HttpResponseRedirect, HttpResponse
from django.shortcuts import render, redirect

from simple.forms import userregistrationform, UploadfileForm
from simple.models import userregistrationmodel, excercisesdata

from user.algorithms.anntest import NeuralNetwork
import numpy as np
import random
from user.algorithms.LogisticRegressionTest import runLogisticAlgo



def index(request):
    return render(request,"index.html")

def userregistration(request):
    if request.method == 'POST':
        form = userregistrationform(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'you are successfully registred')
            return HttpResponseRedirect('user')
        else:
            print('Invalid')

    else:
        form = userregistrationform()
    return render(request,"user/userregistration.html",{'form':form})


def user(request):
    context={}
    return render(request,'user/user.html',context)


def adminlogin(request):
    return render(request,"admin/adminlogin.html")

def adminloginaction(request):
    if request.method == "POST":
        #if request.method == "POST":
            usid = request.POST.get('username')
```

```
            pswd = request.POST.get('password')
            if usid == 'admin' and pswd == 'admin':
                return render(request,'admin/adminhome.html')
            else:
                messages.success(request, 'Invalid user id and password')
    #messages.success(request, 'Invalid user id and password')
    return render(request,'adminlogin.html')


def adminbase(request):
    return render(request,"adminbase.html")

def adminhome(request):

    return render(request,'admin/adminhome.html')


def viewadminuserpage(request):
    userdata = userregistrationmodel.objects.all()
    return render(request,'admin/viewuserdata.html',{'object':userdata})

def activateusers(request):
    if request.method == 'GET':
        usid = request.GET.get('pid')
        authkey = random_with_N_digits(8)
        status = 'activated'
        print("USID = ",usid,authkey,status)
        userregistrationmodel.objects.filter(id=usid).update(authkey=authkey , status=status)
        userdata = userregistrationmodel.objects.all()
        return render(request,'admin/viewuserdata.html',{'object':userdata})
```

CHAPTER-8
INPUT AND OUTPUT DESIGN

INPUT AND OUTPUT DESIGN

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

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-9
SYSTEM STUDY

SYSTEM STUDY

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**

ECONOMICAL FEASIBILITY

The 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.

TECHNICAL FEASIBILITY:

The 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.

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.

CHAPTER-10
SYSTEM TEST

SYSTEM TEST

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.

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.

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.

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.

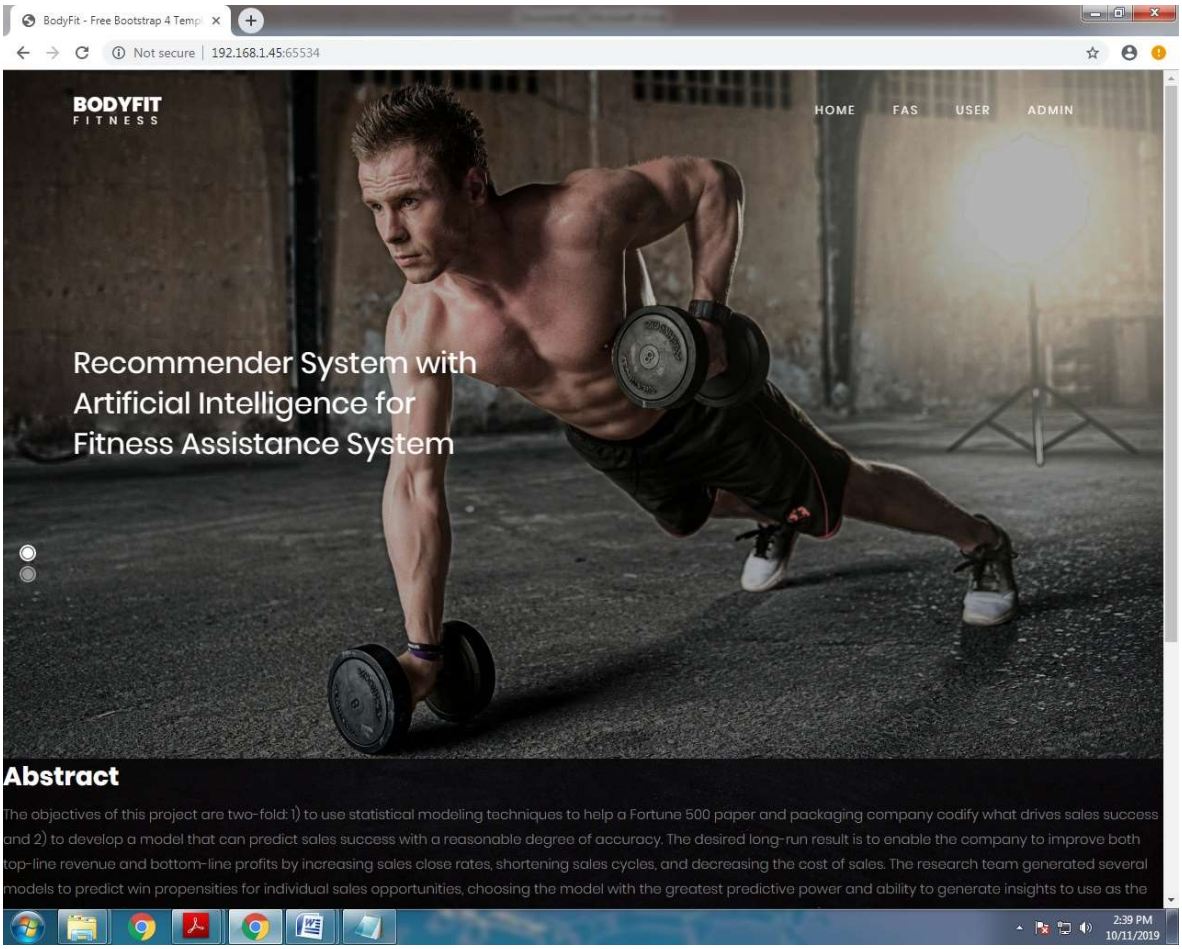
SAMPLE TEST CASES

S.no	Test Case	Excepted Result	Result	Remarks(IF Fails)
1	User REGISTERED	If user registration successfully.	Pass	If user is not registered.
3	ADMIN	user rights will be accepted here.	Pass	If user are not registered.
5	user LOGIN	If user name and password is correct then it will getting valid page.	Pass	If user name or password is not correct.
6	FAS LOGIN	If FAS name and password is correct then it will getting valid page.	Pass	If FAS name or password is not correct.
7	FAS upload csv files	If FAS is correct then it will getting valid page.	Pass	If FAS are not correct.
8	user view recommender exercises data	If users view recommender exercise data based on registered data .it will show valid page.	Pass	If user are not correct.
9	User update daily status	If users Give status and user prediction, it will show valid page .	Pass	If user are not correct.

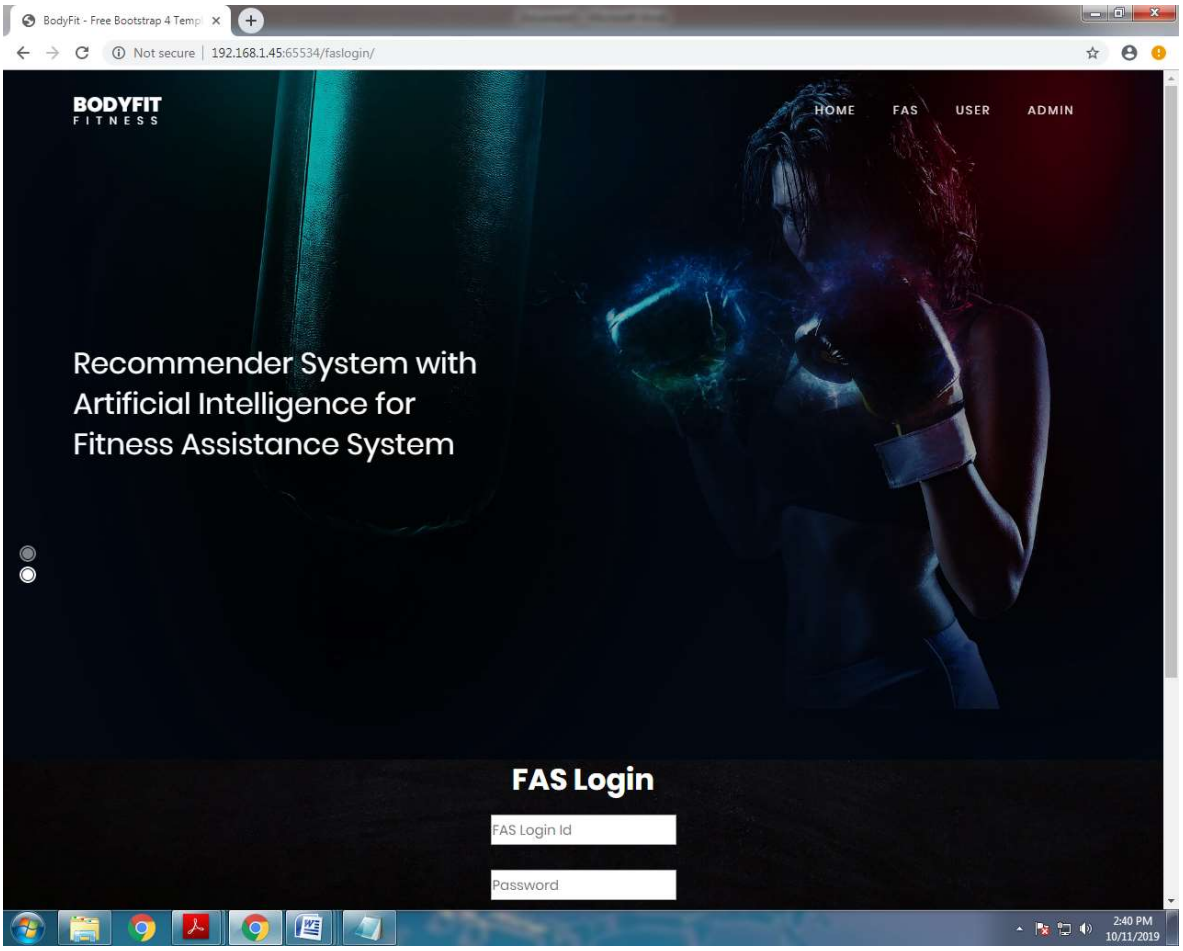
Table-10.1. Sample test cases

CHAPTER-11
SCREENSHOTS

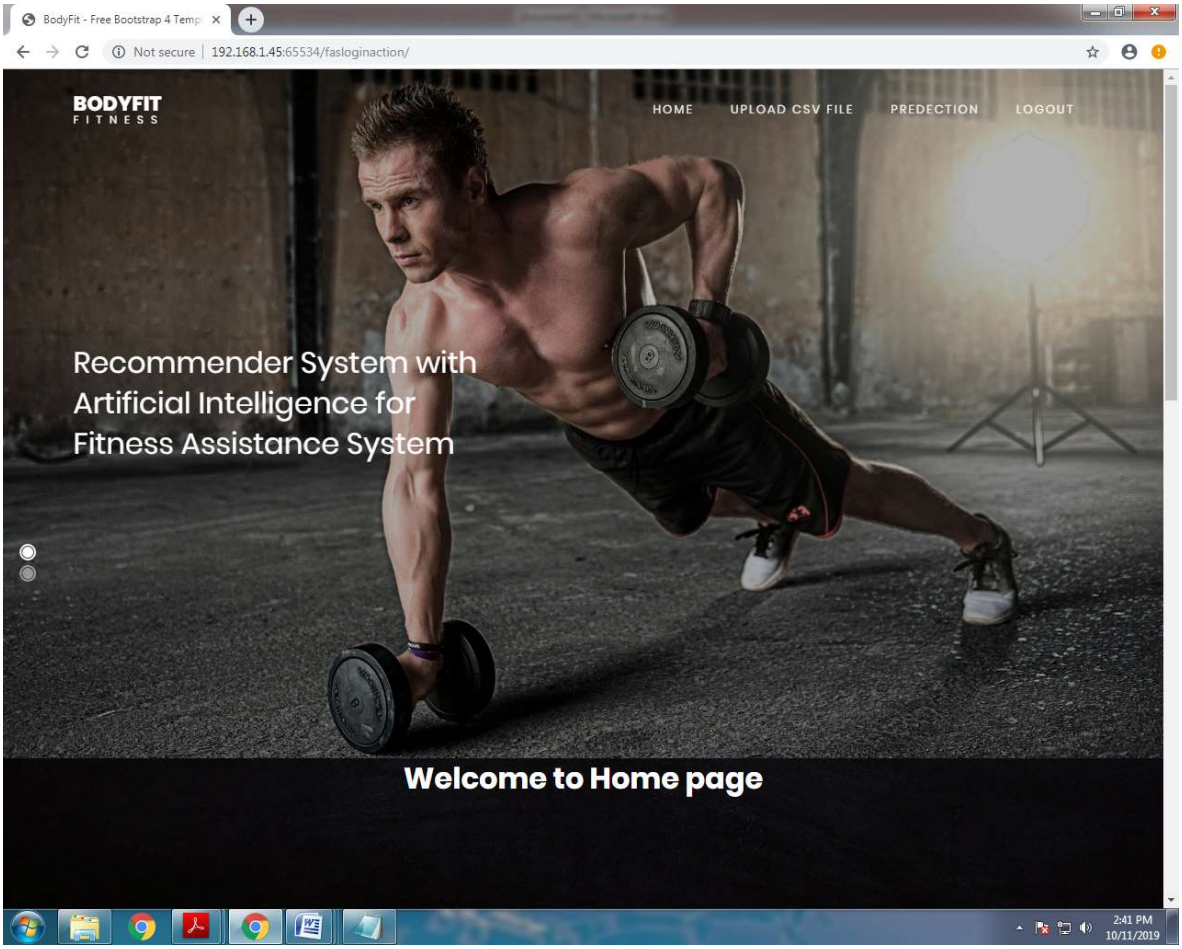
SCREENSHOTS



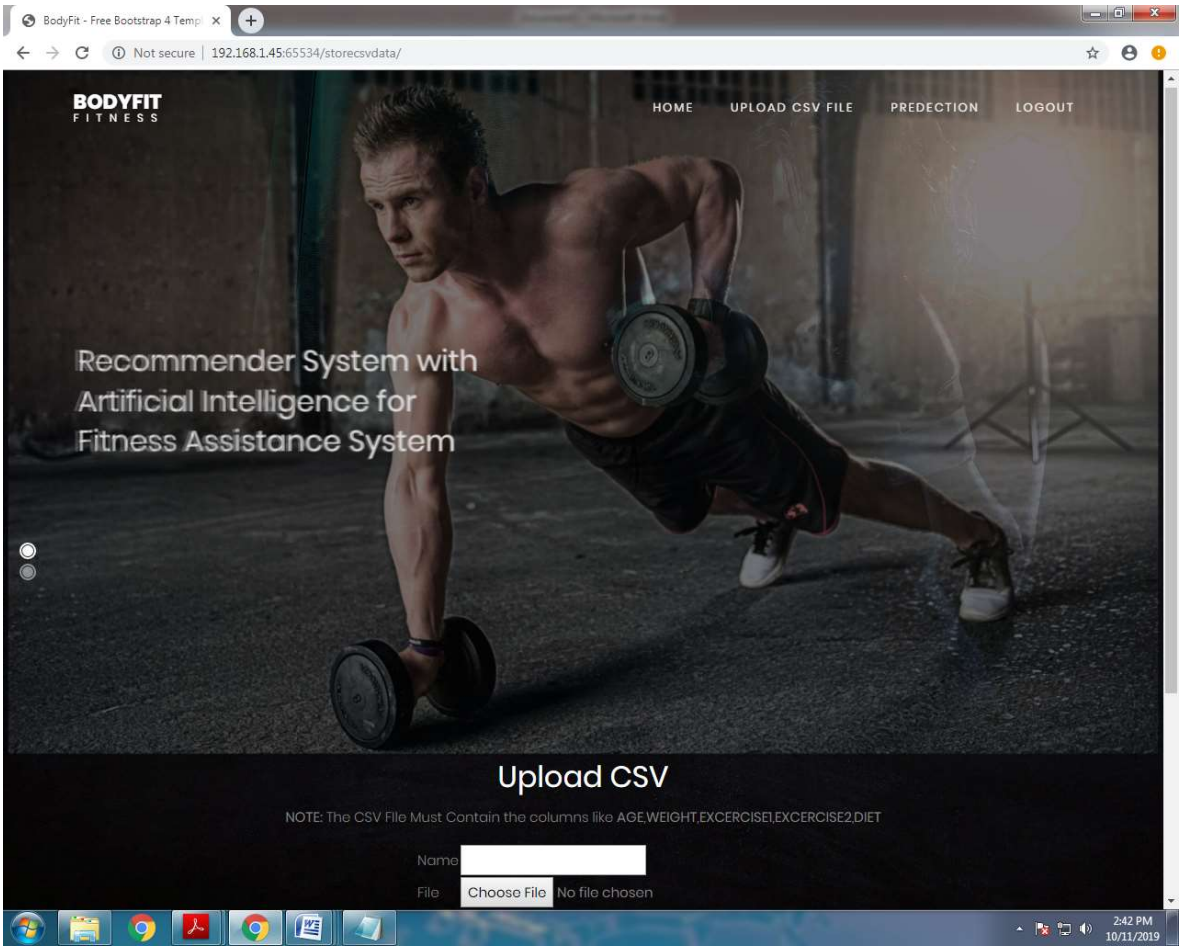
Screen-1: Home Page



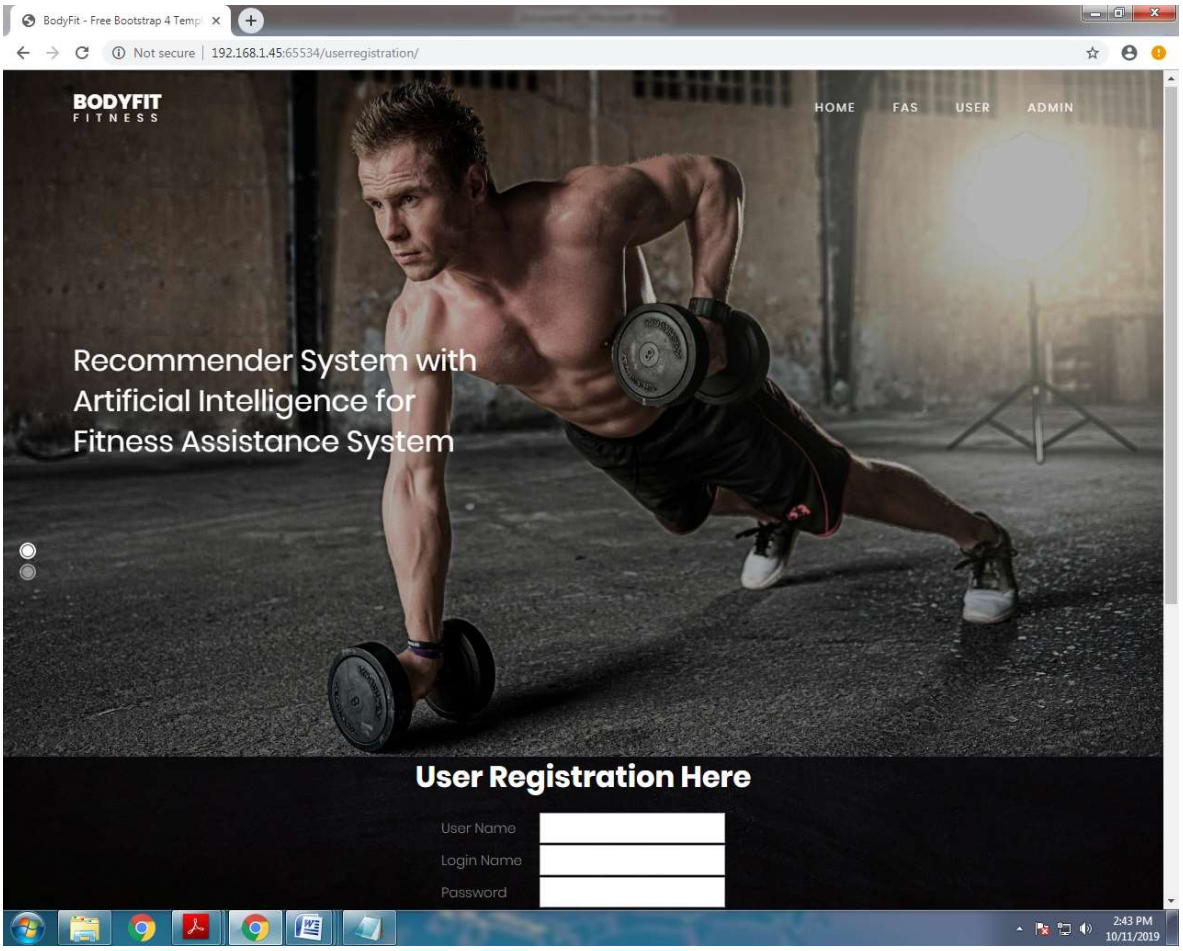
Screen-2: FAS login



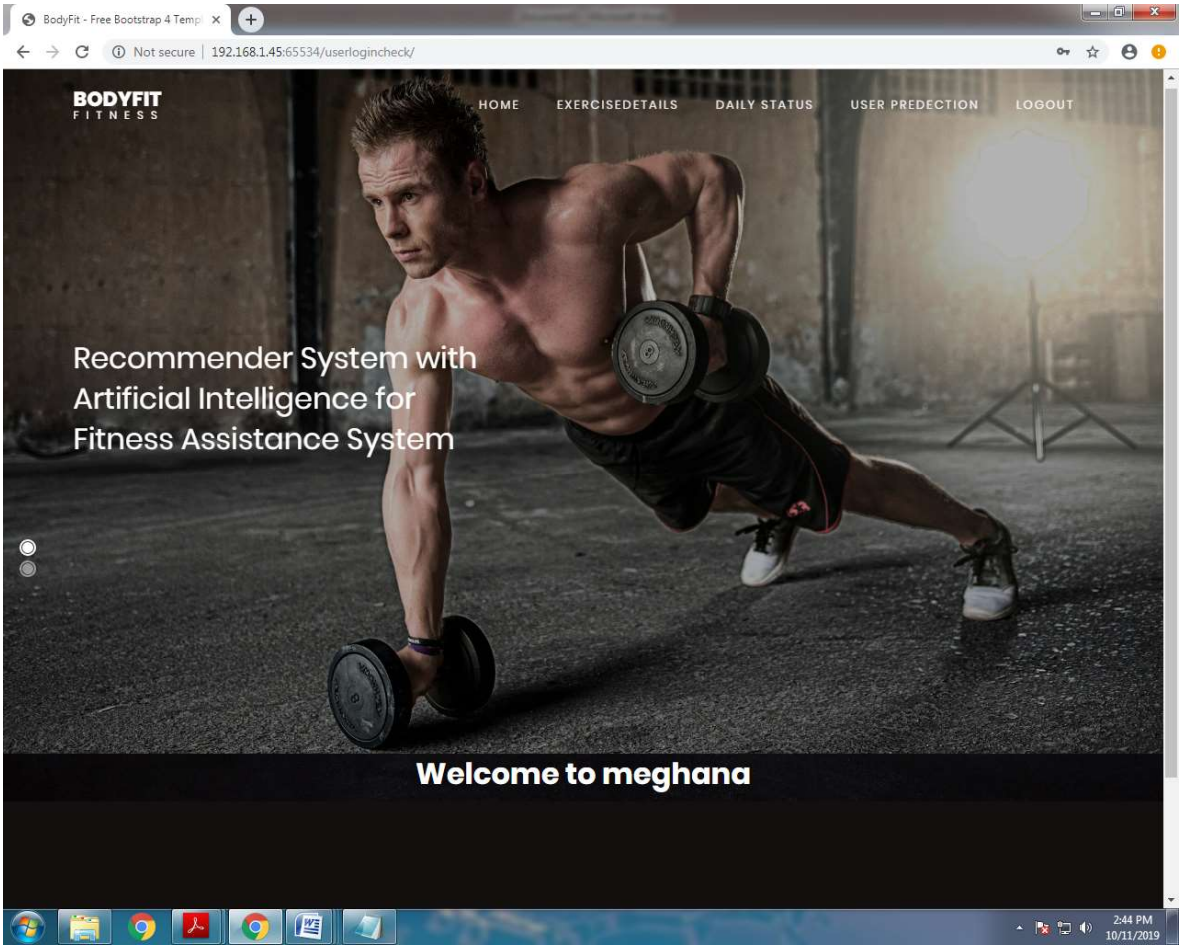
Screen-3: FAS Home page



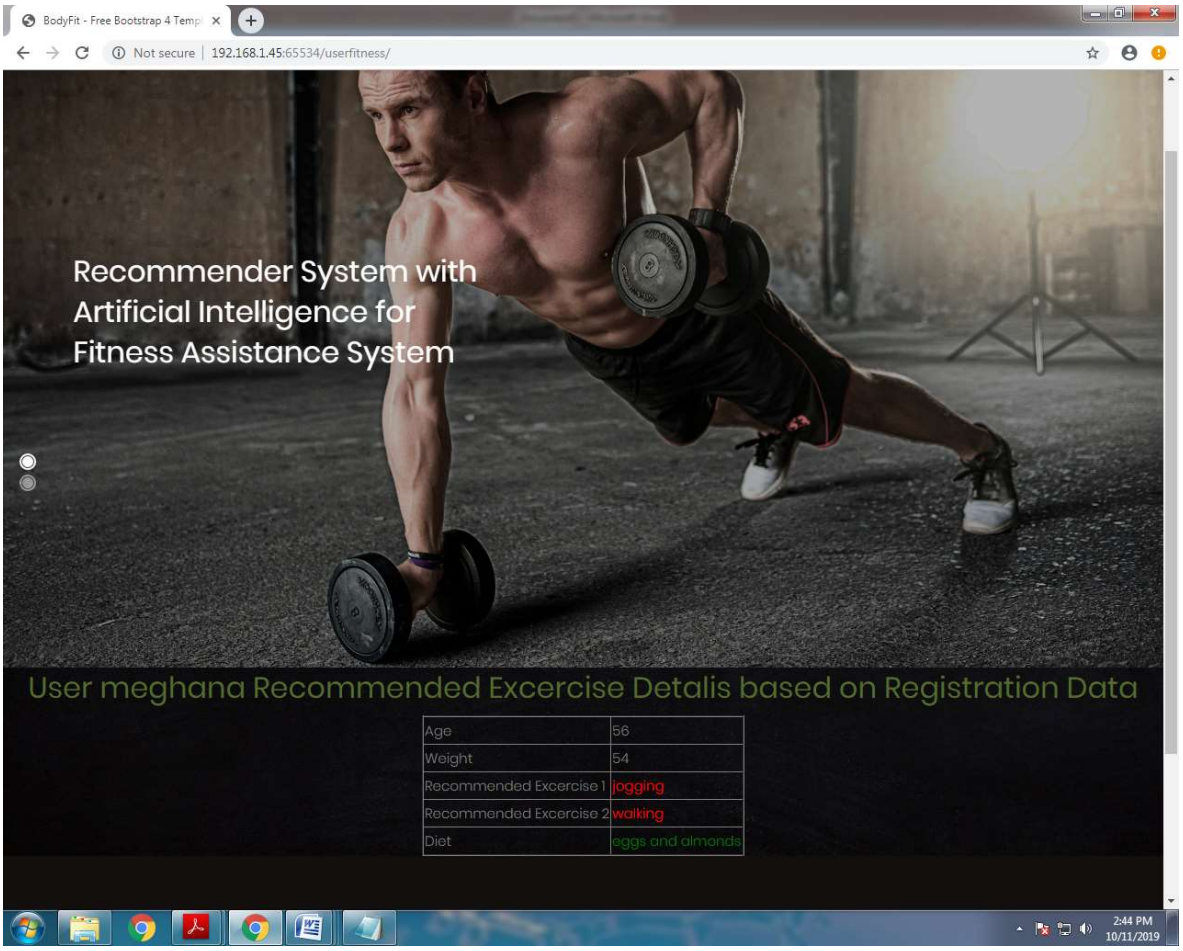
Screen-4: FAS Upload csv file



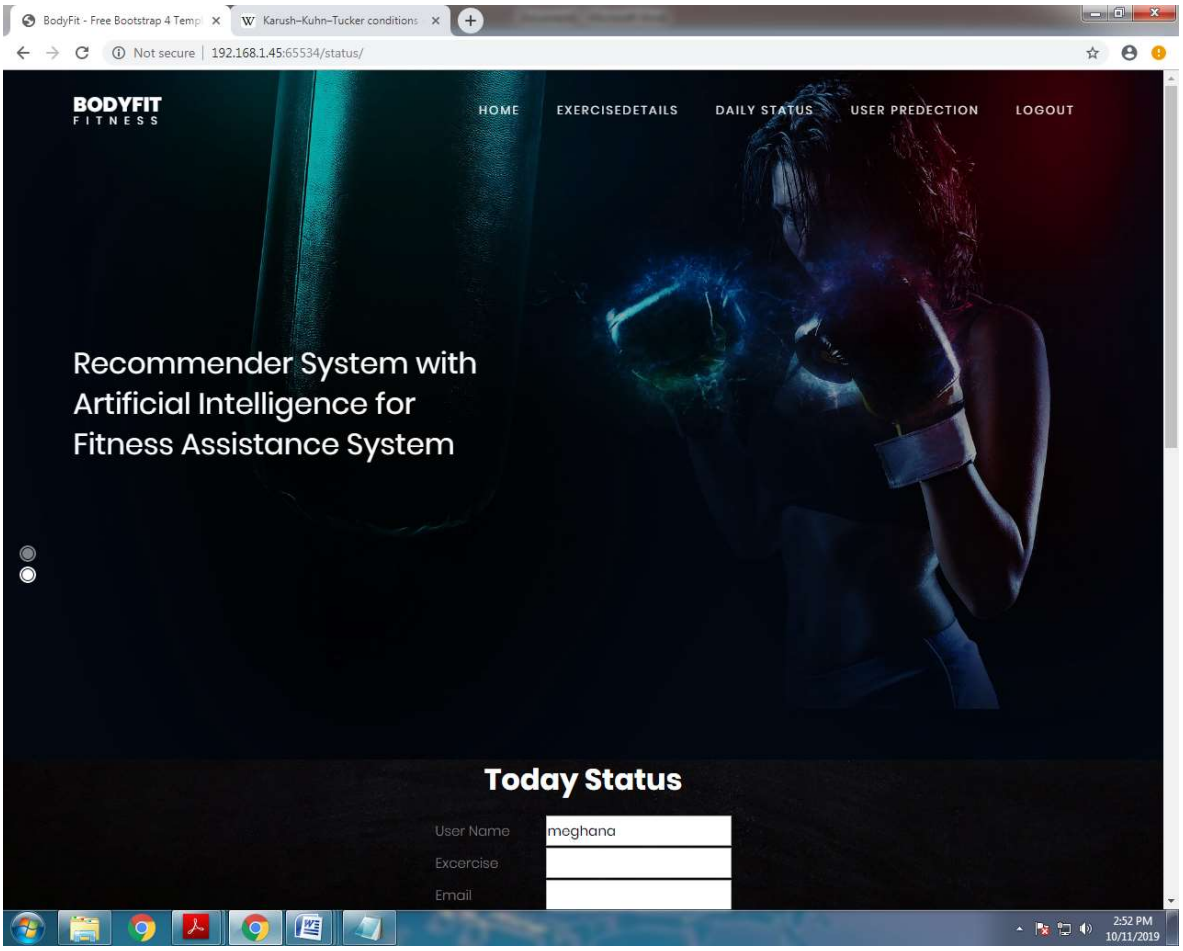
Screen-5: User registration page



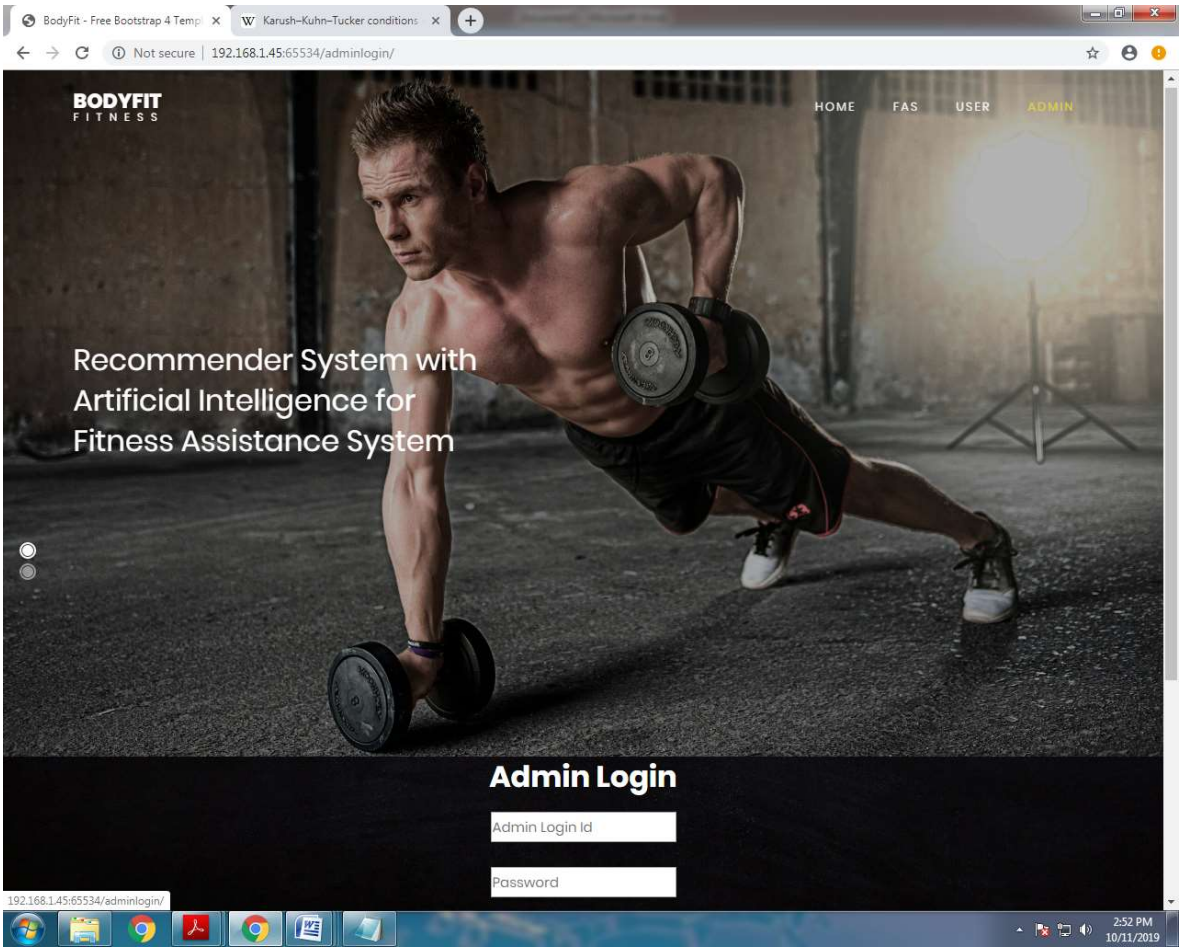
Screen-6: User Homepage



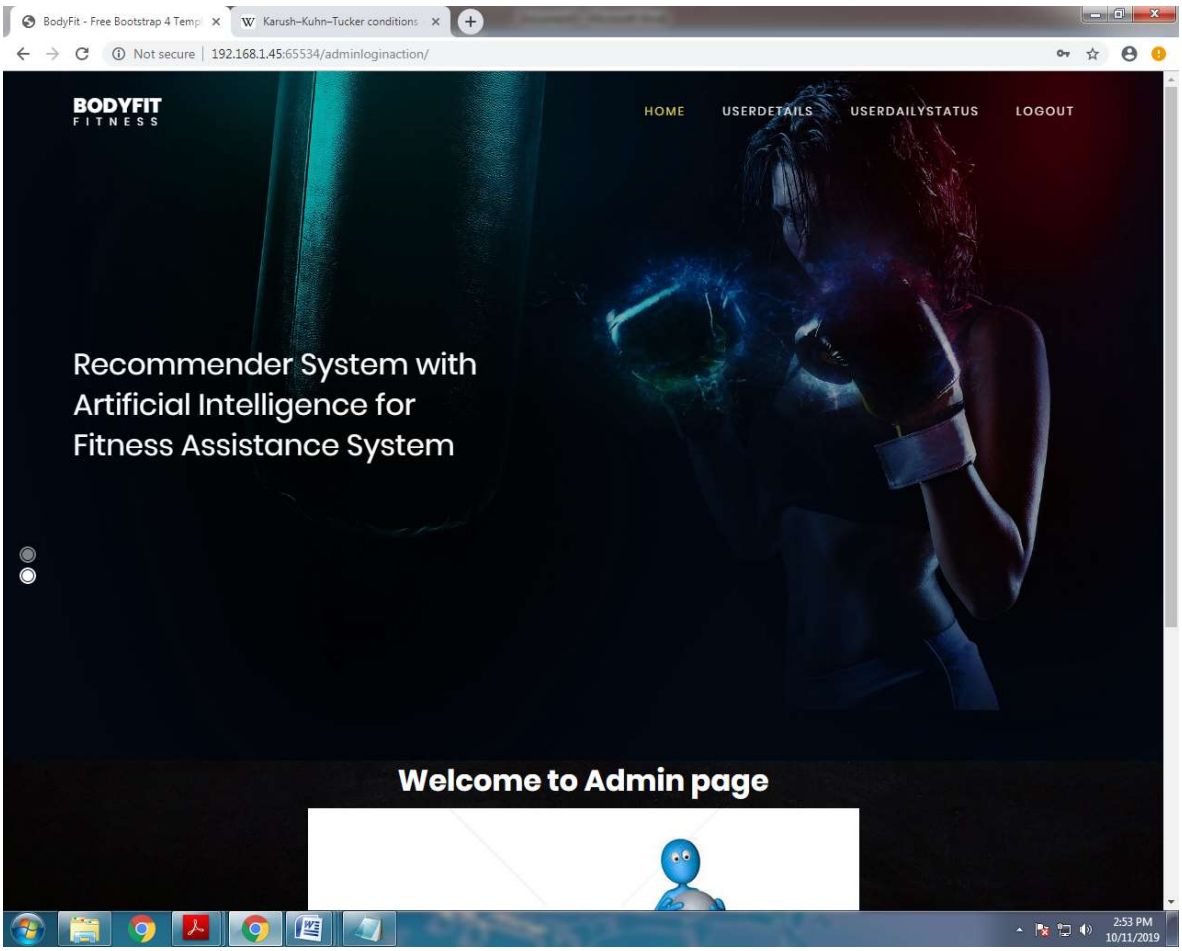
Screen-7: User Exercise Details



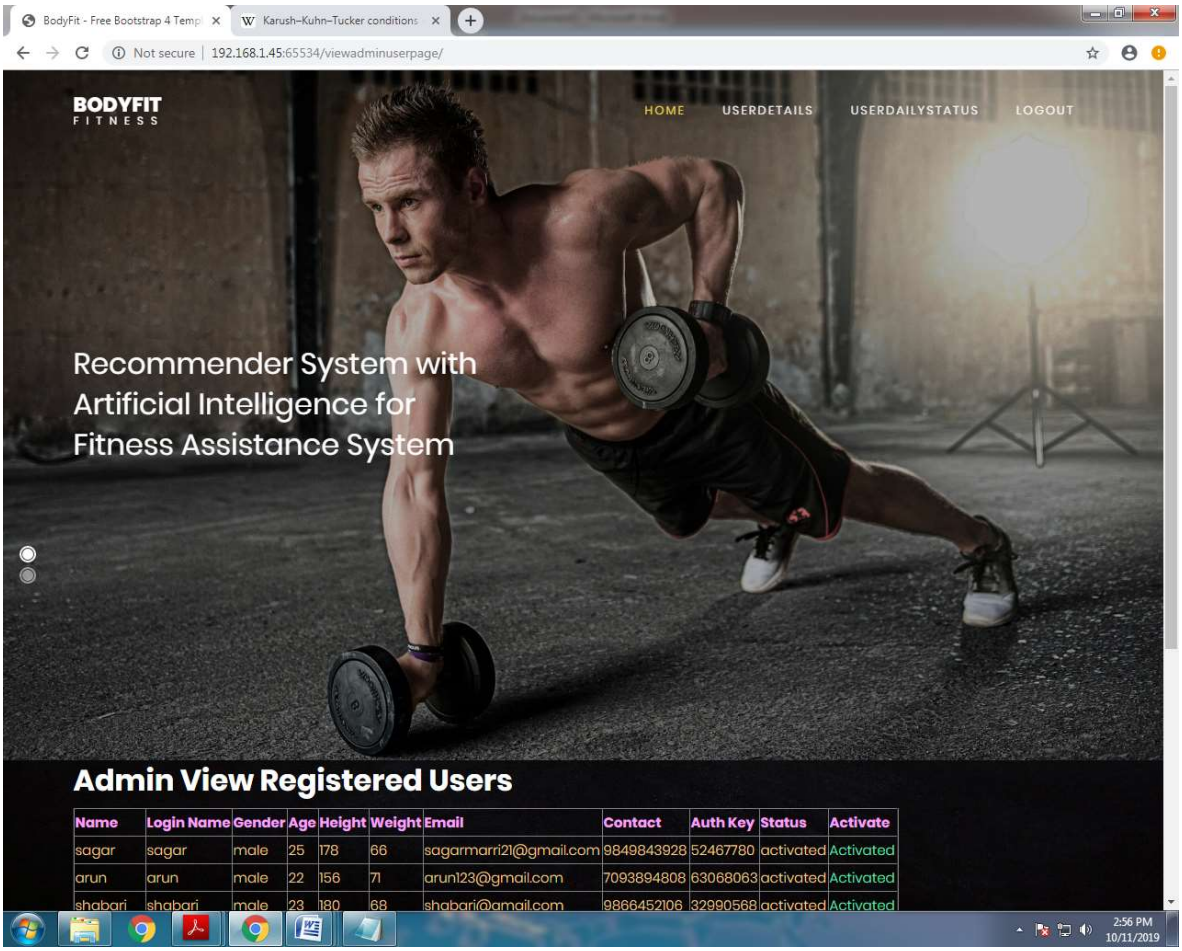
Screen-8: User Daily status



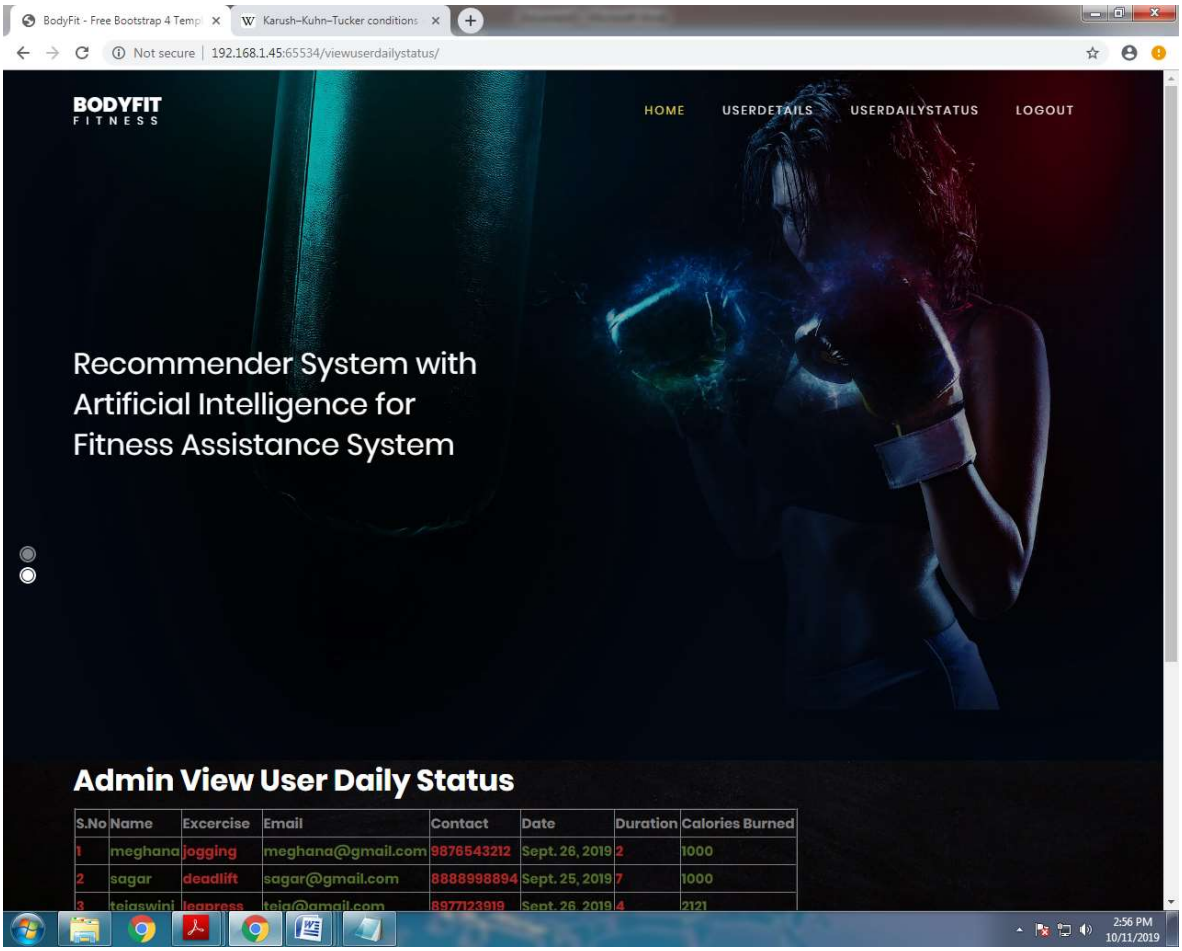
Screen-9: Admin Login



Screen-10: Admin Home page



Screen-11: Admin view registered users



Screen-12: Admin view USER Daily status

CHAPTER-12
CONCLUSION

CONCLUSION

In this study, we proposed RS for fitness assistance system and a novel method for fitness workout recommendation with artificial intelligence algorithms. We developed a system with several machine learning algorithms to predict and train data to give the suggestion for the fitness workout. The ANN with LR implements the prediction of workout parameters with the best accuracy. The proposed RS is expected to give better recommendation for user to do exercise. Table IV illustrates the result of User#1 with the purpose of muscle up between suggested workout and the supposed rules. As can be seen in Table IV, the exercise weight for User #1 is in the range of the supposed rule. In the meanwhile, the repetition and break time also approach the values within the range of the assumed rule as shown in Table I. As future work of this study, we plan to focus on improving the TA module in the proposed RS with Soar agent by designing the RL algorithm to recommend several workouts for existing member's average selection. TA will be developed in future work for improving its features to calculate the epsilon value of epsilon-greedy method, and validate the suggested workout for approaching the stuitable workout plan to the users. Consequently, the proposed RS will play a role of professional trainer for user in future.

CHAPTER-13
FUTURE SCOPE

FUTURE SCOPE

- As future work of this study, we plan to focus on improving the TA module in the proposed RS with Soar agent by designing the RL algorithm to recommend several workouts for existing member's average selection.

- TA will be developed in future work for improving its features to calculate the epsilon value of epsilon-greedy method, and validate the suggested workout for approaching the suitable workout plan to the users

CHAPTER-14
REFERENCES

REFERENCES

1. G. Adomavicius, A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions”, IEEE Trans. On Knowledge and Data Engineering, vol. 17, pp. 734-749, In April 2005.
2. I. Portugal, P. Alencar, and D. Cowan, “The use of machine algorithms in recommender systems: A systematic review”, Expert Systems with Application, vol. 97, pp. 205-227, In May 2018.
3. J. Aguilar, P. Valdiviezo-Díaz, and G. Riofrio, “A general framework for intelligent recommender systems”, Applied Computing and Informatics, vol. 13, pp. 147-160, In July 2017.
4. S. Dharia, V. Jain, J. Patel, J. Vora, S. Chawla, M. Eirinaki, “PRO-Fit: A personalized fitness assistant framework” in Proc. 28th International Conf. on Software engineering and knowledge engineering SEKE, Redwood City, CA, 2016.
5. M. Afzal, S. I. Ali, R. Ali, M. Hussain, T. Ali, W. A. Khan, M. B. Amin, B. H. Kang, and S. Y. Lee, “Personalization of wellness recommendation using contextual interpertation”, Expert Systems with Applications, vol. 98, pp. 506-521, In April 2018.
6. M. Donciu, M. Ionita, M. Dascalu, S. Trasusan-Matu, “The Runner—recommender system of workout and nutrition for Runners”, in Proc. 13th International Symposium on Symbolic and Numeric Algorithms for Scientific (SYNASC), Romania, In 2012.
7. M. Rosemary, H. Andrew, “The essential guide to fitness: for the fitness instructor”, Sydney, Pearson Australia, pp. 135.
8. J. M. Reynolds, T. J. Gordon, R. A. Robergs, “Prediction of one repetition maximum strength from multiple repetition maximum testing and anthropometry”, Journal of Strength and Conditioning Research, vol. 20, pp. 584-592, In August 2006.
9. M. Tokic, “Adaptive e-Greedy Exploration in Reinforcement Learning Based on Value Differences”, in Proc. 33rd Annual German Conference on AI, Germany, In 2010.

CHAPTER-15
BIBLIOGRAPHY

BIBLIOGRAPHY

- http://www.tornadoweb.org/en/stable/
- https://www.egrovesys.com/blog/how-does-the-django-framework-inpython-overcome-the-frameworks-in-php/
- https://pypi.python.org/pypi/radon
- https://www.ibm.com/developerworks/aix/library/au-cleancode/