CoachOneSelf

PROJECT REPORT

Submitted by,

S JUDITH DAVID

MEGHNA RAGHU

ALEENA MARY V S

Guided by,

Mr.Prinson P.T

For the award of the degree of

BCA

(St.Thomas' College (Autonomous), Thrissur, Kerala)

DEPARTMENT OF COMPUTER APPLICATION

ST.THOMAS' COLLEGE (AUTONOMOUS) ,THRISSUR

KERALA

(Affiliated to University of Calicut)

April 2022

ACKNOWLEDGEMENT

We are glad to present our project report entitled “ CoachOneSelf ” prepared as a part of our final year Bachelor of Computer Application Course.

First and foremost we are extremely grateful to God Almighty, whose blessings have given us courage and strength to complete this project successfully.We express our heartiest gratitude to Dr. Martin K.A., Principal, St. Thomas’ College (Autonomous) Thrissur for his inspiration and encouragement throughout our project. We would like to express our gratitude to Mrs.Bindhia K.F, Head of the Department of Computer Application, who has served as a host of valuable encouragement. It is wordless to say thanks to our internal guide Mr. Prinson P.T., Assistant Professor, Department of Computer Application, for his proper guidance and support.We extend our sincere gratitude to all other faculties in the department, who have contributed to fulfill our attempt. It is our greatest pleasure to take this opportunity to infer our gratitude towards Soft Solutions Kochi for their kind considerations and assistance.

We thank all our family members, friends and all other working staff who helped us in successful completion of our project.

Place:Thrissur	S Judith David (THATBCA020)
Date:	Meghna Raghu (THATBCA017)
	Aleena Mary V S(THATBCA014)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person or material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Place: Thrissur

Date:

Signature:
Name:S Judith David
Reg.No:THATBCA020

Signature:
Name:Meghna Raghu
Reg.No: THATBCA017

Signature:
Name:Aleena Mary V S
Reg.No: THATBCA015

ST.THOMAS’ COLLEGE (AUTONOMOUS)

THRISSUR

CERTIFICATE

This is to certify that the project report entitled ‘ **CoachOneSelf** ’ submitted by **S Judith David (THATBCA020),Meghna Raghu(THATBCA017), Aleena Mary V S(THATBCA014)** to St.Thomas’ (Autonomous) College for the award of Degree of Bachelor of Computer Application (BCA) is a bonafide record of the project work carried out by them under my supervision and guidance. The content of the report, in full or parts have not been submitted to any other institutes or university for the award of any other Degree or Diploma.

Signature	College Seal	HOD
Project Guide		

Place: Thrissur

Date:

Certified that the candidate was examined by us in the project viva voce examination held on

…………………..

Examiners:

1: …………………

2: …………………

ST.THOMAS' COLLEGE (AUTONOMOUS)

AFFILIATED TO UNIVERSITY OF CALICUT

THRISSUR-680001,KERALA

College with Potential for Excellence*Nationally Reaccredited with 'A'Grade 3.4/4.00

ESTD.1889

CERTIFICATE

This is to certify that the project report entitled ' **CoachOneSelf** ' submitted by **S Judith David (THATBCA020), Meghna Raghu (THATBCA017), Aleena Mary V S (THATBCA014)**. Final semester Bachelor of Computer Application students of St.Thomas'College(Autonomous), Thrissur for the partial fulfillment of the requirement for the award of Bachelor of Computer Application(BCA) is a record of bonafide work carried out at our organization from November 2021 to March 2022 and was successfully completed.

Place:

Date: H.O.D

CONTENTS

Abstract..1
List of Figures..2
1. Introduction 3
1.1 Background..4
1.2 Project Profile..4
1.3 Environmental characteristics..4
1.3.1 Hardware..4
1.3.2 Software..4
2. Problem Definition and Methodology..5
2.1 Problem Definition..6
2.2 Objectives..6
2.3 Methodology..6
2.4 Scope of the project..7
2.5 Constraints..7
3. Requirement Analysis and Specification..8
3.1 Requirement Analysis/Literature Review..9
3.2 Existing System..9
3.3 Proposed System..9
3.4 Requirement Specification..10
3.4.1 Functional Requirements..10
3.4.2 Non-Functional Requirements..11
3.5 Feasibility Study..11
3.5.1 Technical Feasibility.. 11
3.5.2 Economical Feasibility..12
3.5.3 Operational Feasibility.. 12
4. System Design..13
4.1 Users of the System..14
4.2 Architecture Diagram..14
4.2.1 Architecural Diagram..14
4.2.2 Procedural Design.. 15
4.2.3 Data Flow Diagram..15
4.3 Database Design..19
4.3.1 List of Tables..19

4.3.2 List of Enities and Attributes…..20
4.3.3 E-R Diagram…..26
5. Development..30
5.1 Tools…..30
5.2 Modules…...33
6. Validation Criteria and Implementation…... 34
6.1 Test Plans…...35
6.2 Classifications of Testing...35
6.3 Implementation…... 37
Conclusion...38
Bibliograghy...38
Code..39
User Interface.. 74

ABSTRACT

The ability to protect oneself physically from violence is a basic human right that should be reachable for everyone including children, teenagers and adults. Crimes against old aged people is one of issues we face now , they are feared to live alone or go out by themselves. Children faces bullying, physical and sexual abuse hence it’s important to ensure that boys and girls can defend themselves in an appropriate way if the situation requires it.The main objective of our project is to provide people personalised self-defence education to everyone according to ones respective age , gender , skills and health.

LIST OF FIGURES

Architectural Diagram.. 9
Procedural Design... 9
Data Flow Diagram.. 9
List of tables... 9
ER Diagram.. 9

INTRODUCTION

INTRODUCTION

1.1 Background

CoachOneSelf is a project that aims to give comprehensive awareness and training to all people in a personalized way according to their age,gender and needs. Having the ability to defend yourself increases your self esteem and boosts your confidence.Benefits like these only add to the reason to learn self-defence.Without proper training,your mind finds it hard to adapt while under stress.For that reason,it is best to recieve formal training to defend yourself.

1.2 Project Profile

We create a platform that enable the user to search and find the required people for construction.

CoachOneself is laying em phasis on initiative aimed at empowerig peoples of all age groups through Comprehensive awareness and practical training program.It is anenterprise project in which admin should be instructed by a selfdefence trainer or a selfdefence organization.

1.3 Environmental Characteristics

1.3.1 Hardware Configuration

Processor	: Intel® Core™ i3-5005U
Speed	: 1.60GHz
Memory	: 4GB Ram
Hard Disk	: 1TB

1.3.2 Software Configuration.

Operating system	: Windows 10
Database	: MySQL
	sqlyog 8.1 version
	XAMPP 3.2.4 version
Platform	: NetBeans 8.2 version
	JDK 8.1 version
Frame Work	: Jsp

PROBLEM DEFINITION AND METHODOLOGY

PROBLEM DEFINITION AND METHODOLOGY

2.1 Problem Definition

People irrespective of their age and gender should learn to protect themselves.Since many indivituals are not even aware of self defence they end up missing the opportunity to protect thems and others.Learnng self-defense should be neccesity in today;s world, but its also important to learn self-defene to ones respective age,gender,health and calliber to avoid physical injuries and to ensure proper safety.

2.2 Objectives

Aim is to provide personalized self defence to everyone according to ones respective age , gender , skills and health. Help individuals to be more focused and prepared in unpleasant situations,develop skills, awareness to understand and react in dangerous situations, to become tough oneself so can protect other people in need to, provides balance between both mental and physical health, learn something that is essential to ones life.

2.3 Methodology

User can register to our websites by filling the registration form and then login to ones account using password and email id.Next step of user is to fill in the questinnaire to get the content.Data such as Video , pdf and images are uploaded by admin to each user after reading the answers of their questionnaire.If user is not satisfied with the content they can change the content by filling the feedback.User can access products of self- defense and further buy them by redirected link to flipkart.

There are five modules :

1. Registration Module:

   User registers with their name ,age ,gender ,email , password , contact , address and aadhar.Both users admin and end user can login.
2. Questionnaire Module :

   Admin add questionnaire for users to fill and user fills the questionnaire to get data.

3. Produt Module :

   Admin adds self-defence products. Both admin and user can view and access self-defence tools.

4. Training session Module :

   Admin verifies the user using their aadhar number and then uploads data after verification. Data is uploaded by accessing their questionnaire answers.

5. Feedback Module :

   User can change the data provided by admin by filling the feedback questionnaire.

2.4 Scope of the Project

This project would help a lot of people to find the right people to construct a building, and have an estimate of the construction. In future updates the website could me more interactive with users with new features. New features such has :

i. live teaching sessions and online classes.

ii. Online purchases of self-defense tools.

iii. Platform for buying and selling self-defense tools.

iv. An interactive comment section to share ones self-defence experiences.

2.5 Constraints

Most of the people are not aware of the concept of self-defence what type of self-defense suits an individual.Learning wrong self-defence techniques causes physical injuries and also ineffective self-defense learning.Our website provides questionnaire created by admin who is self-defense trainer or is someone guided by one.This questionnaire helps admin to understand what kind of data should be uploaded to the user.After the accessing the content if the user is not satisfied they can change the content by filling the feedback questionnaire.

REQUIREMENT ANALYSIS AND SPECIFICATION

REQUIREMENT ANALYSIS AND SPECIFICATION

3.1 Requirement Analysis/Literature review

Requirement analysis results in the specification of software's operational characteristics; indicates software's interface with other system elements and establishes constraints the software must meet. Requirement analysis allows the software engineer to elaborate on basic requirements established during earlier requirement engineering tasks and build models that depict user scenarios, functional activities, problem classes and their relationships, system and class behavior and flow of data as it is transformed. Requirement analysis provides the software designer with representation of information, function and behavior that can be translated to architectural, interface and component level designs. Finally the analysis model and the requirement specification provide the development and the customs with the means to access quality once the software is built. The main requirement of this project is to eliminate the communication gap between a person who wants to build a home and the people needed for the process.

3.2 Existing System

Existing system are sites, youtube videos, apps; they provide content which is not personalized hence leading to inefficient results and unnecessary injuries.

Limitations of existing system

1. Free courses on self-defense don't provide complete expertise as it is available to paid courses only.
2. Provides information to join academies and sites and different sites for kids ,women , beginners,etc.
3. Only focuses on one method or situations.
4. Misleading information is provided.

3.3 Proposed system

The proposed system is designed to eliminate all the disadvantages of the existing system.CoachOneSelf is better as it provides personalized content to the users and the pros of this are :

1. Improvement in ones growth.
2. More interesting and easy to learn.
3. Personalization prevents from users from getting injured because of learning wrong self defense.

4. An option to get desired content.
5. free and provides availablity of self defense tools.

3.4 Requirement Specification

3.4.1 Functional Requirement

In software engineering and system engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Functional requirements are the statement of services the system should provide how the system should react to particular input and how the system should behave in particular situation.The functional requirements may also explicitly state what the system should do.Functional requirements depend on the type of software being developed,the expected users of the software,and the general approach taken by the organization when writing requirements.It should be written in natural language so that system users and managers can understand them . Functional system requirements expand the user requirements and are written for system developers.

The main functional requirements are:

1. The admin approves the user before uploading the training session.
2. The users use email and the password provided at the time of registration. By entering email and password.
3. Users can easily login and can change to their desired content.

3.4.2 Non-Functional Requirement

Non-functional requirements are sometimes defined in terms of metrics (something that can be measured about the system) to make them more tangible. Non-functional requirements may also describe aspects of the system that don't relate to its execution, but rather to its evolution over time (e.g. maintainability, extensibility, documentation, etc).Non functional requirements are the requirements that are not directly concerned with the specific services delivered by the system to its users.Non-functional requirements are often more critical than individual functional requirements.These requirements arise through product requirements organisational requirements and external requirements

The non-functional requirement of CoachOneSelf are:

1. Functionality – It provides good functionality.
2. Usability – Requirements can be used effectively.
3. Software should be user friendly.
4. Maintainability – It should be maintainable and able to add new features

3.5 Feasibility Study

Feasibility analysis is an analysis of possible alternative solution to a problem and a recommendation on the best alternative. Before the request is to be approved, it has to be checked whether the system is feasible or not with respect to the following areas. It is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. The objective of feasibility study is not to solve the problem but to acquire a sense of its scope.
Three key considerations are involved in the feasibility analysis:

1. Technical
2. Economical
3. Operational

3.5.1 Technical Feasibility

Technical Feasibility, involves development of a working model of the product or service. It also provides a visual means to share your concept with others. The objective of the technical feasibility step is to confirm that the product will perform and to verify that there are no production barriers. CoachOneSelf is developed by Java and Html languages so it is technically feasible.

3.5.2 Economical Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the software. Its more commonly known as cost benefit analysis, the procedure to determine the benefits and savings that are expected from a project system and compare them with costs. The purpose of accessing economical feasibility is to identify the financial benefits and cost associated with the development of the project.CoachOneSelf is an enterprise project and it can be used for multiple business tecniques such as colloboration with self-defense instituations,ad companies,etc

3.5.3 Operational Feasibility

In operational feasibility it is check if the system will work. Operational feasibility is reviewed in the early stages of project planning. The project or software which is developed has a interactive, user friendly interface. So operational difficulties are almost eliminated. The user interface is designed in such a way that it gets comfortable and understandable for non-technical person to operate easily. And CoachOneSelf is operationally feasible.

SYSTEM DESIGN

4.SYSTEM DESIGN

The system design is the second of the four system development life cycle phase. It is the phase in which the detailed design of the system selected in the study phase accomplished and the users oriented performance specification is converted into a technical design specification.In this phase must design all the aspect from input and output to reports, database and computer process. The analyst must design the following elements of a system, that are files and database, input design, output design, forms and reports, dialogue an interface, system and program structure, distributed system.

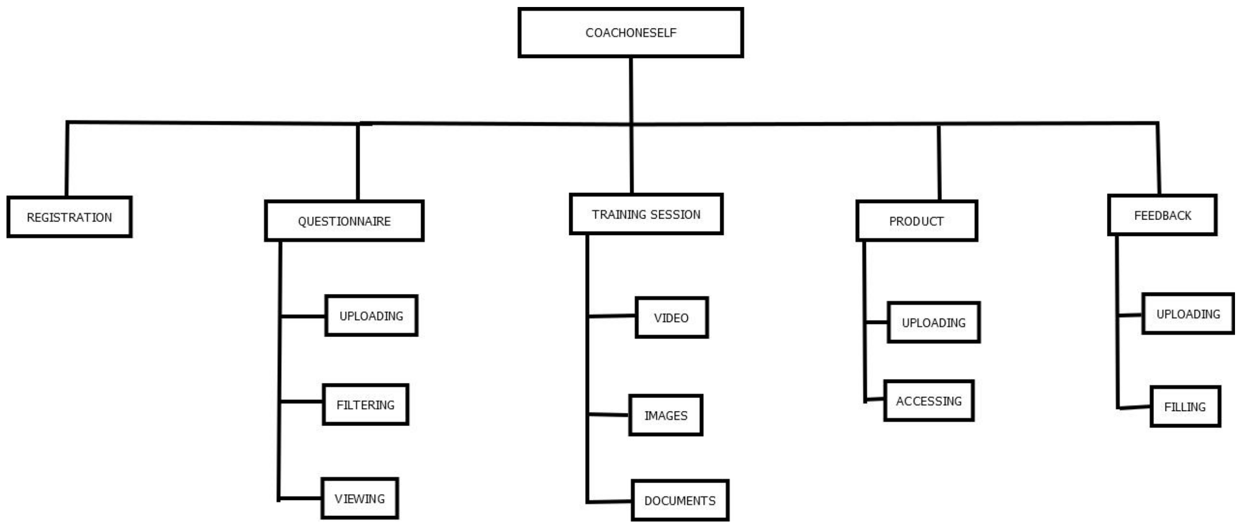
4.1 Users of the system

The main users of this system are admin and user.Our project helps admin to provide efficient and effective self-defence contents and also helps user to access personalized self-defense content. The main objective of our project is to provide quality content to users.

4.2 Architecture diagram

Software architecture involves the high level structure of software system abstraction, by using decomposition and composition, with architectural style and quality attributes. A software architecture design must conform to the major functionality and performance requirements of the system, as well as satisfy the non-functional requirements such as reliability, scalability, portability and availability.

4.1.1 Architectural diagram



4.1.2 Procedural design

1.Registration

1.1 User registers with their name ,age ,gender ,email , password , contact , address and aadhar.

1.2 Both users User1(admin) and end user can login.

2. Questionnaire

2.1 Admin add questionnaire for users to fill.

2.2 User fills the questionnaire to get data.

3. Product

3.1 Admin adds self-defence products.

3.2 Both admin and users can view and access self-defence tools.

4. Training session

4.1 Admin verifies the user using their aadhar number.

4.2 Admin uploads data after verification

4.3 Data is uploaded by accessing their questionnaire answers.

5. Feedback

5.1 Admin adds feedback questions.

5.2 User can change the data provided by admin by filling the feedback questionnaire.

4.1.3 Data flow diagram (DFD)

Data flow diagrams are used widely for modeling the requirements.DFDs show the flow of data through a system.

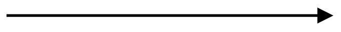
1. All names should be unique. This makes easier to refer items in the DFD. Remember that a DFD is not a flow chart.

2. Arrows in a flow chart represent the order of events; arrows in DFD represent flowing data. A DFD doesn’t imply any order of events.

3. Suppress logical decisions (A diamond shape box is used in flow chart to represent decision points with multiple exit paths of which only one is taken).This implies an ordering of events, which makes no sense in DFD.

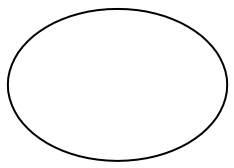
Standard symbols:

1. Data Flow



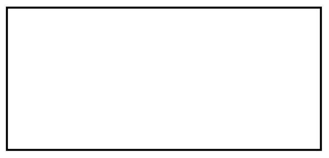
Used to connect process to each other, to sources of sinks; the arrow head indicates direction of data flow.

2. Process



Performs some transformation ofinput data to yield output data.

3.Source/Suit (external entity)



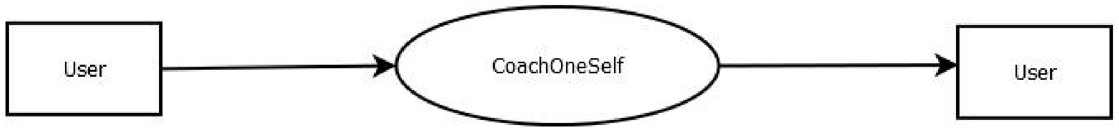
A source of system inputs or sinks of system output.

4. Data store

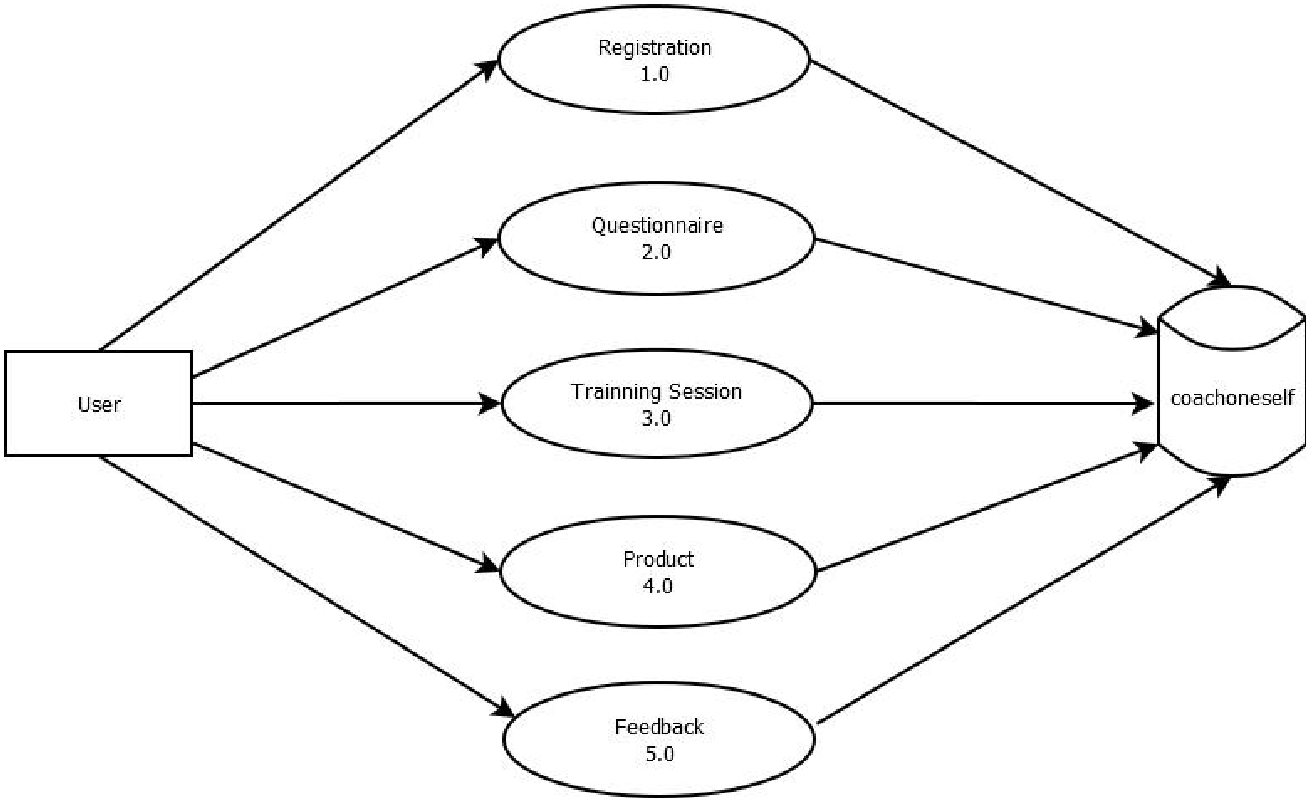
A logical file in which it is used to store table details.

Data-Flow Diagram

Level 0

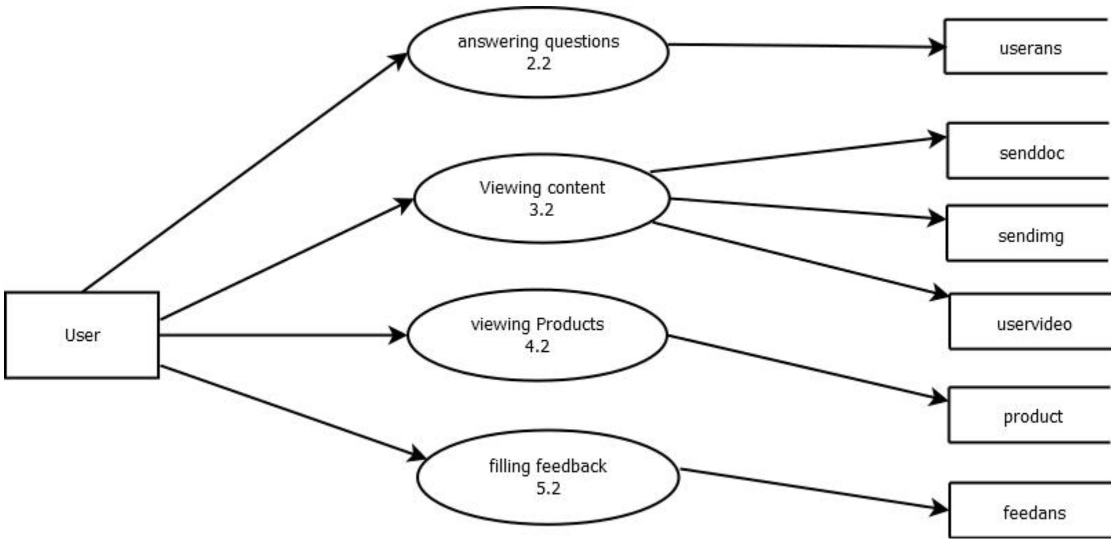


Level 1

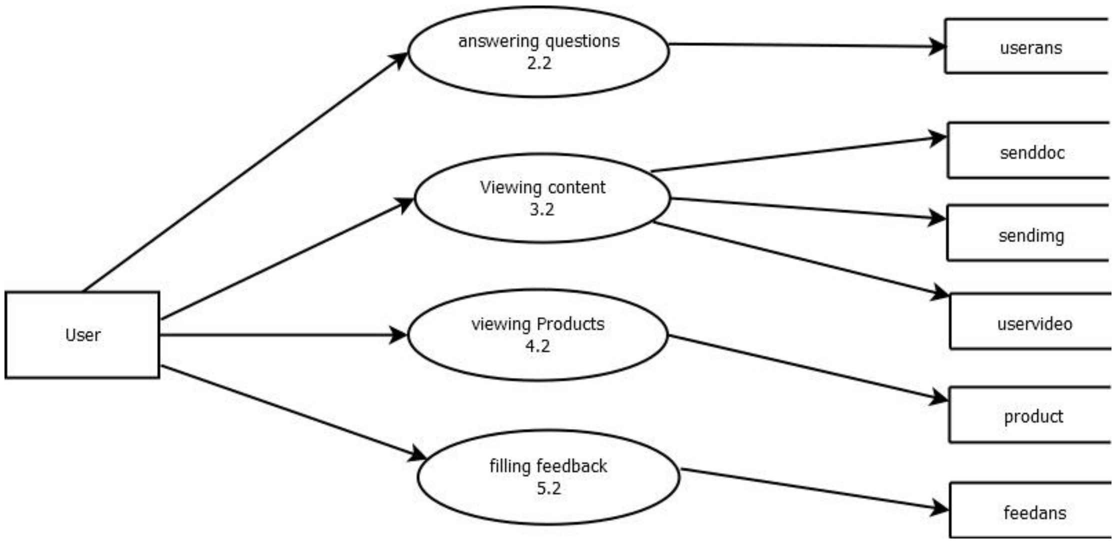


Level 2

Level 2 Admin



Level 2 User



4.3 Database design

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design of an overall database system.The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

1. Determine the data to be stored in the database.
2. Determine the relationships between the different data elements.
3. Superimpose a logical structure upon the data on the basis of these relationships.

4.3.1 List of Tables

1. user
2. login
3. product
4. questions
5. userans
6. images
7. pdf
8. videos
9. Sendimg
10. sendoc
11. uservideo
12. feedback
13. feedans

4.3.2 List of entities and attributes

Table No:1

Table name: **user**

Table Description: To store details of every users.

Field	Type	Size	Constraints	Description
uid	Int	12	Primary key NOT NULL	Unique id for users
name	Varchar	40	NOT NULL	To store name
address	Varchar	400	NOT NULL	To store address
age	Varchar	40	NOT NULL	To store age
gender	Varchar	40	NOT NULL	To store gender
email	Varchar	40	NOT NULL	To store email
contact	varchar	40	NOT NULL	To store contact

Table No: 2

Table name: **login**

Table Description: To store login details.

Fields	Type	Size	Constraints	Description
lid	Int	12	Primary key NOT NULL	Unique id for login
uid	Int	12	Foreign key NOT NULL	Unique id for users
uname	Varchar	40	NOT NULL	To store username
psw	Varchar	40	NOT NULL	To store password
type	Varchar	40	NOT NULL	
age	Varchar	40	NOT NULL	To store age
gender	varchar	40	NOT NULL	To store gender

Table No: 3

Table name: **Product**

Table Description: To store Product details

Field	**Type**	**Size**	**Constraints**	**Description**
pid	int	12	Primary key NOT NULL	Unique id for product
pname	varchar	400	NOT NULL	To store product name
description	varchar	4000	NOT NULL	To store product description
price	varchar	40	NOT NULL	To store price of product
url	varchar	400	NOT NULL	To store link for the product purchase
img	longblob		NOT NULL	To store product image
mprice	longblob		NOT NULL	To store marginal price of product

Table No: 4

Table name: **Questions**

Table Description: To store questionnaire details

Fields	**Type**	**Size**	**Constraints**	**Description**
qid	int	12	Primary key NOT NULL	Unique id for questions
question	Varchar	400	NOT NULL	To store question
a	Varchar	100	NOT NULL	To store option a
b	Varchar	100	NOT NULL	To store option b
c	Varchar	100	NOT NULL	To store option c
d	Varchar	100	NOT NULL	To store option d
age	Varchar	40	NOT NULL	To store age
gender	varchar	40	NOT NULL	To store gender

Table No: 5

Table name: **Userans**

Table Description: To store answers from user details

Fields	**Type**	**Size**	**Constraints**	**Description**
aid	int	12	Primary key NOT NULL	Unique id for answers from user
qid	varchar	50	Foreign key NOT NULL	Unique id for questions
ans	varchar	50	NOT NULL	To store answers
uid	varchar	100	NOT NULL	Unique id for users

Table No: 6

Table name: **Images**

Table Description: To store images details

Fields	**Type**	**Size**	**Constraints**	**Description**
imid	Int	11	Primary key NOT NULL	Unique id for images
name	Varchar	100	NOT NULL	To store image name
age	Varchar	40	NOT NULL	To store age of user
gender	Varchar	40	NOT NULL	To store gender of user
description	Varchar	40	NOT NULL	To store description of the image
file	longblob		NOT NULL	To store image file

Table No: 7

Table name: **Pdf**

Table Description: To store document details

Fields	**Type**	**Size**	**Constraints**	**Description**
pdid	Int	11	Primary key NOT NULL	Unique id for pdf
name	Varchar	100	NOT NULL	To store name of user
age	Varchar	40	NOT NULL	To store age of user
gender	Varchar	40	NOT NULL	To store gender of user
description	Varchar	400	NOT NULL	To store description of the pdf
file	longblob		NOT NULL	To store pdf file

Table No: 8

Table name: **Video**

Table Description: To store video details

Fields	**Type**	**Size**	**Constraints**	**Description**
vid	Int	11	Primary key NOT NULL	Unique id for video
name	Varchar	100	NOT NULL	To store name of user
age	Varchar	40	NOT NULL	To store age of user
gender	Varchar	40	NOT NULL	To store gender of user
description	Varchar	400	NOT NULL	To store description of the video
file	longblob		NOT NULL	To store video file

Table No:9

Table name: **Sendimg**

Table Description: To store image uploading details

Fields	**Type**	**Size**	**Constraints**	**Description**
smid	Int	11	Primary key NOT NULL	Unique id for uploading image
imid	Varchar	33	NOT NULL	To store id of image
uid	varchar	44	NOT NULL	To store id of user

Table No:10

Table name: **Senddoc**

Table Description: To store document uploading details

Fields	**Type**	**Size**	**Constraints**	**Description**
sdid	int	11	Primary key NOT NULL	Unique id for uploading document
did	varchar	33	NOT NULL	To store id of document
uid	varchar	33	NOT NULL	To store id of user

Table No:11

Table name: **Uservideo**

Table Description: To store uploading details of video

Fields	**Type**	**Size**	**Constraints**	**Description**
vsid	int	11	Primary key NOT NULL	Unique id for uploading video
vid	varchar	44	NOT NULL	To store id of video
uid	varchar	44	NOT NULL	To store id of user

Table No:12

Table name: **Feedback**

Table Description: To store feedback details

Fields	**Type**	**Size**	**Constraints**	**Description**
fid	Int	11	Primary key NOT NULL	Unique id for feedback
question	Varchar	100	NOT NULL	To store feedback question
a	Varchar	50	NOT NULL	To store option a
b	Varchar	50	NOT NULL	To store option b

Table No:13

Table name: **Feedans**

Table Description: To store feedback answer details

Fields	**Type**	**Size**	**Constraints**	**Description**
faid	Int	11	Primary key NOT NULL	Unique id for feedback
uid	Varchar	33	NOT NULL	To store user's id
fid	Varchar	33	NOT NULL	To store feedback id
ans	Varchar	33	NOT NULL	To store feedback answer

4.5.1 E R diagram

E-R Diagram is a logical representation of the data for an organization and uses 3 main constraints. That is, data entities, relationships and their associated attribute.

Entities:

An Entity is a fundamental thing of an organization about which data may be maintained. An entity has its own identity, which distinguishes it from each other entity. An entity type is the description of all entities to which a common definition and common relationships attributes applied.

Relationship:

A relationship is a reason for associating two entity types. These relationships are some-times called binary relationships. Because they involve two entity types. Some forms of data model allow more than two entity types to be associated.

Degree of Relationship:

The degree of a relationship is a number of entities that participate in that relationship.

The 3 most common relationships in E-R models are:

1. Unary (Degree 1)
2. Binary (Degree 2)
3. Ternary (Degree 3)

Higher degree relationships are possible but they are rarely encountered in practice.

1. Unary Relationship

This is also called recursive relationship. It is a relationship between the instances of one entity type. An instance is a single occurrence of an entity type. There may be many instances of an entity type.

2. Binary Relationship

It is a relationship between instances of two entity types and is a most common type of relationship encountered in E-R diagram.

3. Ternary Relationship

It is a simultaneous relationship between instances of 3 entity types. Relationship may have provided the association of 3 entities. All the 3 entities are many to many participants. There may be one or many participants in a ternary relationship.

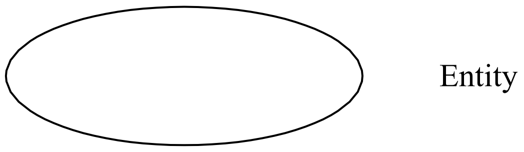
Cardinalities and Optional ties

Suppose that there are two entity types A and b, that are connected by a relationship. The cardinality of the relationship is the number of instances of entity B that can be associated with each instance of entity A. The minimum cardinality of a relationship is the minimum number of instances of an entity B that may be associated with each instance of entity A.

Attributes:

Each entity type has a set of attributes associated with it. An attribute is a property or characteristic of an entity that is of interest to the organization. We use an initial capital letter, followed by lowercase letters, and nouns in naming an attribute. In E-R diagram, we can visually represent an attribute by placing its name as an ellipse with a line connecting it to the associated entity.

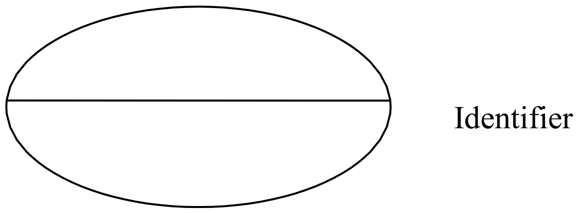
Notation for attribute is:



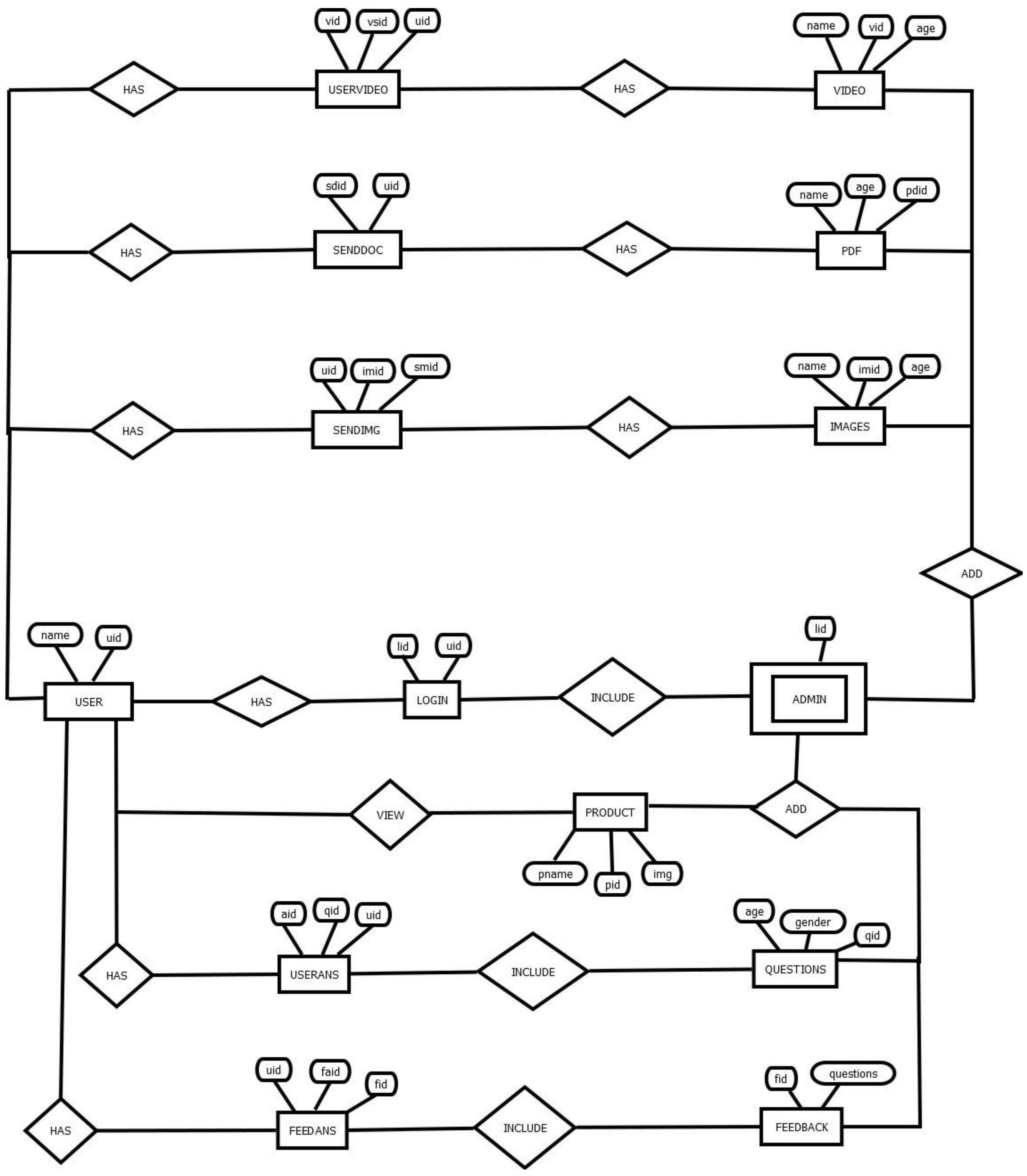
Candidate Key and Identifier:

Every entity type must have an attribute and a set of attributes that distinguish one in-stance from other instances of the same type. A candidate key is an attribute (or a combination of attributes) that uniquely identifies each instance of an entity type. If there is more than one candidate key, the designer must choose one of the candidate keys as the identifier. An identifier is a candidate key that has been selected ---to be used as the unique characteristic for an entity type.

Notation for an identifier:



ER diagram


vid
vsid
uid
USERVIDEO
HAS
HAS
name
vid
age
VIDEO
sdid
uid
SENDDOC
HAS
HAS
name
age
pdid
PDF
uid
imid
smid
SENDIMG
HAS
HAS
name
imid
age
IMAGES
ADD
name
uid
USER
HAS
lid
uid
LOGIN
INCLUDE
lid
ADMIN
VIEW
PRODUCT
ADD
pname
pid
img
aid
qid
uid
USERANS
HAS
INCLUDE
age
gender
qid
QUESTIONS
uid
faid
fid
FEEDANS
HAS
INCLUDE
fid
questions
FEEDBACK

DEVELOPMENT AND IMPLEMENTATION

5.DEVELOPMENT

Software development is the process of Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components. Software development involves writing and maintaining the source code, but in a broader sense, it includes all processes from the conception of the desired software through to the final manifestation of the software, typically in a planned and structured process. Software development also includes research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.

5.1 Tools

NetBeans IDE:

NetBeans IDE is a free and open source integrated development environment for application development on Windows, Mac, Linux, and Solaris operating systems.The IDE simplifies the development of web, enterprise, desktop, and mobile applications that use the Java and HTML5 platforms. The IDE also offers support for the development of PHP and C/C++ applications.

JSP:

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

JDK:

The Java Development Kit (JDK) is a cross-platformed software development environment that offers a collection of tools and libraries necessary for developing Java-based software applications and applets. It is a core package used in Java, along with the JVM and the JRE (Java Runtime Environment). The JDK has a private Java Virtual Machine (JVM) and a few other resources necessary for the development of a Java Application.

JAVA :

Java is an object oriented, class-based, concurrent, secured and general-purpose computer-programming language. It is a widely used robust technology. Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language.

HTML :

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML is being widely used to format web pages with the help of different tags available in HTML language. HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

CSS :

CSS is the language we use to style an HTML document.CSS describes how HTML elements should be displayed. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

JAVASCRIPT :

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

XAMP :

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself.

MYSQL:

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database.

It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

SQLyog :

SQLyog is a powerful GUI tool to manage MySQL and MariaDB servers and databases in physical, virtual, and cloud environments. SQLyog runs on Microsoft Windows. It may also run on Linux and Unix via Wine.

5.2 Modules

1. Registration Module

In this Module the User registers with their name ,age ,gender ,email , password , contact , address and aadhar. Both users admin and end user can login.

2. Questionnaire Module

In this Module the Admin add questionnaire for users to fill and user fills the questionnaire to get data.

3. Product Module

In this module Admin adds self-defence products. Both admin and user can view and access self-defence tools.

4. Training session Module

In this Module Admin verifies the user using their aadhar number and then uploads data after verification. Data is uploaded by accessing their questionnaire answers.

5. Feedback Module

In this Module User can change the data provided by admin by filling the feedback questionnaire.

VALIDATION CRITERIA AND IMPLEMENTATION

VALIDATION CRITERIA AND IMPLEMENTATION

Software validation checks that the software product satisfies or fits the intended needs of all the stakeholders.It is achieved through a series of test that demonstrate conformity with requirements.A test plan outlines the classes of tests to be conducted and a test procedures defines specific test cases that are designed to ensure that alla functional requirements are satisfied,all behavioural characteristics are achieved,all content is accurate and properly presented,all performance requirements are attained,documentation is correct, and usability and other requirements are met .

6.1 Test plans

A test plan is a detailed document that describes the test strategy, objectives, schedule,estimation and deliverables and resources required for testing. Test Plan helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager. It helps people outside the test team such as developers, business managers, customers understand the details of testing. Test Plan guides our thinking. It is like a rule book, which needs to be followed.Important aspects like test estimation, test scope, Test Strategy are documented in Test Plan, so it can be reviewed by Management Team and re-used for other projects.

6.2 Classifications of testing

Unit testing :

Unit testing is defined as a type of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed A unit is the smallest testable part of any software] In procedural programming) a unit could be an entire module, but it is more commonly an individual function or procedure In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method Unit tests forms the basis for component testing.Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.In the case of unit testing of modules, different modules are tested against the specifications produced during design for the modules. Therefore, the unit testing of modules is essentially for verification of the code produced during the coding phase It focuses on the internal processing logic and data structures within the boundaries of a component.

White box testing:

White box testing (also known as glass box testing or structural testing is a software testing method that tests internal structures or workings of a module, as opposed to its functionality (i.e. black-box testing).

White box testing is concerned with testing the implementation of the program- different programming structures and data structures used in the program It is basically a test case design method that uses the control structure of the procedural design to derive test cases.Unit testing of components is performed by it.

Integration testing:

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as I & T (Integration and Testing).Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit Testing.

Black-box testing- Black-box testing focuses on the functional requirements of the software. In black-box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or specifications of the program or module, and the internals of the module or the program are not considered for selection of test cases. In black-box testing, the tester only knows the inputs that can be given to the system and what output the system should give. In other words, the basis for deciding test cases in black box testing is the requirements or specifications of the system or module. This form of testing is also called functional or behavioural testing.

Usability Testing - Usability testing mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives.

Load Testing - Load Testing is necessary to know that a software solution will perform under real-life loads.

Regression Testing - Regression Testing involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.

Recovery Testing -Recovery testing is done to demonstrate a software solution is reliable, trustworthy and can successfully recover from possible crashes.

Migration Testing - Migration testing is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

Functional Testing - Functional Testing involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.

Security Testing- Any computer-based system that manages sensitive information or causes actions that can improperly harm (or benefit) individuals is a target for improper or illegal penetration. Penetration spans a broad range of activities: hackers who attempt to penetrate systems for sport, dissatisfied employees who attempt to penetrate for revenge, dishonest individuals who attempt to penetrate for illicit personal gain.

Stress Testing- Stress tests are designed to confront programs with abnormal situations. Stress testing executes a system in a manner that demands resources in abnormal quantity, frequency, or volume.

Performance Testing- Performance testing is designed to test the run-time performance of software. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as tests are conducted.

Deployment Testing- Deployment testing, sometimes called configuration testing, exercises the software in each environment in which it is to operate. In addition, deployment testing examines all installation procedures and specialized installation software (e.g.. "installers") that will be used by customers and all documentation that will be used to introduce the software to end users.

Hardware/Software Testing - This is when the tester focuses attention on the interactions between the hardware and software during system testing.

Smoke testing- also known as build verification testing, is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. The result of this testing is used to decide if a build is stable enough to proceed with further testing.

6.3 Implementation

Software implementation is the process of integrating an application into an organization's workflow.Implementation simply means carrying out the activities described in your work plan.
It's important to take into account that independently of the nature of the project, implementation takes time ,usually more than it is planned,and that many external constraints can appear,which should be considered when initiating the implementation step.

Here the implementation of our project includes the training session that are uploaded by the admin and the users can easily access them accordingly.This is what happens in implementation phase of the project life cycle. The implementation phase is where you and your project team actually do the project work to produce the training session.

CONCLUSION

CoachOneSelf aim is to provide personalized self-defense to everyone according to one's respective age, gender, skills and health.

In today's world with the increase rate of attacks and crimes learning self-defense should be a vital part of everyone's life. The issues starting from abuses to murders are faced by individuals in all age groups and gender. For instance child sexual abuse, elder abuse, rapes, anxiety, robbery, etc.

This project will help people to learn self-defense and hence becoming stronger mentally and physically. Coach Oneself website provides desired contents to one's choice and ability as a result : it avoids injuries, make them confident to learn and to implement , easy to understand , knowledge about self-defense tools and improvement in ones growth. Self-defense definitely will help people to be prepared and to react strongly to any situation they face.

BIBLIOGRAPHY

Links:

https://www.javatpoint.com/java-tutorial

https://www.tutorialspoint.com/java/index.htm

Books:

"The Complete Reference,Seventh Edition",by Herbert Schildt,MC Graw Hill.

"Software Engineering",by Dr.Abbas T.P. ,Calicut University.

CODE

AddDocAction.jsp

```
<%@page import="dbconnection.dbconnection"%>
<%@page import="java.io.File"%>
<%@page import="java.util.Iterator"%>
<%@page
import="org.apache.commons.fileupload.FileUploadException
"%>
<%@page
import="org.apache.commons.fileupload.disk.DiskFileItemFa
ctory"%>
<%@page import="java.util.List"%>
<%@page
import="org.apache.commons.fileupload.FileItemFactory"%>
<%@page
import="org.apache.commons.fileupload.servlet.ServletFileUp
load"%>
<%@page
import="org.apache.commons.fileupload.FileItem"%>
<%

    FileItem f_item = null;
    String file_name = "";
    dbconnection con = new dbconnection();
    String pname = "", age = "", gender = "", description = "";
    boolean isMultipart =
ServletFileUpload.isMultipartContent(request);
    System.out.println("111");
    if (isMultipart) {
        FileItemFactory factory = new
```

```
DiskFileItemFactory();
            ServletFileUpload upload = new ServletFileUpload(factory);
            //declaring a list of form fields
            List items_list = null;
            //assigning fields to list 'items_list'
            try {
                  items_list = upload.parseRequest(request);
            } catch (FileUploadException ex) {
                  out.println(ex);

    }
          //declaring iterator
          Iterator itr = items_list.iterator();
          //iterating through the list 'items_list'
          if (itr.hasNext()) {
                //typecasting next element in items_list as fileitem
                f_item = (FileItem) itr.next();
                //checking if 'f_item' contains a formfield(common controls like textbox,dropdown,radio buttonetc)
                while (f_item.isFormField()) {
                      //getting fieldname and value
                      String field = f_item.getFieldName();
                      String value = f_item.getString();
                      if (field.equalsIgnoreCase("pname")) {
                            pname = value;
                      }
                      if (field.equalsIgnoreCase("age")) {
                            age = value;
                      }
                      if (field.equalsIgnoreCase("gender")) {
                            gender = value;
                      }
                      if (field.equalsIgnoreCase("description")) {
                      description = value;
                      }


                      f_item = (FileItem) itr.next();
                }
                //else part does the image upload
                file_name = f_item.getName();
                //setting path to store image
                File proj_path = new File(config.getServletContext().getRealPath("/"));
                String file_path = proj_path.getParentFile().getParentFile().getPath() + "\\web\\PrImg\\";
                //creating a file object
                File savedFile = new File(file_path + file_name);
                try {
                      //writing the file object
                      f_item.write(savedFile);
                } catch (Exception ex) {
                      out.println(ex);
                }
          }
    }

    String userid = (String) session.getAttribute("USERID");
    String insert = "insert into `pdf`(`name`,`age`,`gender`,`description`,`file`)VALUES('" + pname + "','" + age +
  "','" + gender + "','" + description + "','" + file_name + "')";
    System.out.println(insert);
    if (con.putData(insert) > 0) {
%>
< script>
    alert("New Document Added Successfully");
    window.location = "../AdminAddPdf.jsp";
</script>
<%
} else {
```

```
%>
<script>
    alert("Failed To Add Document");
    window.location = "../AdminAddPdf.jsp";
</script>
<%
    }
%>
```

AddFeedQueAction.jsp

```
<%@page import="dbconnection.dbconnection"%>
<%
dbconnection con=new dbconnection();
String qu=request.getParameter("qu");
String a=request.getParameter("a");
String b=request.getParameter("b");
String qry="INSERT INTO `feedback`(`question`,`a`,`b`)VALUES('"+qu+"','"+a+"','"+b+"')";
if (con.putData(qry)>0) {
        %>
<script>
        alert("New Question Added Successfully");
        window.location = "../AdminAddFeedQue.jsp";
</script>
<%
        }
else
{
%>
<script>
        alert("Failed to add new question");
        window.location = "../AdminAddFeedQue.jsp";
</script>
<%
```

```
}      %>
<script>
        alert("New Question Added Successfully");
        window.location = "../AdminAddFeedQue.jsp";
</script>
%>AddImageAction.jsp
<%@page import="dbconnection.dbconnection"%>
<%@page import="java.io.File"%>
<%@page import="java.util.Iterator"%>
<%@page import="org.apache.commons.fileupload.FileUploadException"%>
<%@page import="org.apache.commons.fileupload.disk.DiskFileItemFactory"%>
<%@page import="java.util.List"%>
<%@page import="org.apache.commons.fileupload.FileItemFactory"%>
<%@page import="org.apache.commons.fileupload.servlet.ServletFileUpload"%>
<%@page import="org.apache.commons.fileupload.FileItem"%>
<%

        FileItem f_item = null;
        String file_name = "";
        dbconnection con = new dbconnection();
        String pname = "", age = "", gender = "", description = "";
        boolean isMultipart = ServletFileUpload.isMultipartContent(request);
        System.out.println("111");
        if (isMultipart) {
                FileItemFactory factory = new DiskFileItemFactory();
                ServletFileUpload upload = new ServletFileUpload(factory);
                //declaring a list of form fields
                List items_list = null;
                //assigning fields to list 'items_list'
                try {
                        items_list = upload.parseRequest(request);
                } catch (FileUploadException ex) {
                        out.println(ex);
```

```
}
        //declaring iterator
        Iterator itr = items_list.iterator();
        //iterating through the list 'items_list'
        if (itr.hasNext()) {
            //typecasting next element in items_list as fileitem
            f_item = (FileItem) itr.next();
            //checking if 'f_item' contains a formfield(common controls like textbox,dropdown,radio buttonetc)
            while (f_item.isFormField()) {
                //getting fieldname and value
                String field = f_item.getFieldName();
                String value = f_item.getString();
                if (field.equalsIgnoreCase("pname")) {
                    pname = value;
                }
                if (field.equalsIgnoreCase("age")) {
                    age = value;
                }
                if (field.equalsIgnoreCase("gender")) {
                    gender = value;
                }
                if (field.equalsIgnoreCase("description")) {
                    description = value;
                }

                f_item = (FileItem) itr.next();
            }
            //else part does the image upload
            file_name = f_item.getName();
            //setting path to store image
            File proj_path = new File(config.getServletContext().getRealPath("/"));
            String file_path = proj_path.getParentFile().getParentFile().getPath() + "\\web\\PrImg\\";
            //creating a file object
```

```
File savedFile = new File(file_path + file_name);
                        try {
                                //writing the file object
                                f_item.write(savedFile);
                        } catch (Exception ex) {
                                out.println(ex);
                        }
                }
        }

        String userid = (String) session.getAttribute("USERID");
        String insert = "INSERT INTO `images`(`name`,`age`,`gender`,`description`,`file`)VALUES('" + pname +
"','" + age + "','" + gender + "','" + description + "','" + file_name + "')";
        System.out.println(insert);
        if (con.putData(insert) > 0) {
%>
<script>
        alert("New Image Added Successfully");
        window.location = "AdminAddImages.jsp";
</script>
<%
} else {
%>
<script>
        alert("Failed To Add Image");
        window.location = "AdminAddImages.jsp";
</script>
<%
        }
%>
```

AddProductAction.jsp

```
<%@page import="dbconnection.dbconnection"%>
<%@page import="java.io.File"%>
<%@page import="java.util.Iterator"%>

<%@page import="org.apache.commons.fileupload.FileUploadException"%>
<%@page import="org.apache.commons.fileupload.disk.DiskFileItemFactory"%>
<%@page import="java.util.List"%>
<%@page import="org.apache.commons.fileupload.FileItemFactory"%>
<%@page import="org.apache.commons.fileupload.servlet.ServletFileUpload"%>
<%@page import="org.apache.commons.fileupload.FileItem"%>
<%

    FileItem f_item = null;
    String file_name = "";
    dbconnection con = new dbconnection();
    String pname = "", age = "", gender = "", description = "";
    boolean isMultipart = ServletFileUpload.isMultipartContent(request);
    System.out.println("111");
    if (isMultipart) {
        FileItemFactory factory = new DiskFileItemFactory();
        ServletFileUpload upload = new ServletFileUpload(factory);
        //declaring a list of form fields
        List items_list = null;
        //assigning fields to list 'items_list'
        try {
            items_list = upload.parseRequest(request);
        } catch (FileUploadException ex) {
            out.println(ex);
        }
        //declaring iterator
        Iterator itr = items_list.iterator();
```

```
//iterating through the list 'items_list'
        if (itr.hasNext()) {
            //typecasting next element in items_list as fileitem
            f_item = (FileItem) itr.next();
            //checking if 'f_item' contains a formfield(common controls like textbox,dropdown,radio buttonetc)
            while (f_item.isFormField()) {
                //getting fieldname and value
                String field = f_item.getFieldName();
                String value = f_item.getString();
                if (field.equalsIgnoreCase("pname")) {
                    pname = value;
                }
                if (field.equalsIgnoreCase("age")) {
                    age = value;
                }
                if (field.equalsIgnoreCase("gender")) {
                    gender = value;
                }
                if (field.equalsIgnoreCase("description")) {
                    description = value;
                }

                f_item = (FileItem) itr.next();
            }
            //else part does the image upload
            file_name = f_item.getName();
            //setting path to store image
            File proj_path = new File(config.getServletContext().getRealPath("/"));
            String file_path = proj_path.getParentFile().getParentFile().getPath() + "\\web\\PrImg\\";
            //creating a file object
            File savedFile = new File(file_path + file_name);
            try {
                //writing the file object
```

```
                    f_item.write(savedFile);
                     } catch (Exception ex) {
                            out.println(ex);
                     }
               }
        }

        String userid = (String) session.getAttribute("USERID");
          String insert = "INSERT INTO `images`(`name`,`age`,`gender`,`description`,`file`)VALUES('" + pname +
"','" + age + "','" + gender + "','" + description + "','" + file_name + "')";
        System.out.println(insert);
        if (con.putData(insert) > 0) {
%>
<script>
        alert("New Image Added Successfully");
        window.location = "AdminAddImages.jsp";
</script>
<%
} else {
%>
<script>
        alert("Failed To Add Image");
        window.location = "AdminAddImages.jsp";
</script>
<%
        }
%>
```

AddQuestionsAction.jsp

```
<%@page import="java.util.Iterator"%>
<%@page import="java.util.Vector"%>
<%@page import="dbconnection.dbconnection" %>
<%
    dbconnection con=new dbconnection();
    String qu=request.getParameter("qu");
      String a=request.getParameter("a");
      String b=request.getParameter("b");
        String c=request.getParameter("c");
      String d=request.getParameter("d");
        String age=request.getParameter("age");
          String gender=request.getParameter("gender");
    String qry="INSERT INTO
`questions`(`question`,`a`,`b`,`c`,`d`,`age`,`gender`)VALUES('
"+qu+"','"+a+"','"+b+"','"+c+"','"+d+"','"+age+"','"+gender+"')";
if(con.putData(qry)>0)
{
    %>
    <script>
        alert("New Question Added Successfully");
          window.location="../AdminAddQuestions.jsp";
          </script>
          <%
              }
```

```
else
{
        %>
        <script>
                alert("Failed to add new Question");
                window.location="../AdminAddQuestions.jsp";
                </script>
                <%
}
%>
```

AddVideoAction.jsp

```
<%@page import="dbconnection.dbconnection"%>
<%@page import="java.io.File"%>
<%@page import="java.util.Iterator"%>
<%@page
import="org.apache.commons.fileupload.FileUploadException
"%>
<%@page
import="org.apache.commons.fileupload.disk.DiskFileItemFa
ctory"%>
<%@page import="java.util.List"%>
<%@page
import="org.apache.commons.fileupload.FileItemFactory"%>
<%@page
import="org.apache.commons.fileupload.servlet.ServletFileUp
load"%>
```

```
<%@page
import="org.apache.commons.fileupload.FileItem"%>
<%      FileItem f_item = null;
        String file_name = "";
        dbconnection con = new dbconnection();
        String pname = "", age = "", gender = "", description = "";
        boolean isMultipart =
ServletFileUpload.isMultipartContent(request);
        System.out.println("111");
        if (isMultipart) {
                FileItemFactory factory = new
DiskFileItemFactory();
                ServletFileUpload upload = new
ServletFileUpload(factory);
                //declaring a list of form fields
                List items_list = null;
                //assigning fields to list 'items_list'
                try {
                        items_list = upload.parseRequest(request);
                } catch (FileUploadException ex) {
                        out.println(ex);
                }
                //declaring iterator
                Iterator itr = items_list.iterator();
                //iterating through the list 'items_list'
                if (itr.hasNext()) {
```

```
   //typecasting next element in items_list as fileitem
                    f_item = (FileItem) itr.next();
                    //checking if 'f_item' contains a
formfield(common controls like textbox,dropdown,radio
buttonetc)
                    while (f_item.isFormField()) {
                        //getting fieldname and value
                        String field = f_item.getFieldName();
                        String value = f_item.getString();
                        if (field.equalsIgnoreCase("pname")) {
                            pname = value;
                        }
                        if (field.equalsIgnoreCase("age")) {
                            age = value;
                        }
                        if (field.equalsIgnoreCase("gender")) {
                            gender = value;
                        }
                        if (field.equalsIgnoreCase("description"))
{
                            description = value;
                        }

                        f_item = (FileItem) itr.next();
                    }
                    //else part does the image upload
```

```
file_name = f_item.getName();
                //setting path to store image
                File proj_path = new
File(config.getServletContext().getRealPath("/"));
                String file_path =
proj_path.getParentFile().getParentFile().getPath() +
"\\web\\Videos\\";
                //creating a file object
                File savedFile = new File(file_path +
file_name);
                try {
                    //writing the file object
                    f_item.write(savedFile);
                } catch (Exception ex) {
                    out.println(ex);
                }
            }
        }

        String userid = (String) session.getAttribute("USERID");
        String insert = "INSERT INTO
`video`(`name`,`age`,`gender`,`description`,`file`)VALUES('"
+ pname + "','" + age + "','" + gender + "','" + description + "','"
+ file_name + "')";
        System.out.println(insert);
        if (con.putData(insert) > 0) {
%>
```

```
<script>
    alert("New Video Added Successfully");
    window.location = "../AdminAddVideo.jsp";
</script>
<%
} else {
%>
<script>
    alert("Failed To Add Video");
    window.location = "../AdminAddVideo.jsp";
</script>
<%
    }
%>
```

AnsQuestions.jsp

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<%@page import="dbconnection.dbconnection"%>
<%


    dbconnection con = new dbconnection();


    String qid = (String) session.getAttribute("QID");
    String ans = request.getParameter("ans");
    String userid = (String) session.getAttribute("USERID");
```

```
 String sqry = "SELECT COUNT(*) AS COUNT FROM
`userans` WHERE `qid`='" + qid + "' AND `uid`='" + userid +
"'";
        System.out.println("qry" + sqry);
        Iterator it = con.getData(sqry).iterator();
        if (it.hasNext()) {


            Vector v = (Vector) it.next();
            int max_vid = Integer.parseInt(v.get(0).toString());
            System.out.println(max_vid);


            if (max_vid == 0) {
                String qry = "INSERT INTO
`userans`(`qid`,`ans`,`uid`)VALUES('" + qid + "','" + ans + "','"
+ userid + "')";
                System.out.println("ins" + qry);


                if (con.putData(qry) > 0) {
%>
<script>
        alert("Data Updated Successfully");
        window.location = "../UserViewAllQue.jsp";
</script>
<%
} else {
%>
```

```
<script>
    alert("Failed to update !!");
    window.location = "../UserViewAllQue.jsp";
</script>
<%
    }


} else {


    String dqry = "update `userans` set `ans`='" + ans + "'
where `qid`='" + qid + "'";
    System.out.println("del" + dqry);
    if (con.putData(dqry) > 0) {
        System.out.println("success");
%>
<script>
    alert("Data Updated Successfully");
    window.location = "../UserViewAllQue.jsp";
</script>
<%
} else {
%>
<script>
    alert("Failed to update !!");
    window.location = "../UserViewAllQue.jsp";
```

```
</script>
<%
                    }
               }
        }

%><a href="../UserViewAllQue.jsp">
```

FeedBackAns.jsp

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<%@page import="dbconnection.dbconnection"%>
<%

        dbconnection con = new dbconnection();

        String qid = (String) session.getAttribute("FID");
        String ans = request.getParameter("ans");
        String userid = (String) session.getAttribute("USERID");

        String sqry = "SELECT COUNT(*) AS COUNT FROM
`feedans` WHERE `fid`='" + qid + "' AND `uid`='" + userid +
"'";
        System.out.println("qry" + sqry);
        Iterator it = con.getData(sqry).iterator();
        if (it.hasNext()) {
```

```
Vector v = (Vector) it.next();

            int max_vid = Integer.parseInt(v.get(0).toString());

            System.out.println(max_vid);

            if (max_vid == 0) {

                  String qry = "INSERT INTO
`feedans`(`uid`,`fid`,`ans`)VALUES('" + userid + "','" + qid +
"','" + ans + "')";

                  System.out.println("ins" + qry);


                  if (con.putData(qry) > 0) {

%>

<script>

      alert("Feedback Added Successfully");

      window.location = "../UserViewFeed.jsp";

</script>

<%

} else {

%>

<script>

      alert("Failed to Add !!");

      window.location = "../UserViewFeed.jsp";

</script>

<%

      }


} else {
        String dqry = "update `feedans` set `ans`='" + ans + "'
where `fid`='" + qid + "'";
        System.out.println("del" + dqry);
        if (con.putData(dqry) > 0) {
                System.out.println("success");
%>
<script>
        alert("Feedback Updated Successfully");
        window.location = "../UserViewFeed.jsp";
</script>
<%
} else {
%>
<script>
        alert("Failed to update !!");
        window.location = "../UserViewFeed.jsp";
</script>
<%
                        }


                }
        }


%><a href="../UserViewFeed.jsp">
```

LoginAction.jsp

```
<%@page
import="dbconnection.dbconnection"%>
<%@page
import="javax.swing.JOptionPane"%>
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<%@page contentType="text/html"
pageEncoding="UTF-8"%>


<%
    dbconnection con = new dbconnection();
    String uname =
request.getParameter("email");
    String psw =
request.getParameter("Password");
    String n, userid, age, gender;
    String qry = "select * from login where
uname='" + uname + "'and psw='" + psw +
"'";
    System.out.println("qry::" + qry);
    Iterator i = con.getData(qry).iterator();
    if (i.hasNext()) {
        Vector v = (Vector) i.next();
        n = v.get(4).toString();
        userid = v.get(1).toString();
```

```
session.setAttribute("USERID", userid);
            age = v.get(5).toString();
            session.setAttribute("AGE", age);
            gender = v.get(6).toString();
            session.setAttribute("GENDER",
gender);
            if (n.trim().equals("admin")) {

response.sendRedirect("../AdminHome.jsp");
            } else if (n.trim().equals("user")) {
%>
<script>
      alert("Login Successful");
      window.location = "../UserHome.jsp";
</script>
<%
      } else {
            System.out.println("invalid user
name or password");
      }
} else {
%>
<script>
      alert("Invalid username or password
||Request in progress    ");
      window.location = "../Login.jsp";
```

```
</script>
<%
        }
%>
```

RemoveUser.jsp

```
<%@page
import="dbconnection.dbconnection"%>
<%
        dbconnection con = new dbconnection();


        String id = request.getParameter("id");


        String qry = "DELETE FROM `login`
WHERE `uid`='" + id + "'";


        if (con.putData(qry) > 0) {
%>
<script>
        alert("Removed user Successfully");
        window.location =
"../AdminViewUsers.jsp";
</script>
<%
} else {
```

```
%>
<script>
        alert("Failed");
        window.location =
"../AdminViewUsers.jsp";
</script>
<%      }


%>
```

SendImagesAction.jsp

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<%@page
import="dbconnection.dbconnection"%>
<%


dbconnection con=new dbconnection();
String Uid = (String)
session.getAttribute("USERID");
String id=request.getParameter("id");
String insertqry = "SELECT COUNT(*) AS
COUNT FROM `sendimg` WHERE
`imid`='"+id+"' AND `uid`='"+Uid+"'";
        System.out.println(insertqry);
        Iterator it =
con.getData(insertqry).iterator();
```

```
  if (it.hasNext()) {

              Vector v = (Vector) it.next();

              int max_vid =
Integer.parseInt(v.get(0).toString());

              System.out.println(max_vid);

              if (max_vid == 0) {

String qry="INSERT INTO
`sendimg`(`imid`,`uid`)VALUES('"+id+"','"
+Uid+"')";

if (con.putData(qry)>0) {


        %>

<script>

        alert("Image Sent");


        window.location =
"../AdminViewImage.jsp";

</script>

<%


        }

else

{

%>

<script>

        alert("Failed To Send");
        window.location =
"../AdminViewImage.jsp";
</script>
<%
}
}
else
{
%>
<script>
        alert("Image Already Send");


        window.location =
"../AdminViewImage.jsp";
</script>
<%
}
}


%>
```

SendPdfAction.jsp

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<%@page
import="dbconnection.dbconnection"%>
<%
```

```
dbconnection con=new dbconnection();
String Uid = (String)
session.getAttribute("USERID");
String id=request.getParameter("id");
String insertqry = "SELECT COUNT(*) AS
COUNT FROM `senddoc` WHERE
`did`='"+id+"' AND `uid`='"+Uid+"'";
        System.out.println(insertqry);
        Iterator it =
con.getData(insertqry).iterator();
        if (it.hasNext()) {
                Vector v = (Vector) it.next();
                int max_vid =
Integer.parseInt(v.get(0).toString());
                System.out.println(max_vid);
                if (max_vid == 0) {
String qry="INSERT INTO
`senddoc`(`did`,`uid`)VALUES('"+id+"','"+
Uid+"')";
if (con.putData(qry)>0) {


        %>
<script>
        alert("Document Sent");
        window.location =
"../AdminViewPdf.jsp";
</script>
<%

        }
else
{
%>
<script>
        alert("Failed To Send");

        window.location =
"../AdminViewPdf.jsp";
</script>
<%
}
}
else
{
%>
<script>
        alert("Document Already Send");

        window.location =
"../AdminViewPdf.jsp";
```

```
</script>
<%
}
}
%>
```

SendVideosAction.jsp

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<%@page
import="dbconnection.dbconnection"%>
<%


        dbconnection con = new dbconnection();
        String Uid = (String)
session.getAttribute("USERID");
        String id = request.getParameter("id");
        String insertqry = "SELECT COUNT(*)
AS COUNT FROM `uservideo` WHERE
`vid`='" + id + "' AND `uid`='" + Uid + "'";
        System.out.println(insertqry);
        Iterator it =
con.getData(insertqry).iterator();
        if (it.hasNext()) {
                Vector v = (Vector) it.next();
                int max_vid =
Integer.parseInt(v.get(0).toString());
```

```
System.out.println(max_vid);
            if (max_vid == 0) {
                  String qry = "INSERT INTO
`uservideo`(`vid`,`uid`)VALUES('" + id +
"','" + Uid + "')";
                  if (con.putData(qry) > 0) {


%>
<script>
      alert("Video Sent");


      window.location =
"../AdminViewVideos.jsp";
</script>
<%} else {
%>
<script>
      alert("Failed To Send");


      window.location =
"../AdminViewVideos.jsp";
</script>
<%
      }
} else {
%>
<script>
        alert("Video Already Send");


        window.location =
"../AdminViewVideos.jsp";
</script>
<%
                }
        }


%>
```

UserRegAction.jsp

```
<%@page import="dbconnection.Otp"%>
<%@page
import="dbconnection.dbconnection"%>
<%@page import="java.util.Random"%>
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>


<%
        dbconnection con = new dbconnection();
        System.out.println("in proregister
page...");
        String name =
request.getParameter("name");
        String address =
        request.getParameter("address");
        String email =
request.getParameter("email");
        String phone =
request.getParameter("contact");
        String age =
request.getParameter("age");
        String gender =
request.getParameter("gender");
        String psw =
request.getParameter("psw");
        String aadhar =
request.getParameter("aadhar");



          String MSG = "Your registration is
successful. Your user name is '" + email + "'
and password is '" + psw + "' LCCEKM";


        String insertqry = "SELECT COUNT(*)
AS COUNT FROM `user` WHERE
`email`='" + email + "' OR `contact`='" +
phone + "' OR `aadhar`='" + aadhar + "'";
        System.out.println(insertqry);
        Iterator it =
con.getData(insertqry).iterator();
        if (it.hasNext()) {
```

```
Vector v = (Vector) it.next();
            int max_vid =
Integer.parseInt(v.get(0).toString());
            System.out.println(max_vid);


            if (max_vid == 0) {


    String qry = "INSERT INTO
  `user`(`name`,`address`,`age`,`gender`,`ema
  il`,`contact`,`aadhar`) VALUES('" + name
  + "', '" + address + "','" + age + "', '" +
  gender + "', '" + email + "', '" + phone +
  "','"+aadhar+"')";
                  String qry1 = "INSERT
INTO
`login`(`uid`,`uname`,`psw`,`type`,`age`,`gen
der`) VALUES((SELECT MAX(`uid`)
FROM `user`), '" + email + "', '" + psw + "',
'user', '0','"+gender+"')";
                  if (con.putData(qry) > 0 &&
con.putData(qry1) > 0) {
                        System.out.println("Data
inserted to User table");
                      int
a=Integer.parseInt(age);
                      if (a<=18) {
                                String qry2
```

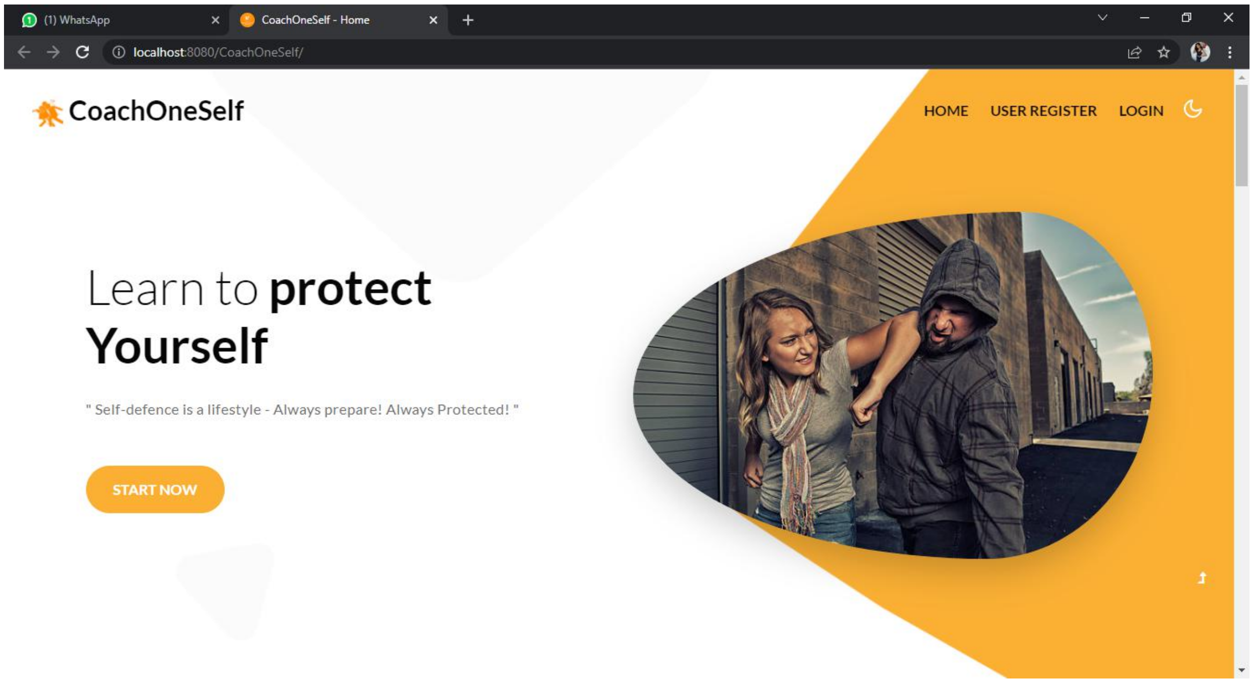
```
="UPDATE `login` SET `age`='a' WHERE
`uname`='"+email+"'";

if(con.putData(qry2)>0)
                                    {

System.out.println("Successfull");
                                    }
                           }
                     else if(a>18 && a<45){
                           String qry3
="UPDATE `login` SET `age`='b' WHERE
`uname`='"+email+"'";

if(con.putData(qry3)>0)
                                    {

System.out.println("Successfull");
                                    }
                     }
                     else if(a>=45)
                     {
                           String qry4
="UPDATE `login` SET `age`='c' WHERE
`uname`='"+email+"'";

if(con.putData(qry4)>0)
```

```
{

System.out.println("Successfull");
                                        }
                        }

%>

<script>
                   alert("New User Added
sucesfully");
      window.location = "../UserRegister.jsp";

</script>

<% } else {

%>
<script>
      alert("Unsuccessful");

      window.location = "../UserRegister.jsp";
</script>
<% }
} else {
```

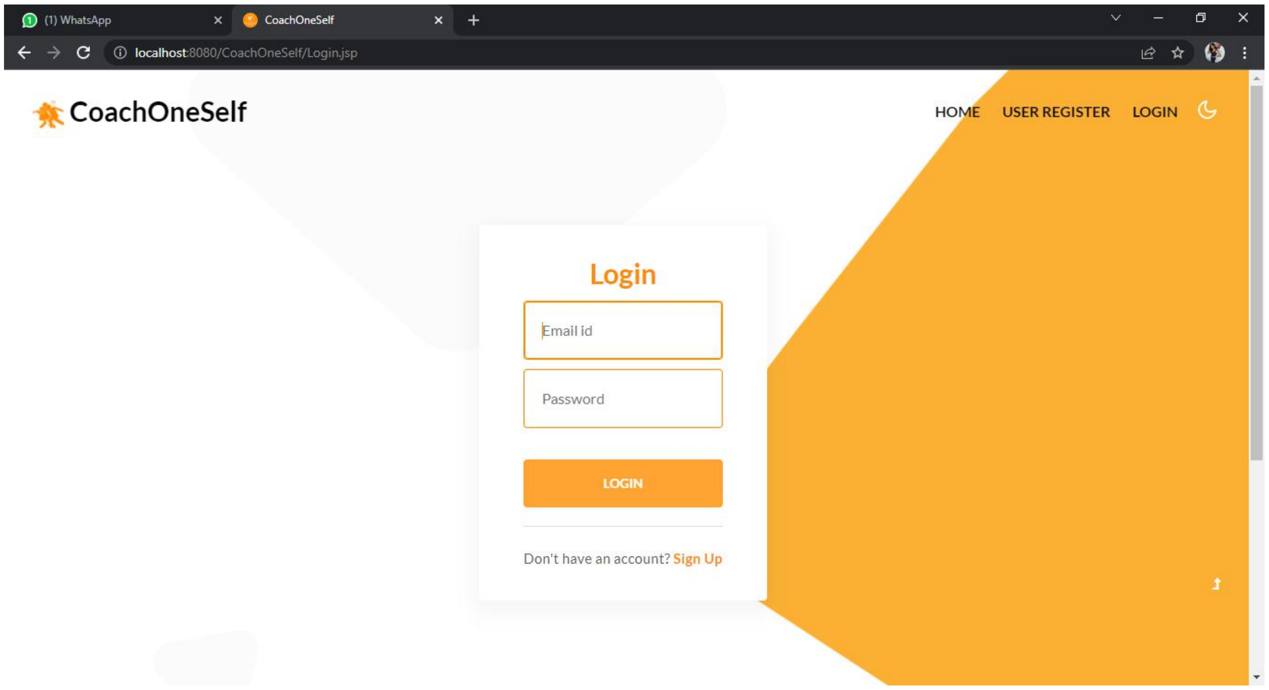
```
%>
<script>
        alert("Account Already Exist");


        window.location = "../UserRegister.jsp";
</script>
<% }
        }
%>
```

User Interface

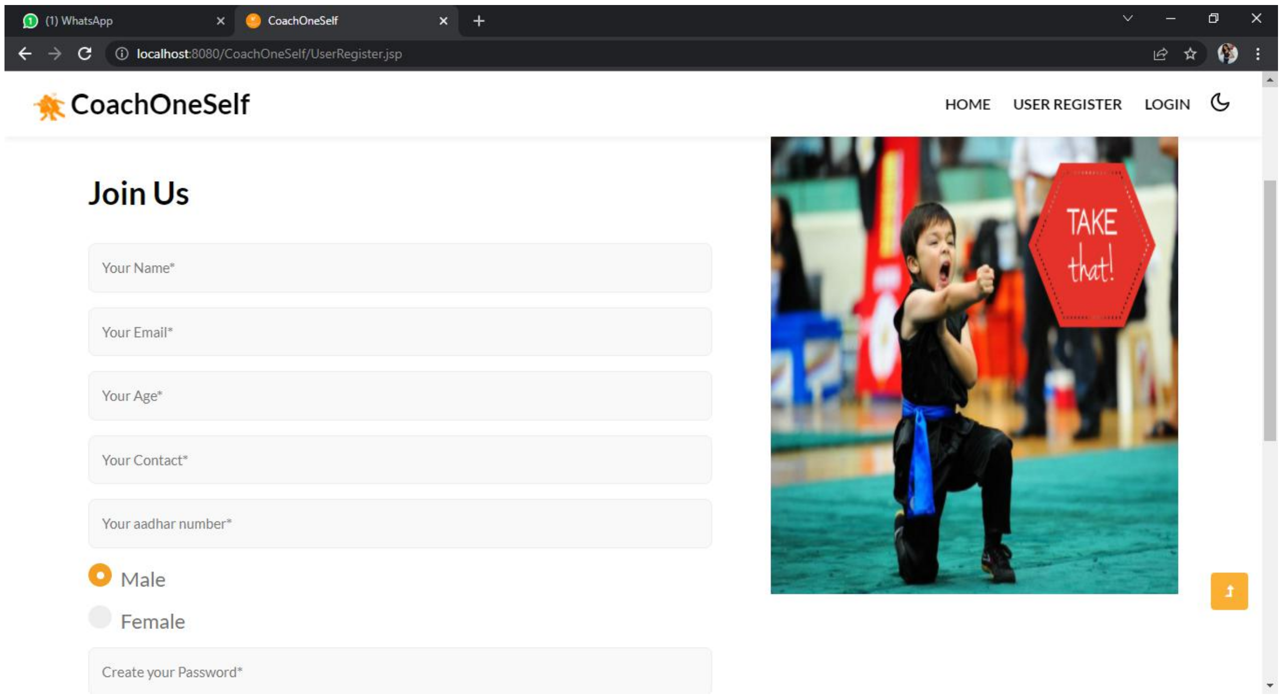
Home Page


(1) WhatsApp
CoachOneSelf - Home
localhost:8080/CoachOneSelf/
CoachOneSelf
HOME
USER REGISTER
LOGIN
Learn to protect Yourself
" Self-defence is a lifestyle - Always prepare! Always Protected! "
START NOW

Login Page



Registration page

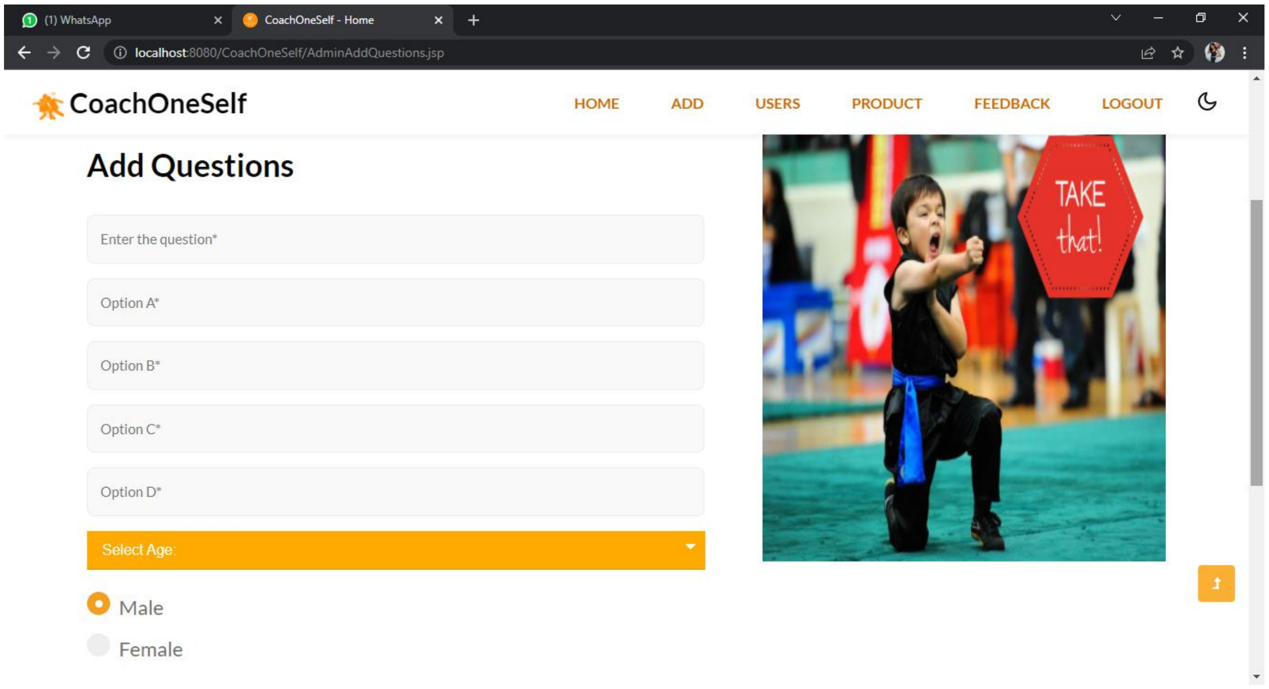

(1) WhatsApp
CoachOneSelf
localhost:8080/CoachOneSelf/UserRegister.jsp

CoachOneSelf
HOME USER REGISTER LOGIN

Join Us

Your Name*
Your Email*
Your Age*
Your Contact*
Your aadhar number*
Male [✓]
Female
Create your Password*

Admin add questionnaire

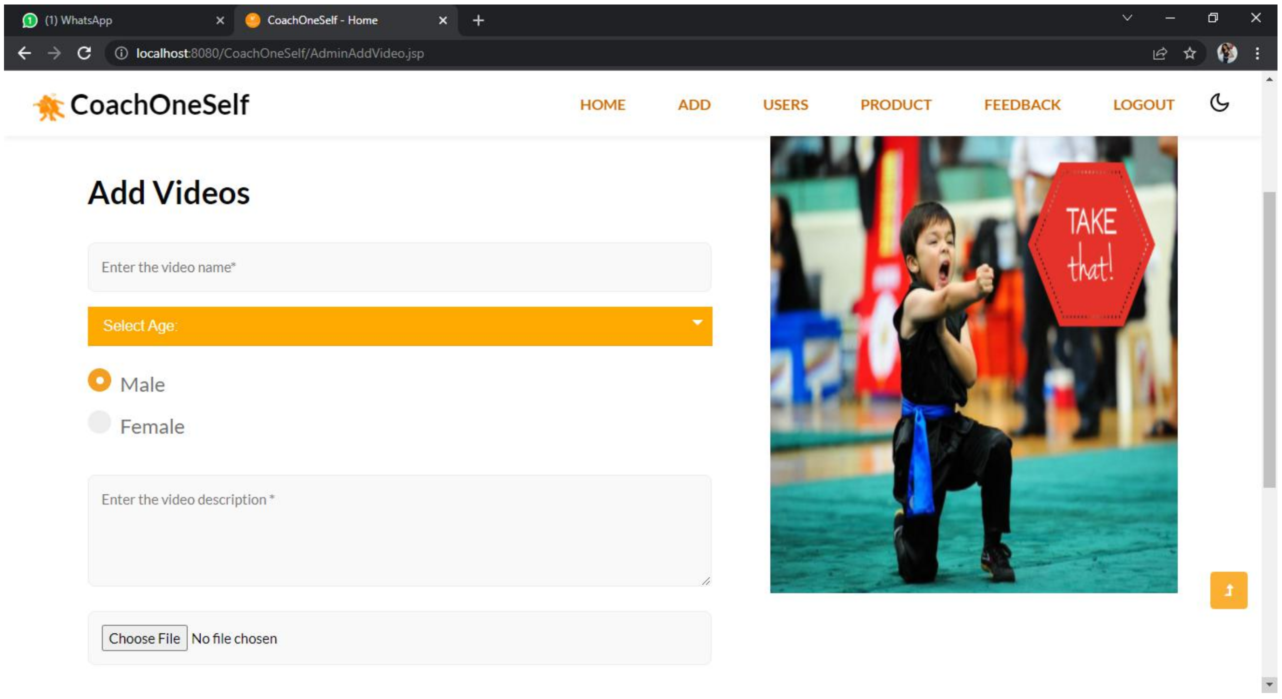

(1) WhatsApp
CoachOneSelf - Home
localhost:8080/CoachOneSelf/AdminAddQuestions.jsp

CoachOneSelf
HOME ADD USERS PRODUCT FEEDBACK LOGOUT

Add Questions

Enter the question*
Option A*
Option B*
Option C*
Option D*
Select Age
Male [✓]
Female

Admin add Videos



(1) WhatsApp | CoachOneSelf - Home

localhost:8080/CoachOneSelf/AdminAddVideo.jsp

CoachOneSelf

HOME ADD USERS PRODUCT FEEDBACK LOGOUT

Add Videos

Enter the video name*

Select Age:

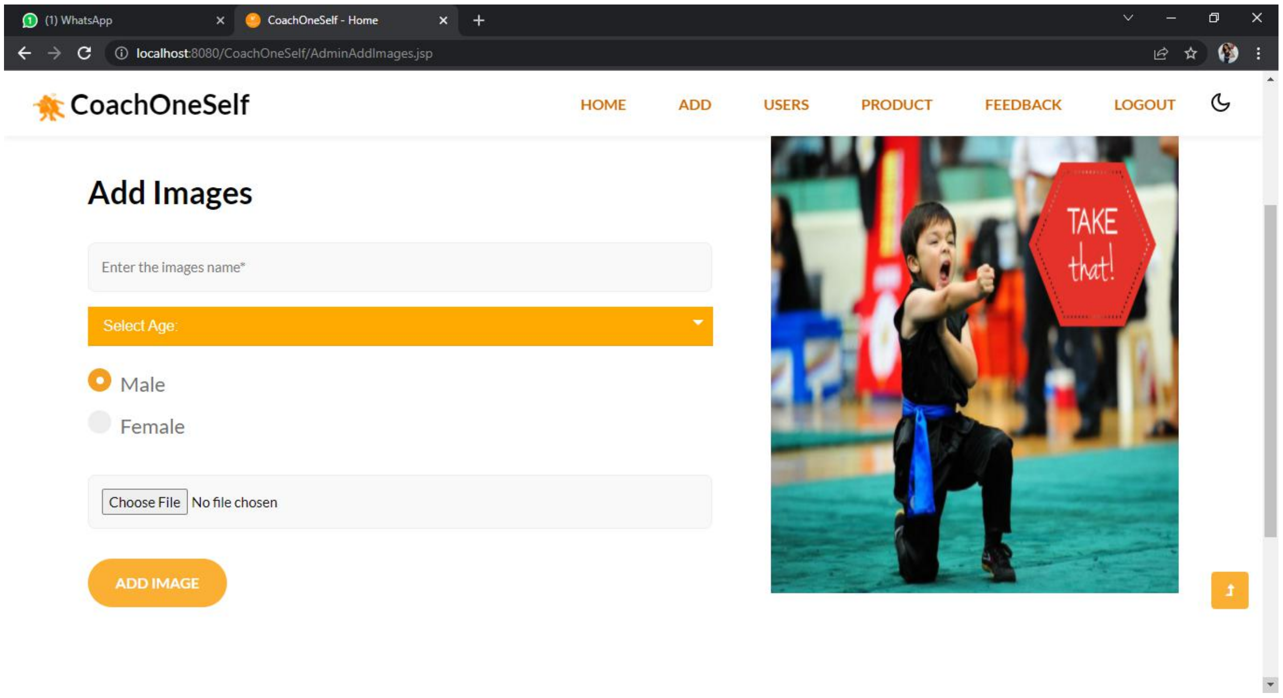
Male

Female

Enter the video description *

Choose File No file chosen

Admin adds images



(1) WhatsApp | CoachOneSelf - Home

localhost:8080/CoachOneSelf/AdminAddImages.jsp

CoachOneSelf

HOME ADD USERS PRODUCT FEEDBACK LOGOUT

Add Images

Enter the images name*

Select Age:

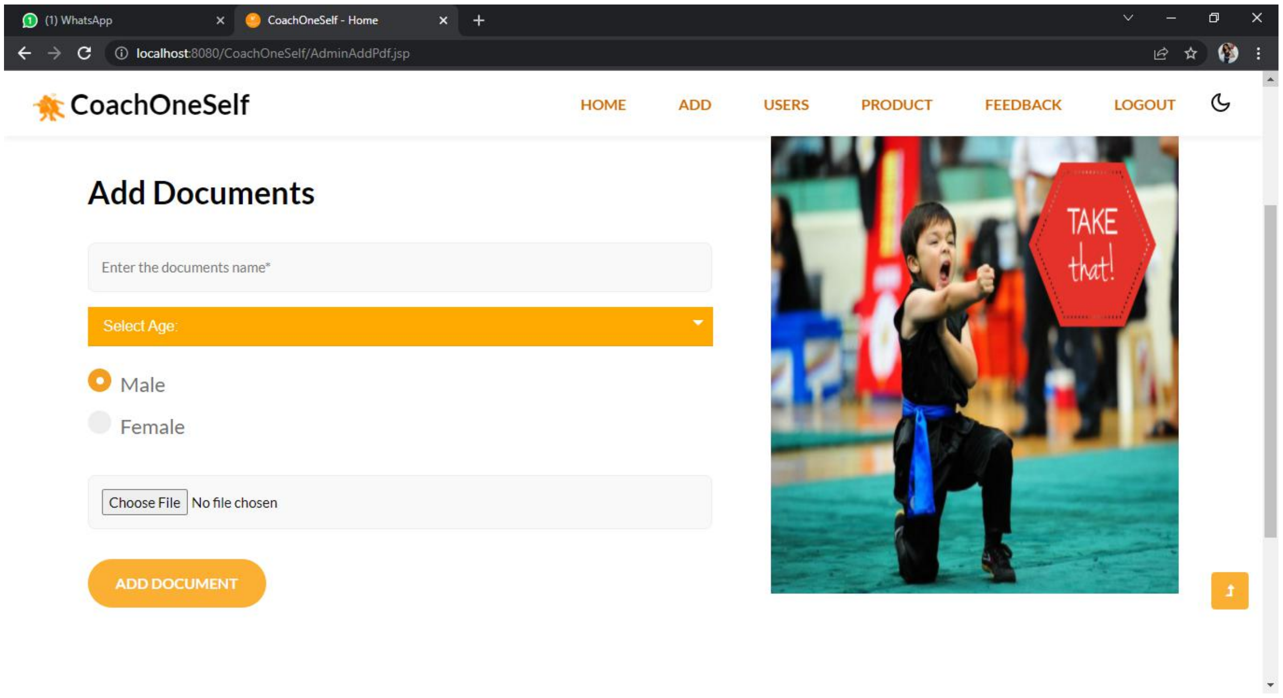
Male

Female

Choose File No file chosen

ADD IMAGE

Admin adds documents



Add Documents

Enter the documents name*

Select Age:

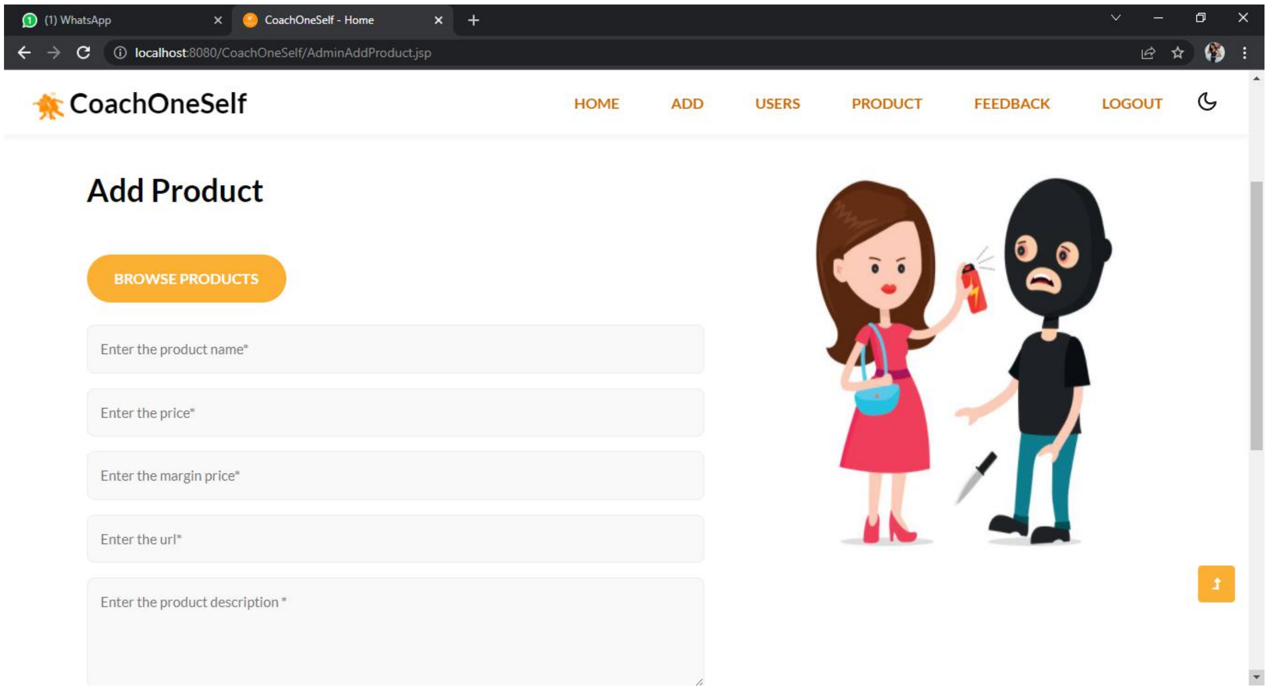
Male

Female

Choose File No file chosen

ADD DOCUMENT

Admin adds Product



Add Product

BROWSE PRODUCTS

Enter the product name*

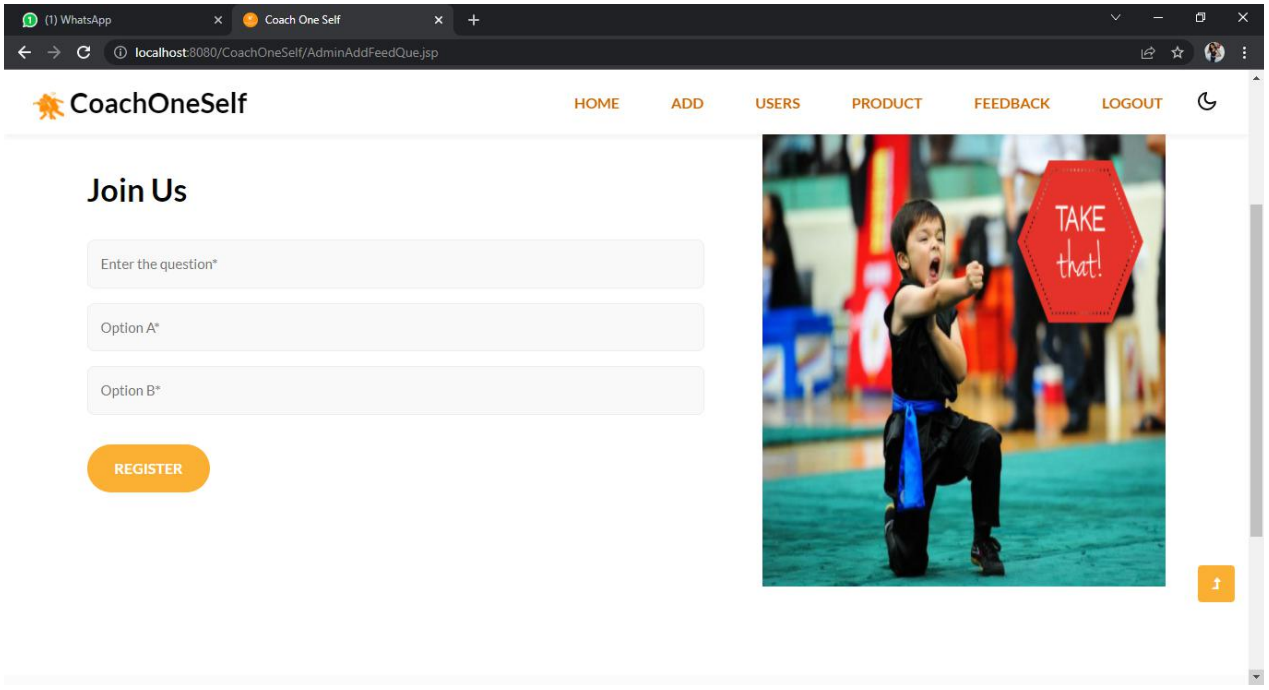
Enter the price*

Enter the margin price*

Enter the url*

Enter the product description *

Admin adds Feedback



Join Us

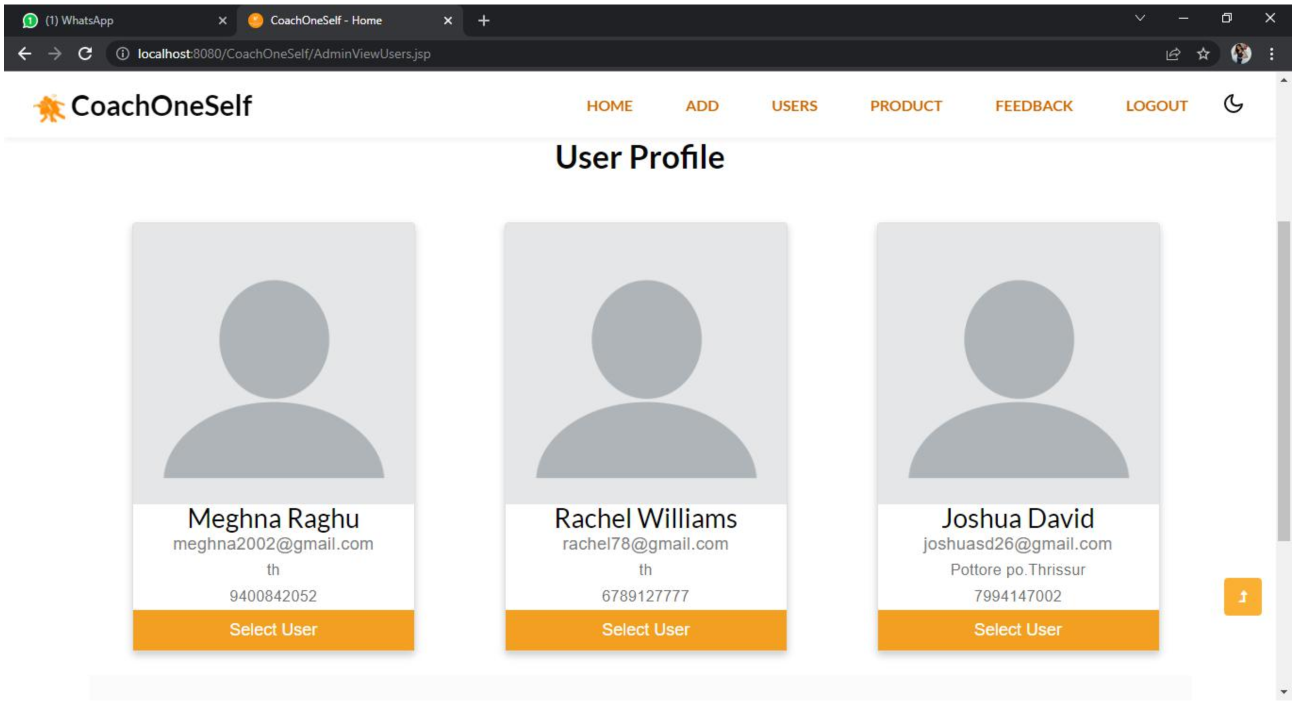
Enter the question*

Option A*

Option B*

REGISTER

Admin view users



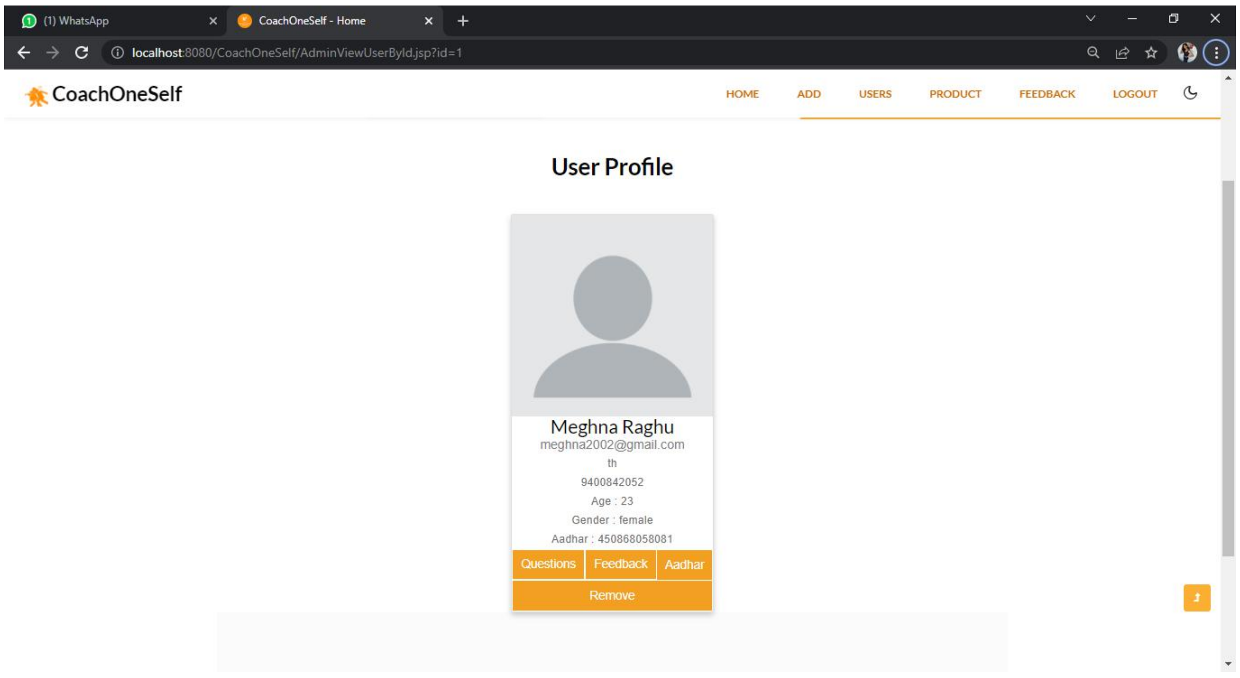
User Profile

Meghna Raghu
meghna2002@gmail.com
th
9400842052
Select User

Rachel Williams
rachel78@gmail.com
th
6789127777
Select User

Joshua David
joshuasd26@gmail.com
Pottore po.Thrissur
7994147002
Select User

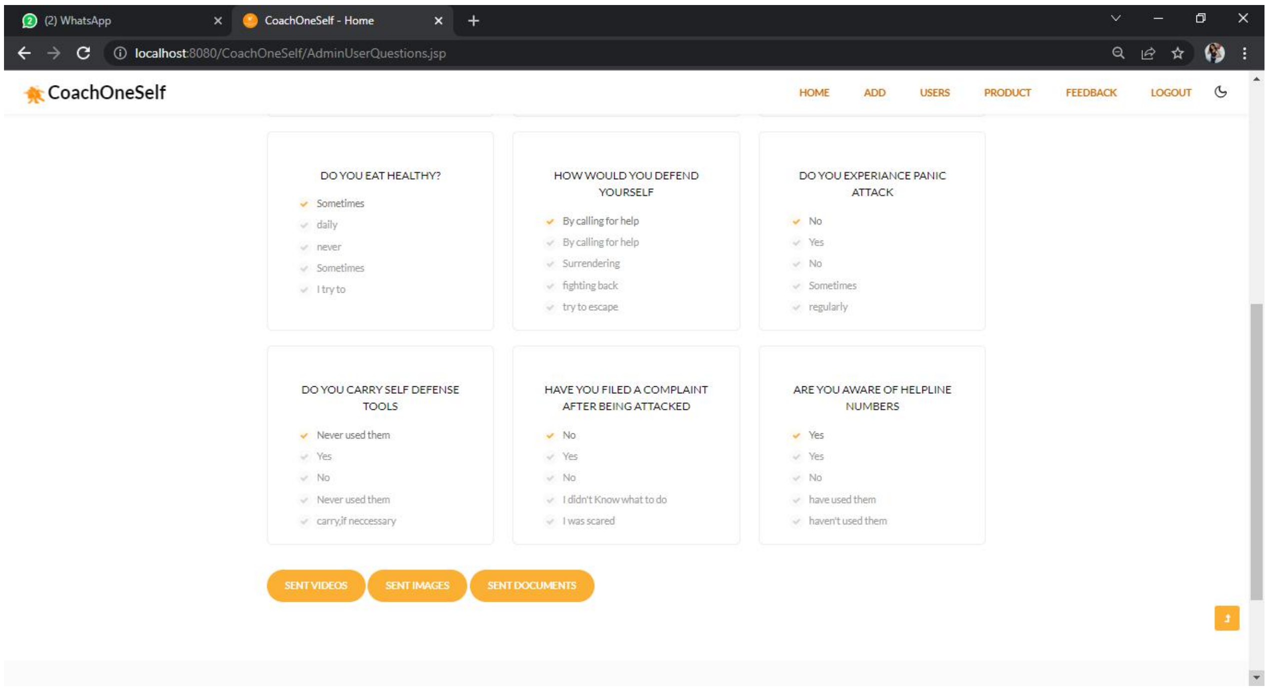
User profile



User Profile

Meghna Raghu
meghna2002@gmail.com
th
9400842052
Age : 23
Gender : female
Aadhar : 450868058081

Admin Uploads trainning session



DO YOU EAT HEALTHY?
- ✓ Sometimes
- daily
- never
- Sometimes
- I try to

HOW WOULD YOU DEFEND YOURSELF
- ✓ By calling for help
- By calling for help
- Surrendering
- fighting back
- try to escape

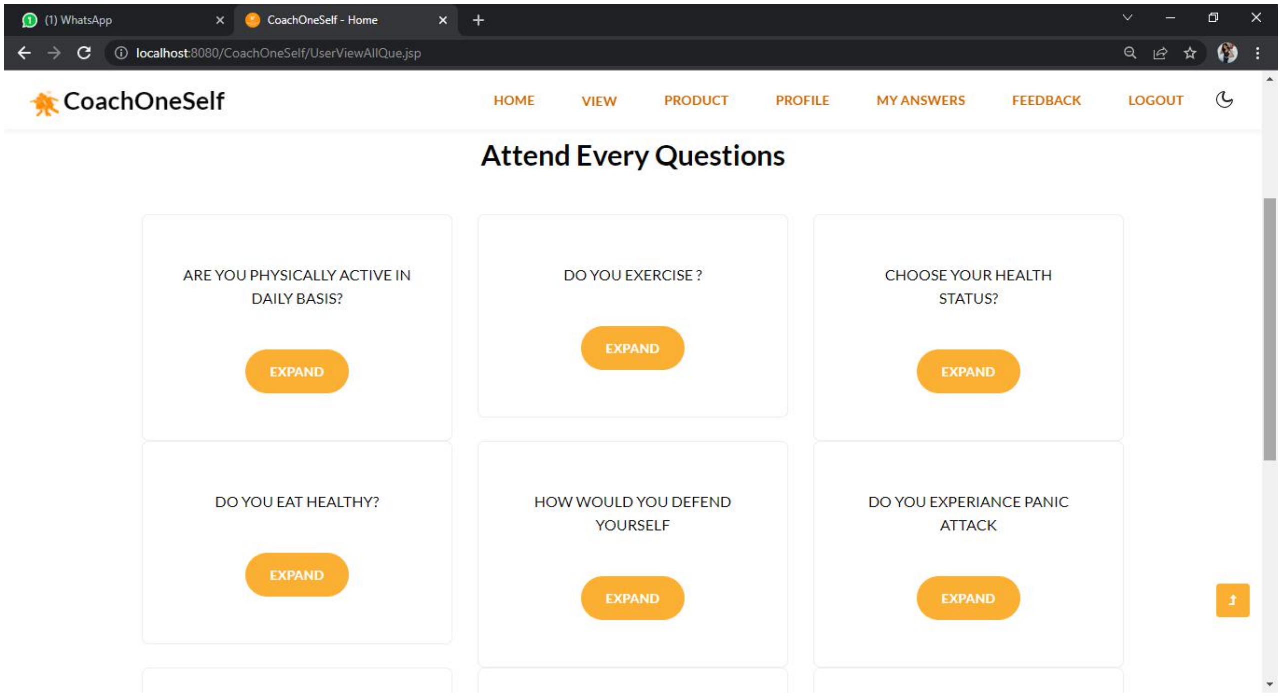
DO YOU EXPERIANCE PANIC ATTACK
- ✓ No
- Yes
- No
- Sometimes
- regularly

DO YOU CARRY SELF DEFENSE TOOLS
- ✓ Never used them
- Yes
- No
- Never used them
- carry,if neccessary

HAVE YOU FILED A COMPLAINT AFTER BEING ATTACKED
- ✓ No
- Yes
- No
- I didn't Know what to do
- I was scared

ARE YOU AWARE OF HELPLINE NUMBERS
- ✓ Yes
- Yes
- No
- have used them
- haven't used them

User fills questionnaire



(1) WhatsApp
CoachOneSelf - Home
localhost:8080/CoachOneSelf/UserViewAllQue.jsp

CoachOneSelf

HOME VIEW PRODUCT PROFILE MY ANSWERS FEEDBACK LOGOUT

Attend Every Questions

ARE YOU PHYSICALLY ACTIVE IN DAILY BASIS?
EXPAND

DO YOU EXERCISE ?
EXPAND

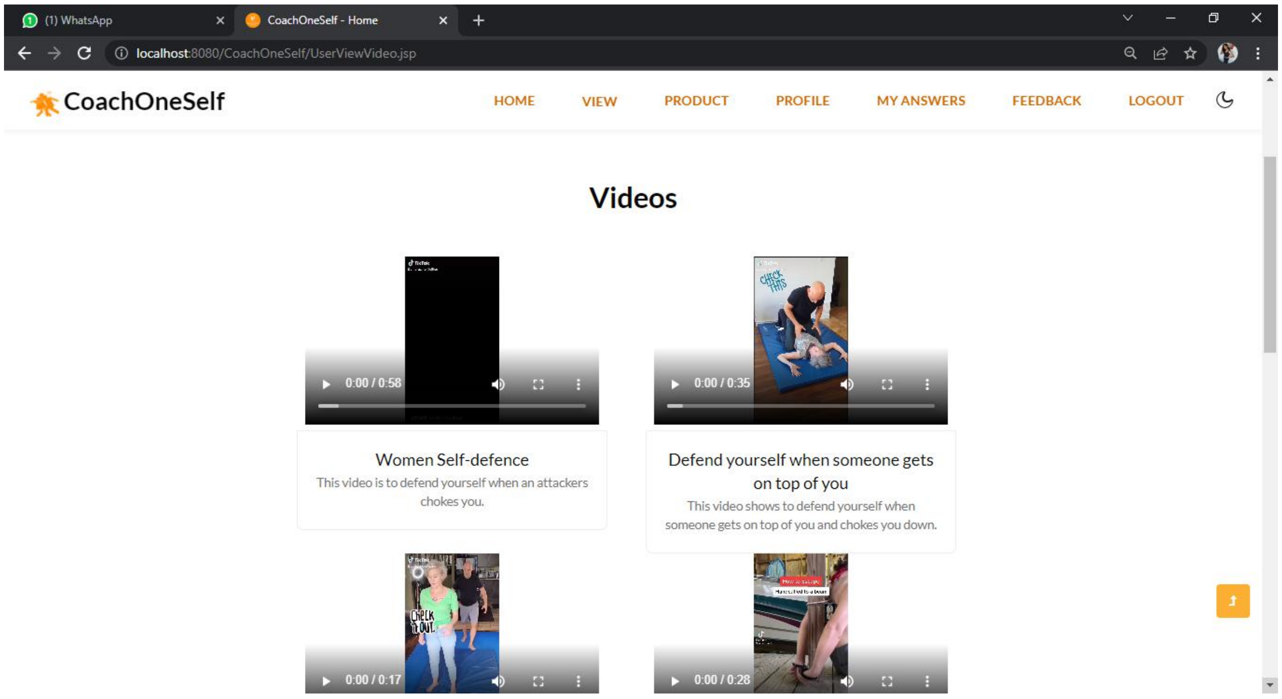
CHOOSE YOUR HEALTH STATUS?
EXPAND

DO YOU EAT HEALTHY?
EXPAND

HOW WOULD YOU DEFEND YOURSELF
EXPAND

DO YOU EXPERIANCE PANIC ATTACK
EXPAND

User View Videos



(1) WhatsApp
CoachOneSelf - Home
localhost:8080/CoachOneSelf/UserViewVideo.jsp

CoachOneSelf

HOME VIEW PRODUCT PROFILE MY ANSWERS FEEDBACK LOGOUT

Videos

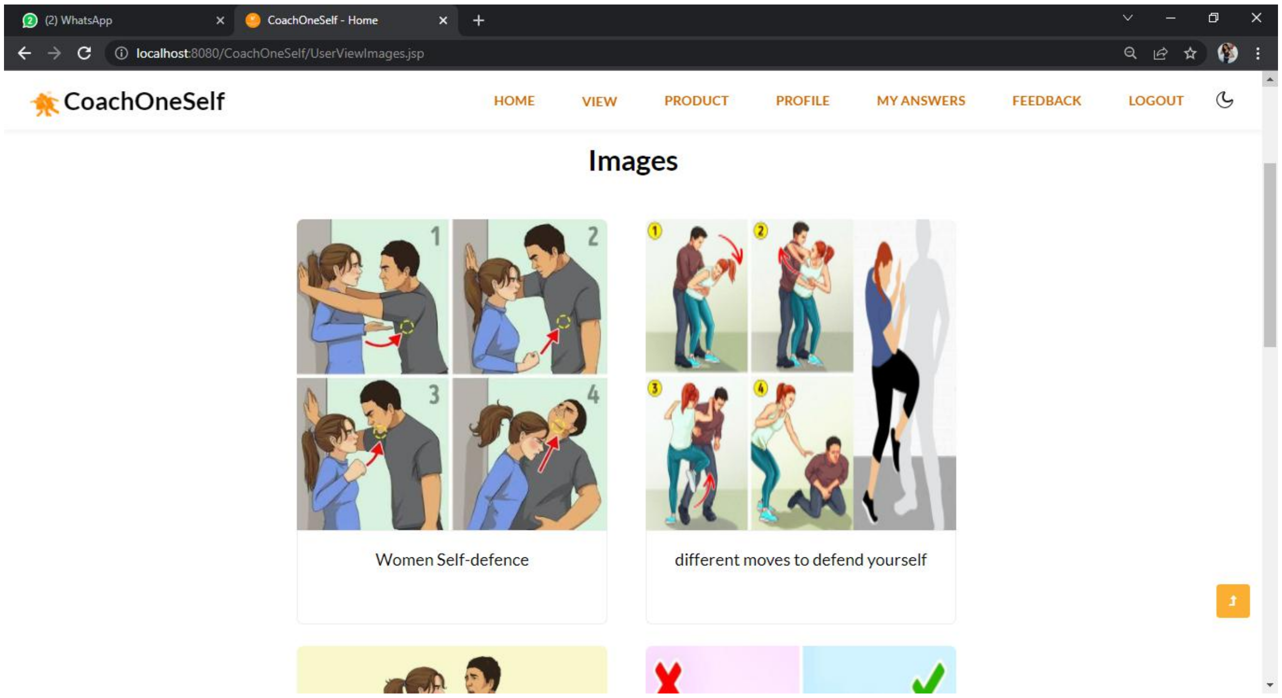
0:00 / 0:58
Women Self-defence
This video is to defend yourself when an attackers chokes you.

0:00 / 0:35
Defend yourself when someone gets on top of you
This video shows to defend yourself when someone gets on top of you and chokes you down.

0:00 / 0:17

0:00 / 0:28

User Views images

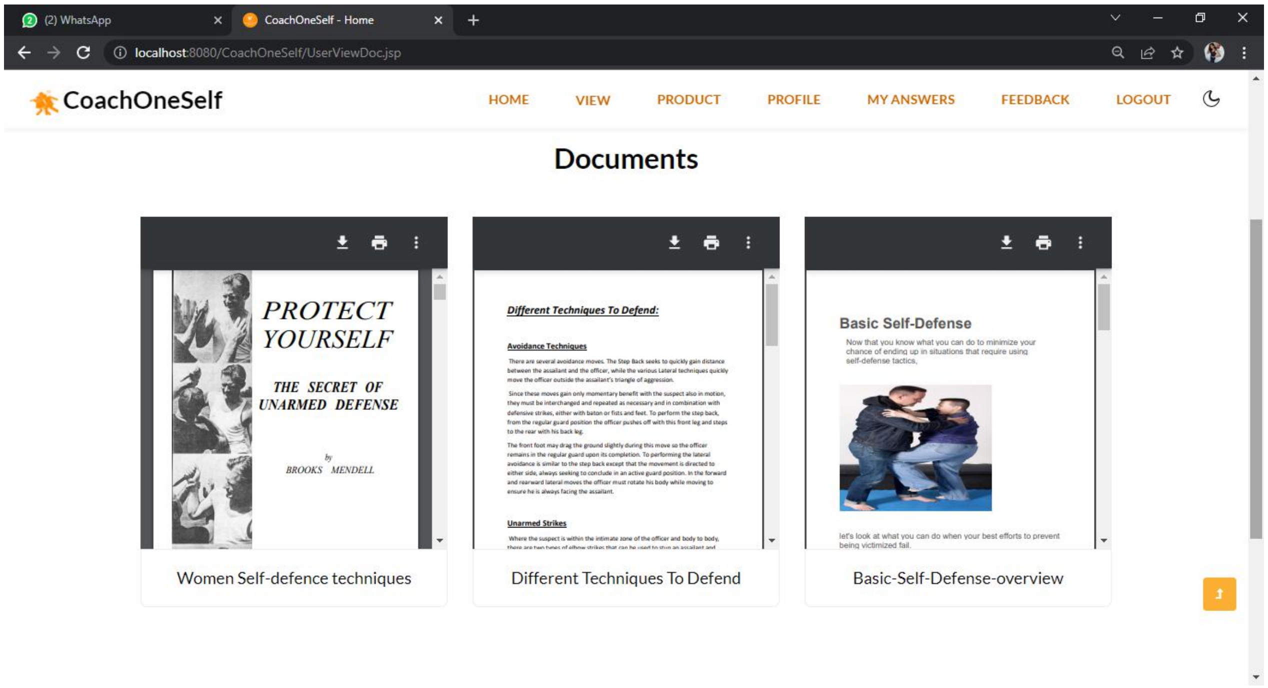


Images

Women Self-defence

different moves to defend yourself

User Views documents



Documents

PROTECT YOURSELF

THE SECRET OF UNARMED DEFENSE

by BROOKS MENDELL

Women Self-defence techniques

Different Techniques To Defend:

Avoidance Techniques

There are several avoidance moves. The Step Back seeks to quickly gain distance between the assailant and the officer, while the various Lateral techniques quickly move the officer outside the assailant's triangle of aggression.

Since these moves gain only momentary benefit with the suspect also in motion, they must be interchanged and repeated as necessary and in combination with defensive strikes, either with baton or fists and feet. To perform the step back, from the regular guard position the officer pushes off with this front leg and steps to the rear with his back leg.

The front foot may drag the ground slightly during this move so the officer remains in the regular guard upon its completion. To performing the lateral avoidance is similar to the step back except that the movement is directed to either side, always seeking to conclude in an active guard position. In the forward and rearward lateral moves the officer must rotate his body while moving to ensure he is always facing the assailant.

Unarmed Strikes

Where the suspect is within the intimate zone of the officer and body to body,

Different Techniques To Defend

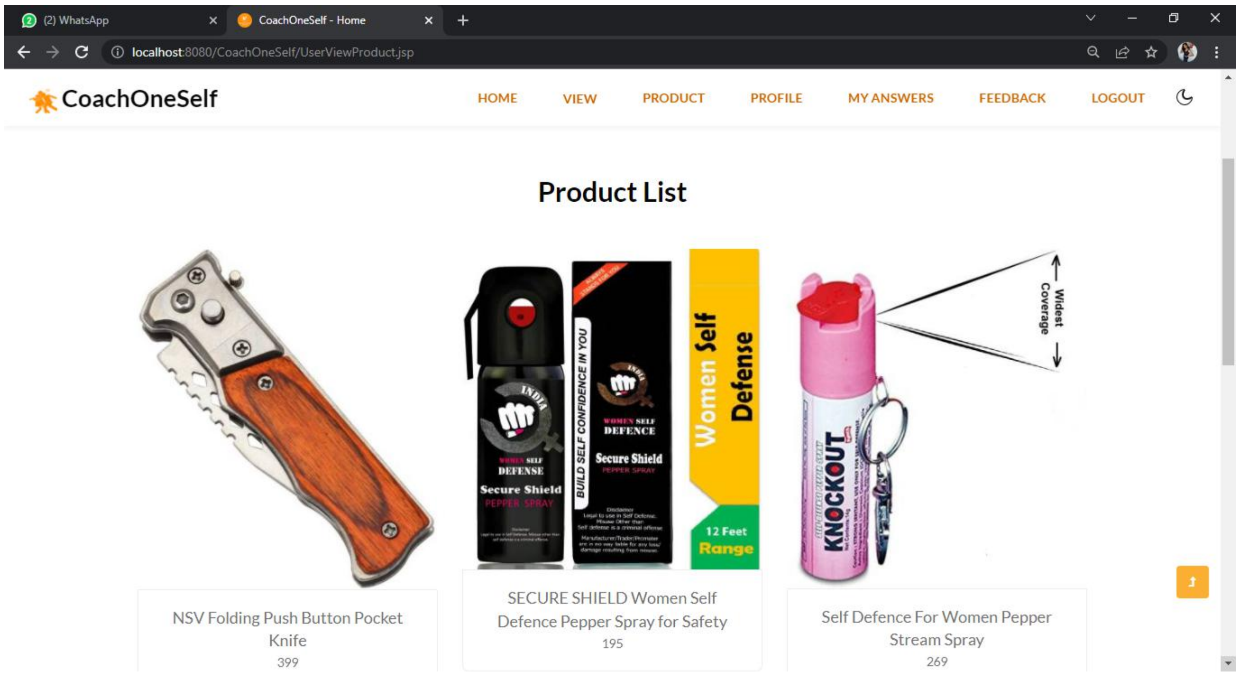
Basic Self-Defense

Now that you know what you can do to minimize your chance of ending up in situations that require using self-defense tactics,

let's look at what you can do when your best efforts to prevent being victimized fail.

Basic-Self-Defense-overview

User view Product list



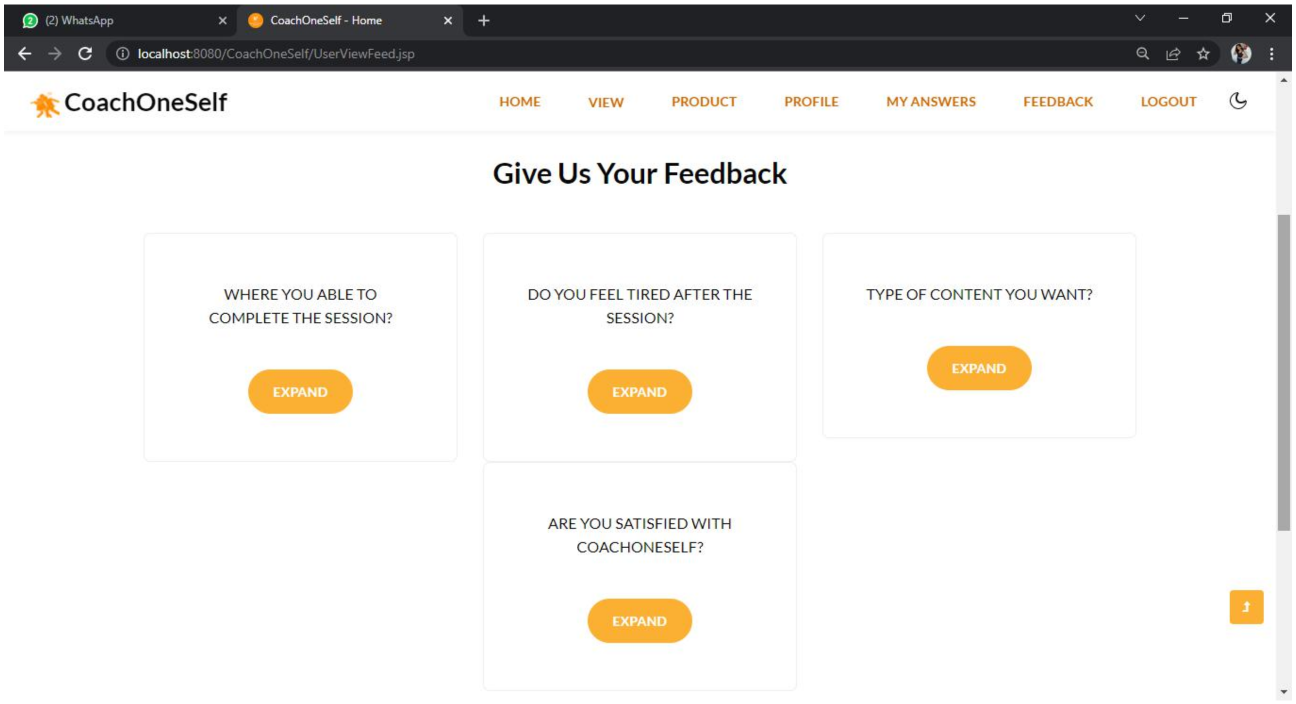
Product List

NSV Folding Push Button Pocket Knife
399

SECURE SHIELD Women Self Defence Pepper Spray for Safety
195

Self Defence For Women Pepper Stream Spray
269

User fills Feedback



Give Us Your Feedback

WHERE YOU ABLE TO COMPLETE THE SESSION?

DO YOU FEEL TIRED AFTER THE SESSION?

TYPE OF CONTENT YOU WANT?

ARE YOU SATISFIED WITH COACHONESELF?