

ELICITING USER PREFERENCES BASED ON ONLINE GAME

Project Report submitted in partial fulfillment of the requirement

for the degree of

Bachelor of Technology

In

Computer Science & Engineering

Under the Supervision of

Mr. Arvind Kumar

By

Reema Mittal (111235)



Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

CERTIFICATE

This is to certify that project report entitled “**ELICITING USER PREFERENCES BASED ON ONLINE GAME**”, submitted by **Reema Mittal** in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Wanknaghat, Solan has been carried out under my supervision. This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date:

Mr. Arvind Kumar

Assistant Professor

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organization. I would like to extend my sincere thanks to all of them. I would like to express my gratitude to all those who gave us the possibility to complete this project. I want to thank the Department of CSE in JUIT for giving me the permission to commence this project in the first instance, to do the necessary research work.

I am deeply indebted to my project guide Mr Arvind Kumar, whose help, stimulating suggestions and encouragement helped me in all the time of research on this project. I feel motivated and encouraged every time I get his encouragement. I am thankful for his aspiring guidance, invaluable constructive and friendly advice during the project work. I am sincerely grateful to him for sharing truthful and illuminating views on a number of issues related to the project. For his coherent guidance throughout the tenure of the project, I feel fortunate to be taught by him, who gave me his unwavering support.

Table of Content

CHAPTER 1	INTRODUCTION.....	1
1.1	Motivation.....	2
1.2	Aim	3
1.3	Advantages of this approach.....	3
1.3.1	Flickr Interestingness.....	3
1.3.2	Voting.....	4
1.3.3	Hot or Not	4
1.4	Applications	5
1.4.1	Global ranking of images.....	5
1.4.2	Collaborative filtering.....	5
1.4.3	Gender prediction.....	5
1.5	Challenges.....	6
CHAPTER 2	LITERATURE REVIEW.....	7
2.1	Early Work in Human Computation.....	7
2.2	Luis Von Ahn.....	8
2.3	Taxonomy of Methods.....	10
2.3.1	Absolute versus Relative Judgments	10
2.3.2	Total versus Partial Judgments	11
2.3.3	Random Access versus Predefined Access	11
2.3.4	“I Like” Versus “Others Like”	12
2.3.5	Direct Versus Indirect.....	13

2.4	Existing Methods	13
2.4.1	Flickr Interestingness.....	13
2.4.2	Voting.....	14
2.4.3	Hot or Not.....	15
CHAPTER 3	PROJECT PLAN AND DESIGN	16
3.1	Mechanism.....	16
3.2	Collecting Gender Biased Images.....	17
3.3	Gender prediction.....	18
3.4	Collaborative filtering.....	18
3.5	Global rankings	20
3.5.1	Empirical Winning Rate (EWR).....	20
3.5.2	ELO Rating.....	20
3.6	Functionalities description for Online Game.....	21
3.6.1	Login/register.....	21
3.6.2	Finding of random player	22
3.6.3	Start the game	22
3.6.4	Select features	22
3.6.5	Game Play.....	23
3.6.6	Maintaining the score	23
3.6.7	Predicting the gender	23
3.6.8	collabrative filtering.....	23
CHAPTER 4	IMPLEMENTATION	25
4.1.	Technologies Used.....	25

4.1.1. Front-End Technologies	25
4.1.2. Back End Technologies	25
4.2 Users of the project	25
4.2.1 Administrator:	26
4.2.2 User:	26
4.2.3 Other people:	26
4.3 Product perspective	26
4.3.1 System interfaces	26
4.3.2 User Interfaces	26
4.4 User characteristics	27
4.5 Constraints	27
4.6 Modules.....	27
4.7 Module 1 : Online game	28
4.7.1 User Login	29
4.7.2 User registration.....	32
4.7.3 Play Web Page	39
4.7.4 Select features	40
4.7.5 Start Game Page.....	41
4.7.6 Game pages	42
4.8 Module 2:Finding the input data set	48
4.8.1 Algorithm to find gender biased images	48
4.9 Module 3: Gender prediction	51
4.9.1 Algorithm for gender prediction	51

4.10 Module 3: Collaborative Filtering	54
4.10.1 Algorithm for collaborative filtering	54
CHAPTER 5 RESULTS AND DISCUSSION.....	56
5.1 Mapping of images	56
5.2 Input for gender biased images algorithm	57
5.3 Output for gender biased images algorithm.....	58
5.4 Input for gender prediction algorithm.....	59
CHAPTER 6 CONCLUSION AND FUTURE SCOPE.....	61
6.1 Conclusions.....	61
6.2 Future scope of the project.....	62
REFERENCES	63

List of Figures

fig.No	Title	Page No.
2.2.1	Luis Von Ahn	8
2.3.5.1	A Taxonomy of Methods	13
3.2.1	Women prefer the image on the left while men prefer the image on the right.	17
3.2.2	For this pair of images it is hard to guess which one is preferred by women and which one by men	8
3.6.1	Flow Chart of the Game	24
4.4.1	Table showing functions of online game	29
4.7.1.1	Flow Chart of the Login functionality	30
4.7.1.2	UI of the Login Page	31
4.7.1.3	UI of the incorrect Login	32
4.7.2.1	Flow chart for User Registration	33
4.7.2.1.1	Flow Chart of the Email Notification	35
4.7.2.1.2	UI for User Registration	36
4.7.2.1.3	UI for unsuccessful User Registration	37
4.7.2.1.4	UI for Successful Registration	38
4.7.3.1	UI for Play Game Page	39

4.7.4.1	UI for select features	40
4.7.5.1	UI for Start Game Page	41
4.7.6.1	UI for first page	42
4.7.6.2	UI for second page	43
4.7.6.3	UI for third page	44
4.7.6.4	UI for fourth page	45
4.7.6.5	UI for fifth page	46
4.7.6.6	UI for sixth page	47
5.1.1	Database view of mapping of images to the specific ids.	56
4.8.2.1	Database view of table gp2	57
5.3.1	Database view of table gp1	58
5.4.1	Database view of table showing the preferences made by the user over the pair of image (A,B)	59
5.5.1	View of the gender prediction of the player according to the preferences	60

ABSTRACT

Eliciting user preferences for large datasets and creating rankings based on these preferences has many practical applications in community based sites. This project gives a new method to elicit user preferences with an online game. In this game we do not ask users what they prefer but what a random person would prefer and rewards them if their prediction is correct. This is done through the implementation of two-player game in which initially a random player is chosen with whom the game will be played and if the random player is not available at that time, the game will be played with the computer (bot) which will appear as if the user is playing game with another human player. Then each player is shown two images and is asked to click on the image their partner would prefer. Here, the player has to think that whether he should pick the one he like best, or the one he think his partner likes best, or the one he think his partner thinks he like best, etc. The game has proven to be enjoyable, has attracted tens of thousands of people and has already collected millions of judgments. Through the judgments collected for the particular user on the basis of game played by the user we can predict the gender of the user. Thus we will show that how merely observing user preferences on a specially chosen set of images can predict a user's gender with high probability. This shows that responding to a request for information, such as which of two images a user prefers, can actually reveal significant information about a person. Under these circumstances, it becomes questionable whether people really can protect their privacy online. This game can collect many types of useful data from the players. It is possible to extract a global beauty ranking within a large collection of images. Also after a person has played the game on a small number of pairs of images , it is possible to extract the person's general image preferences which is known as collaborative filtering. Thus, Computer games offers the wealth of opportunities for research as computer game worlds offer varying degree of sophistication with which we can investigate a particular area of interest.

This project introduces a game that asks two randomly chosen partners which of these two images they think their partner would prefer. If both partners click on the same image, they both obtain points, whereas if they click on different images, neither of them receives points. Though seemingly simple, this game presents players with a strangely recursive conundrum that should they pick the one they like best, or the one they think their partner likes best, or the one they think their partner thinks they like best, etc. On studying the consequences of observing tens of thousands of people play this game on the Internet. The findings suggest that the game can collect many types of useful data from the players. First, it is possible to extract a global "beauty" ranking within a large collection of images. Secondly, it is possible to extract the person's general image preferences. Third, this model can determine a player's gender with high probability. Here, the concentration is on the specific application of images, the wider implication of the findings is that asking partners in a two-player game to guess which of two options their partner will choose represents a viable mechanism for extracting user preferences and data. The game here I concentrated on, called Matchin, has been played by tens of thousands of players for large periods of time. Matchin follows the spirit of —Games with a Purpose ,i.e., games that are fun to play and at the same time collect useful data for tasks that computers cannot yet perform. E.g. The ESP Game is the most well-known game in this category. Whereas the ESP Game collects labels for images, Matchin collects information about how appealing an image is. Similar games have been proposed for tagging music for ranking of music, object detection in image and collection of common-sense facts.

1.1 Motivation

Online gaming is becoming more and more popular and enjoyable among the today's youth.

This popularity of online games can be in the essential way to gather some useful information. These online games offers the wealth of opportunities for research as computer game worlds offer varying degree of sophistication with which we can investigate a particular area of interest. Matchin follows the spirit of —Games with a Purpose, i.e., games that are fun to play and at the same time collect useful data for tasks that computers cannot yet perform. This game concentrates on the specific application of images and collects many types of useful data from the chosen preferences from the player.

First it extracts a global beauty ranking within a large collection of images. Extracting global rankings of large collections of images based on beauty has applications to image search and computer vision. On knowing which images are more appealing could allow for a search engine that displays the more appealing images first. This data could be used to train algorithms that automatically assess the quality of an image. Secondly, after a person has played the game on a small number of pairs of images, it is possible to extract the person's general image preferences. This problem is known as collaborative filtering and is well studied for the case of users giving absolute ratings. Third, we use the players' preferences between images to create a simple gender model. Based on only ten pair-wise judgments, our model can determine a player's gender with high probability. This shows that responding to a request for seemingly benign information, such as which of two images a user prefers, can actually reveal significant information about a person.

Thus online games are becoming beneficial and efficient way to gather the useful information which can be used in different aspects.

1.2 Aim

The aim of this project is to make use of online game to collect useful data from the chosen preferences of the player. The aim is to implement a two player game in which firstly a random player is used against the other player. If random player is not available the user will play against the computer (bot). Then each player is shown two images and asked to click on the image their partner would prefer and rewards them if their prediction is correct and their preferences are collected. Then gender of the players are predicted with high probability on the basis of preferences on a specially chosen set of images. Also the user preferences will help to rank the images

1.3 Advantages of this approach

This method does not involve any ranking or prediction on the basis of meta-information. Neither they are supposed to rank or vote images which often create dilemma and confusion in the minds of people that they might have ranked the previous images wrong. Images are also pre-chosen, so there are very less chances for any kind of malicious activities. Also the judgments are relative, so there are very less chances of producing false results. This approach overcomes the problems which were associated with the following described methods:

1.3.1 Flickr Interestingness

This is the algorithm to rank images. It is at partly based on metadata such as the quantity of user-entered meta-data concerning the media object, the number of users who have assigned metadata to the media object, an access pattern related to the media object, a lapse of time related to the media object, and/or; on the relevance of metadata to the media object.

Flickr's interestingness does not measure beauty directly. Some of the meta-data measures how much other people will possibly like an image. A link to an image, for example, is usually created because the authors think the image might be interesting to their readers. However, the problem with all methods that rely on metadata (like number of comments) is that established long term users who have many friends on that network have an advantage. Ultimately, it is not clear whether interestingness measures the interestingness of the image or the popularity of its author

1.3.2 Voting

The simplest method of eliciting user preferences is just to let users vote on images, using either approval disapproval or a rating scale (e.g., 1 to 5 stars). Users can search for particular items and vote on them (random access). This is possibly the most frequently used method on the Web. These methods, since they are based on random access, share the common characteristic that some items receive many votes while others receive few. This leads to a new problem of combining these ratings into a global ranking. If two items have the same average rating, but one has 1,000 votes while the other one only has 10 votes, the one with more votes should probably be ranked higher.

1.3.3 Hot or Not

It uses a voting system from 1 to 10. It is limited in that it ranks only images of people. The most important difference from the previously mentioned method is that a normal user is given random images to rate; they cannot search for them. However, it is still possible for people to send a link to an image to a friend who can then rate the picture. Therefore, it is still easy for malicious users to cheat and rate their friends' pictures higher than they might rank otherwise.

1.4 Applications

This game can collect many types of useful data from the players which can be used in various applications:

1.4.1 Global ranking of images

It is possible to extract a global beauty ranking within a large collection of images. A good global ranking can be extracted like this is the same problem as inferring the skill of chess players by just looking at their wins and losses. Extracting global rankings of large collections of images based on beauty has applications to image search and computer vision. In the case of image search, knowing which images are more appealing could allow for a search engine that displays the more appealing images first. In the case of computer vision, this data could be used to train algorithms that automatically assess the quality of an image (e.g., a camera that tells the user how good a picture is).

1.4.2 Collaborative filtering

After a person has played the game on a small number of pairs of images, it is possible to extract the person's general image preferences which would help for presenting future data on the basis of these preferences.

1.4.3 Gender prediction

The players' preferences between images can be used to create a simple gender model. Based on only ten pair-wise judgments, this model can determine a player's gender with high probability. This helps in various online sites to show users the data suitable and preferable according to their gender.

1.5 Challenges

- In order to predict the gender of a user we have to select proper gender bias images for which firstly the game has to be played by thousands of users knowing their gender and then collecting their results on which then gender biasing will be done in order to collect the proper gender biased images pair which is a time consuming process.
- In to obtain ranking of images, firstly we have to select set of thousands of 100 images which is difficult as new images keeps on adding every day, so we can't have any specific set.

2.1 Early Work in Human Computation

Human-based computation (apart from the historical meaning of "computer") research has its origins in the early work on interactive evolutionary computation. The idea behind interactive evolutionary algorithms is due to Richard Dawkins. In the Biomorphs software accompanying his book *The Blind Watchmaker* (Dawkins, 1986) the preference of a human experimenter is used to guide the evolution of two-dimensional sets of line segments. In essence, this program asks a human to be the fitness function of an evolutionary algorithm, so that the algorithm can use human visual perception and aesthetic judgment to do something that a normal evolutionary algorithm cannot do. However, it is difficult to get enough evaluations from a single human if we want to evolve more complex shapes. Victor Johnston and Karl Sims extended this concept by harnessing the power of many people for fitness evaluation (Caldwell and Johnston, 1991; Sims, 1991). As a result, their programs could evolve beautiful faces and pieces of art appealing to public. These programs effectively reversed the common interaction between computers and humans. In these programs, the computer is no longer an agent of its user, but instead, a coordinator aggregating efforts of many human evaluators. These and other similar research efforts became the topic of research in aesthetic selection or interactive evolutionary computation (Takagi, 2001), however the scope of this research was limited to outsourcing evaluation and, as a result, it was not fully exploring the full potential of the outsourcing.[3]

A concept of the automatic Turing test pioneered by Moni Naor (1996) is another precursor of human-based computation. In Naor's test, the machine can control the access of humans and computers to a service by challenging them with a natural language processing (NLP) or computer vision (CV) problem to identify humans among them. The set of problems is chosen in a way that they have no algorithmic solution that is both

effective and efficient at the moment. If it existed, such an algorithm could be easily performed by a computer, thus defeating the test. In fact, Moni Naor was modest by calling this an automated Turing test. The Imitation Game described by Alan Turing (1950) didn't propose using CV problems. It was only proposing a specific NLP task, while the Naor test identifies and explores a large class of problems, not necessarily from the domain of NLP that could be used for the same purpose in both automated and non-automated versions of the test.[3]

Finally, Human-based genetic algorithm (HBGA) encourages human participation in multiple different roles. Humans are not limited to the role of evaluator or some other predefined role, but can choose to perform a more diverse set of tasks. In particular, they can contribute their innovative solutions into the evolutionary process, make incremental changes to existing solutions, and perform intelligent recombination. In short, HBGA allows humans to participate in all operations of a typical genetic algorithm. As a result of this, HBGA can process solutions for which there are no computational innovation operators available, for example, natural languages. Thus, HBGA obviated the need for a fixed representational scheme that was a limiting factor of both standard and interactive EC. These algorithms can also be viewed as novel forms of social organization coordinated by a computer[3]

2.2 Luis Von Ahn



Fig. 2.2.1 Luis Von Ahn

Luis von Ahn (born 1979) is a Guatemalan entrepreneur and an associate professor in the Computer Science Department at Carnegie Mellon University. He is known as one of the pioneers of crowd sourcing. He is the founder of the company reCAPTCHA, which was sold to Google in 2009, and the co-founder and CEO of Duolingo, a popular language-learning platform.

As a professor, his research includes CAPTCHAs and human computation, which has earned him international recognition and numerous honours.

Von Ahn's early research was in the field of cryptography. With Nicholas J. Hopper and John Langford, he was the first to provide rigorous definitions of steganography and to prove that private-key steganography is possible.

In 2000, he did early pioneering work with Manuel Blum on CAPTCHAs, computer-generated tests that humans are routinely able to pass but that computers have not yet mastered. These devices are used by web sites to prevent automated programs, or bots, from perpetrating large-scale abuse, such as automatically registering for large numbers of accounts or purchasing huge number of tickets for resale by scalpers. Von Ahn's Ph.D. thesis, completed in 2005, was the first publication to use the term "human computation" that he had coined for methods that combine human brainpower with computers to solve problems that neither could solve alone. Von Ahn's Ph.D. thesis is also the first work on Games With A Purpose, or GWAPs, which are games played by humans that produce useful computation as a side-effect. The most famous example is the ESP Game, an online game in which two randomly paired people are simultaneously shown the same picture, with no way to communicate. Each then lists a number of words or phrases that describe the picture within a time limit, and are rewarded with points for a match. This match turns out to be an accurate description of the picture, and can be successfully used in a database for more accurate image search technology. The ESP Game was licensed by Google in the form of the Google Image Labeller, and is used to improve the accuracy of the Google Image Search. Von Ahn's games brought him further coverage in the mainstream media. His thesis won the Best Doctoral Dissertation

In 2007, von Ahn invented reCAPTCHA, a new form of CAPTCHA that also helps digitize books. In reCAPTCHA, the images of words displayed to the user come directly from old books that are being digitized; they are words that optical character recognition could not identify and are sent to people throughout the web to be identified. ReCAPTCHA is currently in use by over 100,000 websites and is transcribing over 40 million words per day.

As of 2014, von Ahn is working on Duolingo, a company that aims to coordinate millions of people to translate the Web into every major language.[7]

2.3 Taxonomy of Methods

2.3.1 Absolute versus Relative Judgments

First, we make a distinction between absolute and relative judgments. An absolute judgment is a judgment that assigns an absolute score to an item, such as a star rating from 1-5 where 1 is worst and 5 is best. On the other hand, relative judgment only compares items, i.e. this image is better than that image.

Absolute ratings have two problems: calibration (or better, the lack of calibration) and limited resolution. Calibration is the problem of defining what a particular rating means compared with previous ratings and compared with other people's ratings. For example, if I usually assign 1 or 2, my 4 might have the same meaning as someone else's 5. Also, judgment may change during the rating process: for example, a user might in their first rating give a 5 to a good image, only to discover later that there are far better images. Thus, they may want to change their first rating to a 4. [1]

This creates systematic errors in the data. Limited resolution is the problem of assigning a rating to an image that is only marginally better than a different image. Assuming that the rating system only has 5 levels, the user might give it the same rating even though they clearly think it is better (but not good enough for the next level). In this case we lose

information. To overcome the problem of limited resolution, one could simply use a rating system with finer granularity, say from 1 to 100. However, many judges will not adapt to this system, but instead keep a scale of 1 to 10 in their mind and map 8 to 80 and so forth. Relative judgments have the advantage that they are usually easy to make. In most cases, they do not change after seeing new information, i.e., a user who prefers image A over image B will still do so after they have seen other images. Even if the absolute ratings of image A and image B change over time, their relative ordering is likely to stay the same. Therefore, old absolute ratings are more likely to be inaccurate than old relative ratings.[1]

2.3.2 Total versus Partial Judgments

By total judgments, we mean that the judges are required to make judgments about all of the images. In the case of absolute ratings, the user is required to rate all n images. In the case of relative ratings, the user is required to compare every image with every other image, which is on the order of $O(n^2)$ comparisons. Total judgments are, therefore, infeasible for large datasets. Partial judgments, however, have the problem of how to deal with incomplete data.[1]

2.3.3 Random Access versus Predefined Access

By random access, we mean that the judges are allowed to search for particular items and then rate them. This has the advantage that the judges can focus on rating things in which they are most interested. However, it has a major drawback: it opens the door to malicious manipulation. Judges could easily search for their own pictures and always give them the highest ratings. This behaviour cannot easily be stopped on the Internet since the cost of creating new fake identities is extremely low and it is not (currently) possible to tell fake accounts from real accounts. Another drawback of random access

methods is that some items receive many ratings while others receive few. In such cases, combining the ratings becomes difficult.

By predefined access, we mean methods where the users are given images to rate in a predefined sequence. Thus, the users cannot influence which images they can rate. While theoretically it is still possible to cheat just by waiting for one's own images, it is much harder. In a method employing random access, the chance of being able to rate one's own images is 1. For a method that randomly shows one out of n images (with replacement) to rate, the chance of being able to rate one's own image is $1/n$, and the expected time to wait until one sees their own image is n . Therefore, methods that use predefined access have the desirable property that the possibility of cheating decreases as the amount of data increases.[1]

2.3.4 “I Like” Versus “Others Like”

Another important distinction between methods is whether the judges are asked —what do you like? Or what do you think others will like? Although the difference looks subtle at first, it has major implications. We can compare this to the problem of predicting elections. The most common way is to poll potential voters and ask them who they would vote for in the upcoming election. One then takes the sample average as an estimate of the future election result. This is the ‘I like’ case. The other option is to ask them, who do they think is going to win the election? In this case they will not only consider their own opinion but also the opinion of their friends and relatives in combination with external information (polls, news, etc.). This is the ‘others like’ case.

In the ‘I like’ case, every voter automatically becomes a weak predictor, because every voter only has a limited amount of information at his/her disposal. In this ‘others like’ case we can make a further distinction between methods that ask for what one particular partner might prefer and what other people in general prefer.[1]

2.3.5 Direct Versus Indirect

By direct, we mean methods that ask the judges about the ‘beauty’ of an image. Indirect methods would infer beauty through meta-information. Examples of meta-information are number of views, number of comments, number of tags, and number of pages linking to a particular image. Indirect methods have the disadvantage that, once the methods are known, their ratings can quickly be subjected to cheating. People could easily create many comments on their own images, add lots of tags, create dummy pages linking to their images, etc. This means that even though an indirect method might use predefined access (for example by crawling images and counting incoming links, etc.), people still have random access to the meta-information and can change it in any way they want[1].

	Flickr	Voting	Hot/Not	Matchin
Partial	Yes	Yes	Yes	Yes
Direct	No	Yes	Yes	Yes
Predefined access	No	No	Partly	Yes
Others like	Partly	No	No	Yes
Relative	No	No	No	Yes

Fig. 2.3.5.1 A Taxonomy of Methods

2.4 Existing Methods

2.4.1 Flickr Interestingness

The popular online photo sharing Web site Flickr has developed its own algorithm to rank images. Although their algorithm has not been published, we know from their patent

application that it is at least partly based on meta-data such as the quantity of user-entered meta-data concerning the media object, the number of users who have assigned metadata to the media object, an access pattern related to the media object, a lapse of time related to the media object, and/or; on the relevance of metadata to the media object. (We note that all these meta-data can easily be faked.)

This means that Flickr's 'interestingness' does not measure 'beauty' directly. Some of the meta-data measures how much other people will possibly like an image. A link to an image, for example, is usually created because the authors think the image might be interesting to their readers. However, the problem with all methods that rely on meta-data (like number of comments) is that established long-term users who have many friends on that network have an advantage. Ultimately, it is not clear whether 'interestingness' measures the interestingness of the image or the popularity of its author.[1]

2.4.2 Voting

Perhaps the simplest method of eliciting user preferences is just to let users vote on images, using approval/ disapproval or a rating scale (e.g., 1 to 5 stars). Users can search for particular items and vote on them (random access). This is possibly the most frequently used method on the Web: Digg, YouTube and others all use variants of this scheme to rate and rank their content. These methods, since they are based on random access, share the common characteristic that some items receive many votes while others receive few. This leads to a new problem of combining these ratings into a global ranking. If two items have the same average rating, but one has 1,000 votes while the other one only has 10 votes, the one with more votes should probably be ranked higher. However, generalizing this principle is non-trivial.[1]

2.4.3 Hot or Not

The Internet site 'Hot or Not' uses a voting system from 1 to 10. It is limited in that it ranks only images of people. The most important difference from the previously mentioned sites is that a normal user is given random images to rate; they cannot search for them. However, it is still possible for people to send a link to an image to a friend who can then rate the picture. Therefore, it is still easy for malicious users to cheat and rate their friends' pictures higher than they might rank otherwise.[1]

CHAPTER 3 PROJECT PLAN AND DESIGN

3.1 Mechanism

Matchin is a two-player game that is played over the Internet. At the beginning of the game, a player is matched randomly with another person who is visiting the game's Web site at the same time. If there is no other player available at the same time, we pair them with a bot (a computer that plays as if it was a human). After the player is matched with its partner (either human or machine), they play several rounds. In each round, the two players see the same two images and both are asked to click on the image that they think their partner likes better. If they agree, they both receive points. Thus, if the players want to score many points, they not only have to consider which image they prefer, but also which image their partner might prefer.

The players could get many points by quickly picking the images at random (e.g., always choosing the one on the left). This will allow players to get 100 points with 50% probability in every round even without taking the time to look at every image and thus made the game less enjoyable. To make the game more fun, the players are given more points for consecutive agreements. More specifically, Matchin uses a sigmoid function for scoring games. While the first match only earns a few points, the second and third match in a row earn exponentially more points. All clicks are recorded and stored in a database. The bot uses these stored clicks to emulate a human as closely as possible. When it sees two images, it clicks on the image that was considered to be better by a human in an earlier game. The bot mimics the same person for the entire game. Also, the bot waits exactly as long as the human did. The bot's clicks are not recorded.

3.2 Collecting Gender Biased Images

For the gender prediction, we have to find a pair of images (A, B) that has a strong gender bias. The strong gender bias images are selected on the basis of parameter conditional entropy $H[G|X]$ of the player's gender G given that we know the player's decision X , where $A > B$ means that image A was considered better than image B :

$$H[G|X] = \Pr_x H[G|X = x] \quad x \in A < B, A > B$$

A pair (A, B) has a large gender bias (and is therefore good for determining the gender of a new player) if the conditional entropy $H[G|X]$ is small, i.e., learning the decision tells us a lot about the gender. The necessary conditional probabilities $\Pr(G = g|X = x)$ can be computed with Bayes' rule given the class conditionals $\Pr(X = x|G = g)$. For the class conditionals, we trained two ELO predictors, one with male players only and one with female players only. We then compute $H[G|X]$ for many pairs of images and select pairs for which $H[G|X]$ is smaller than a fixed threshold value.[2]



Fig 3.2.1 Women prefer the image on the left while men prefer the image on the right.



Fig 3.2.2 For this pair of images it is hard to guess which one is preferred by women and which one by men

3.3 Gender prediction

Once we know their judgments on the 10 pairs, we use a simple naïve Bayes classifier to predict their gender. The naïve Bayes classifier assumes that the individual decisions are independent given the gender and chooses the label g that maximizes the likelihood of the data:[2]

$$\hat{g} = \operatorname{argmax}_g \Pr(G = g) \prod_{i=1}^{10} \Pr(X_i = x_i | G = g)$$

3.4 Collaborative filtering

In the previous methods, we treated all users equally. In the collaborative filtering setting, we want to find out not only about general preferences, but also about each individual's preferences. This allows us to recommend images to each user based on his/her preferences. Also, we can compare users and images with each other (which users are similar and which images are similar?). Therefore, we have new collaborative filtering algorithm Relative SVD that uses only comparative judgments as its input. Relative SVD is based on collaborative filtering with absolute ratings. We adapt their model to work in

a setting where we only have information about relative ratings. In this model, each user i and each image j is described by a feature vector of length K . We store the user feature vectors in a $n(\text{users}) \times K$ matrix \mathbf{U} , where each row is a user's feature vector. Similarly, we store the image feature vectors in a $n(\text{images}) \times K$ matrix \mathbf{V} . We say that the amount by which user i likes image j is equal to the dot product of their feature vectors:

$$\text{Likes}(u_i, v_j) = u_i^T v_j$$

We interpret the data gathered from our game as a set D of triplets (i, j, k) where i is a user and j is the image that was preferred over the image k in a comparison. We set d_{ijk} equal to 1 for each element in the training set:

$$\forall i, j, k \in D: d_{ijk} = 1$$

The error for a particular decision e_{ijk} between a sample from the training data and our model can then be written as

$$e_{ijk} = d_{ijk} - (\text{likes}(u_i, v_j) - \text{likes}(u_i, v_k))$$

And the total sum of squared errors (SSE) as:

$$\text{SSE} = \sum_{(i,j,k) \in D} e_{ijk}^2$$

The goal is to find the feature matrices that minimize the total sum of squared errors:

$$(U,V) = \operatorname{argmin}_{(u,v)} SSE$$

In order to minimize this error, we compute the partial derivatives for each user feature vector and for each image feature vector.

3.5 Global rankings

There are several methods to combine the relative judgments into a global ranking.

3.5.1 Empirical Winning Rate (EWR)

The simplest form of a ranking function is to use the empirical winning rate as an estimate for an image's quality. The empirical winning rate is the number of times an image was preferred over a different image, divided by the total number of comparisons in which it was included. The empirical winning rate is easy to understand, but has two problems. For images that have a low degree (because they have taken part in few comparisons), the empirical winning rate might be artificially high or low. The second problem is that it does not take the quality of the competing image into account, i.e., winning against a bad image is worth the same as winning against a good image.[2]

3.5.2 ELO Rating

The ELO rating tries to overcome the problem. In this, we first initialize each image's ELO rating R_i to 1,600. Before each comparison between two images i and j we compute their expected scores (i.e., their expected chance of winning in this comparison) E_i and E_j according to a scaled logistic function:

$$E_i = 1/(1+10^{(R_i-R_j/400)}) \quad E_j = 1/(1+10^{R_j-R_i/400})$$

The factor 400 is chosen such that a player whose ELO score is 200 higher than another player's ELO score has a chance of winning of about 75%. After the comparison, we know that either image i or image j won, i.e., we know the true score S_i :

$S_i = 1$ if image i won, and

$S_i = 0$ if image i lost.

The prediction error is $S_i - E_i$. We then update the image's ELO rating R_i accordingly:

$$R_i \leftarrow R_i + K(S_i - E_i) \quad [2]$$

Thus, if the *expected score* of image i is above its *true score* the image's ELO rating will be adjusted downward, otherwise it will be adjusted upward. K is a model parameter that defines by how much the scores of the two images are changed. A large value of K makes the scores more sensitive to winning or losing a single comparison. To compute the ELO ratings, we iterate over all comparisons in our training set and update the R_i 's accordingly. We then use the image's ELO ratings as our ranking function: $f_{ELO} i = E_i$.

3.6 Functionalities description for Online Game

3.6.1 Login/register

If the user wants to play this game, he has to first login for the game.

- If the user is an existing user, his account is already present or created, he has to enter the details-username and password. Then the authentication is performed on these

details. If client is authenticated, he goes to the next else they remain on the same page until the correct details are entered.

- If the user is new user, they have to first create account for the game. For this, they have to enter the details-username, email id and password. These details are then validated and are then stored in users database of the game, after successful registration, users have to login for the game and then goes to the next page.

3.6.2 Finding of random player

When the clicks on the play button, search for a random player gets started with whom the player will play the game. When the player clicks on start button their status in the database is set to 1. Then random player is searched with the help of this status. We check for all the players in the database whose status are set to 1 and selects any one player with whom game will be played. If no user is found bot (computer) will act as a random player.

3.6.3 Start the game

After findings the random player for the game, a message is shown that the player has been found, whether you want to start the game or not. The game is not started until the player clicks on the start and until other player has not started. Also the player is asked to select some features show on the screen.

3.6.4 Select features

After clicking the start button the user have to select the various features from the multiple features shown as per their choice.

3.6.5 Game Play

After both the players has started the game ten same pair of images(having gender biasing) are shown to both the player and they are asked to choose one image from each pair which they think their partner would prefer. Each player clicks on the image they think their partner would like and then after clicking the image, they goes to the next image and the game continues so on.

3.6.6 Maintaining the score

If both the players click on the same image, both the players will get the score, else none will get the score and their score remains the same i.e their will be no reduction in the score if both of them clicks the different image. With every same click the score will increase exponentially as it will make the game more interesting.

The separate tables are maintained in the database for both the users. In each table we will store the judgments of the user's choice. The judgment will consist the name of the image that the user has chosen against the other image.

3.6.7 Predicting the gender

At the end, the gender of both the users will be predicted.

3.6.8 collabrative filtering

On the basis of features selected by the user on the start page the user preferences are calculated.

- Flow chart of the game

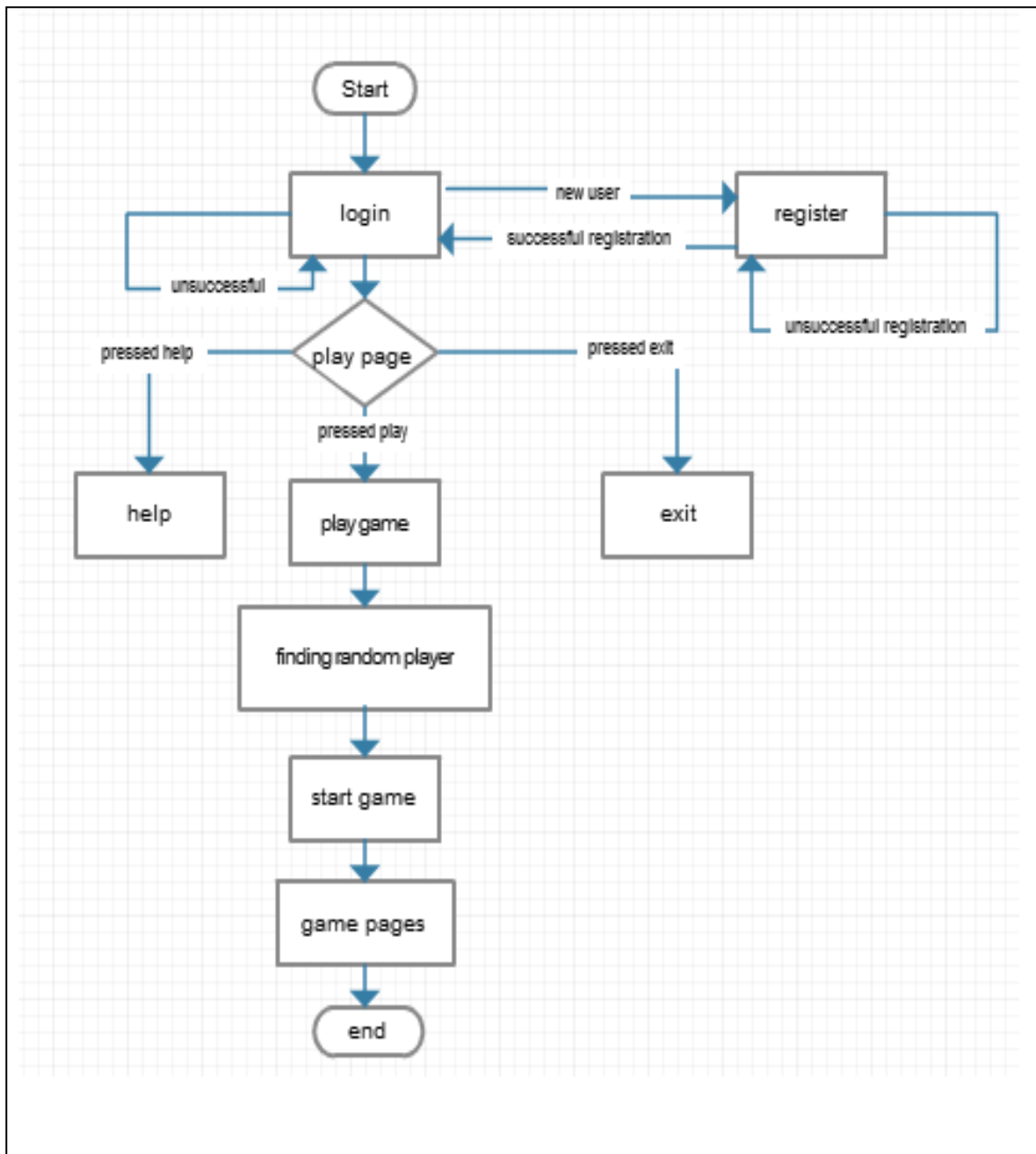


Fig 3.6.1 flow chart of the game

This chapter includes the detailed design, procedure and technologies used. The system's mock screen shots of currently implemented functionalities are also shown.

4.1. Technologies Used

This section describes the front and the back end technologies that will be used to implement the system.

4.1.1. Front-End Technologies

Being a web application, the front end, i.e. the user interface is designed using HTML, CSS and JavaScript.

4.1.2. Back End Technologies

The language used for back end development of web applications is JAVA.

MySQL is used as database system. Tool used for back end development is NetBeans IDE 7.2.

4.2 Users of the project

The game is relevant for three broad classes:

4.2.1 Administrator:

- **Maintenance**

The administrator's primary job is the maintenance of the game. He/she must ensure the smooth functioning of the game. Other tasks include ensure that policies of the game are respected.

- **Database administrator:**

The administrator is responsible for ensuring that the data gets stored properly and safely in the database.

4.2.2 User:

The user will use the game for their entertainment.

4.2.3 Other people:

This game is also useful for the other people as the preferences will result in the useful information which will be helpful for many other applications.

4.3 Product perspective

4.3.1 System interfaces

The product runs from within a web-browser. It must be flexible enough to run easily with a variety of operating systems, computer architectures and be simple to use.

4.3.2 User Interfaces

The game will have user-friendly interfaces.

4.4 User characteristics

- The user doesn't require any special technical skills
- The user must know how to operate internet

4.5 Constraints

- The product must be web-based.
- There are no memory requirements
- The computer must be equipped with web browser

4.6 Modules

The entire project has been divided into four basic modules which are as follows:

- **Module 1 : Online game**
- **Module 2 : Input data set**
- **Module 3 : Gender prediction**
- **Module 4 : Collaborative filtering**

4.7 Module 1 : Online game

- **Product functions**

USE CASE	DESCRIPTION
Login	Submit Register Forgot password
Register	Register Login
Play	Play Help Exit
Feature	Feature 1-----feature n
Start	Start
First page	Image A Image B
Second page	Image A Image B
Third page	Image A Image B

Fourth page	Image A Image B
Fifth page	Image A Image B
Sixth page	Image A Image B
Seventh page	Image A Image B
Eighth page	Image A Image B
Ninth page	Image A Image B
Tenth page	Image A Image B

Fig 4.7.1 table showing functions of online game

4.7.1 User Login

- The user has to login to play the game.
- It provides authentication by verifying the identity of the user in terms of username and password.

- The front end of this functionality is developed using HTML and CSS, and JavaScript is used for form validations. JAVA servlets is used for backend. JDBC is used for communication with the database.

- **Flow Chart of login functionality**

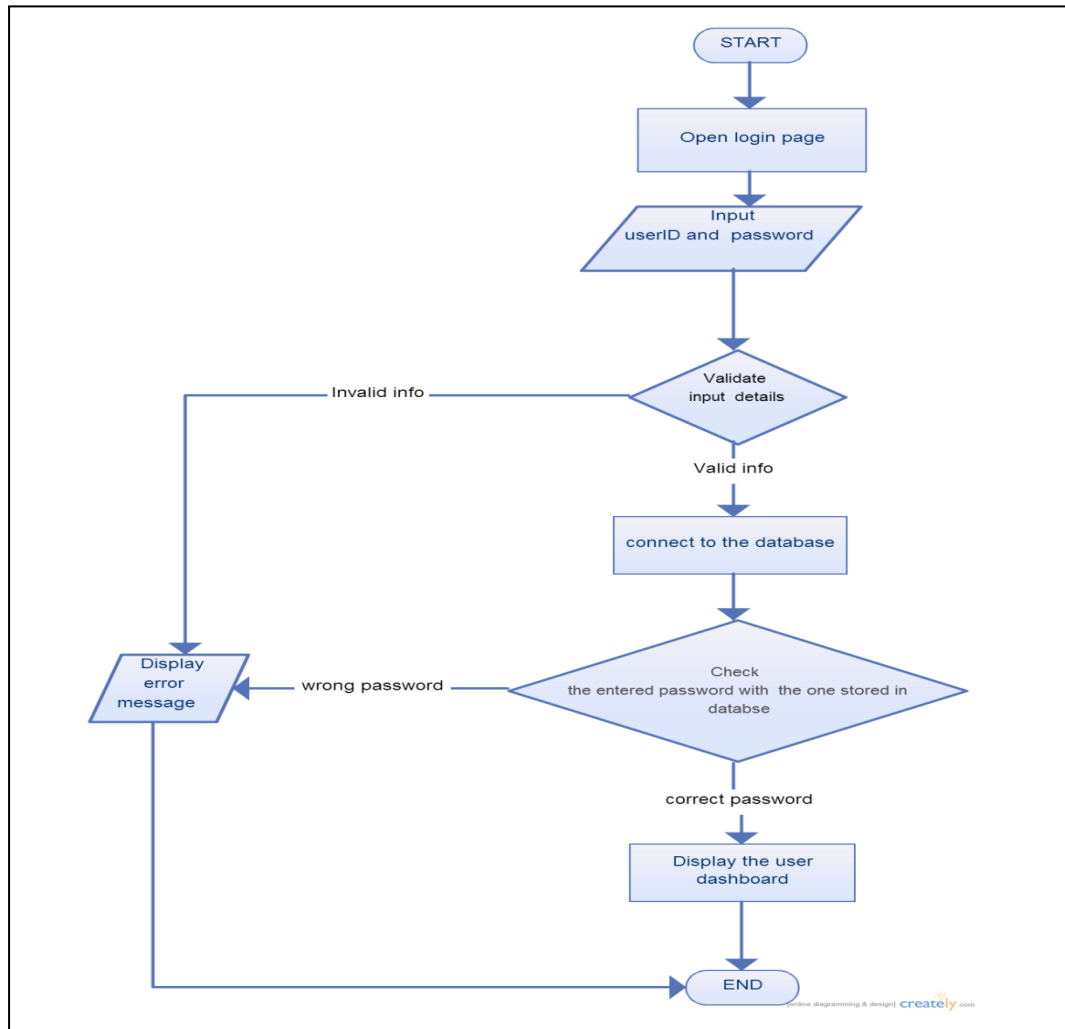


Fig. 4.7.1.1. Flow Chart of the Login functionality

- **User Interface of login page**

Given below is a screen shot of the user interface of the login page.

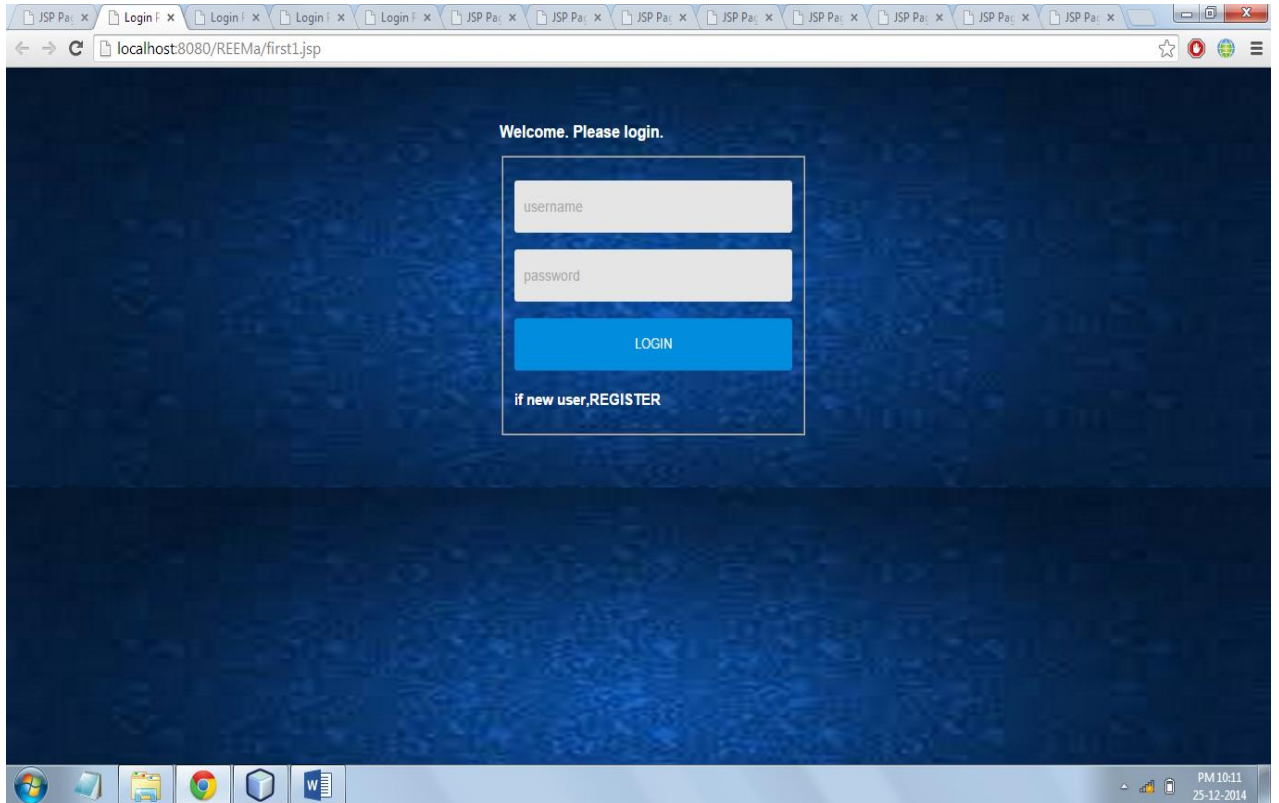


Fig. 4.7.1.2. UI of the Login Page

1. Username: Numeric allowed , alphanumeric allowed , special characters not allowed
2. Password : numeric allowed , alphanumeric allowed, length must be less than 20

- **Incorrect login UI**

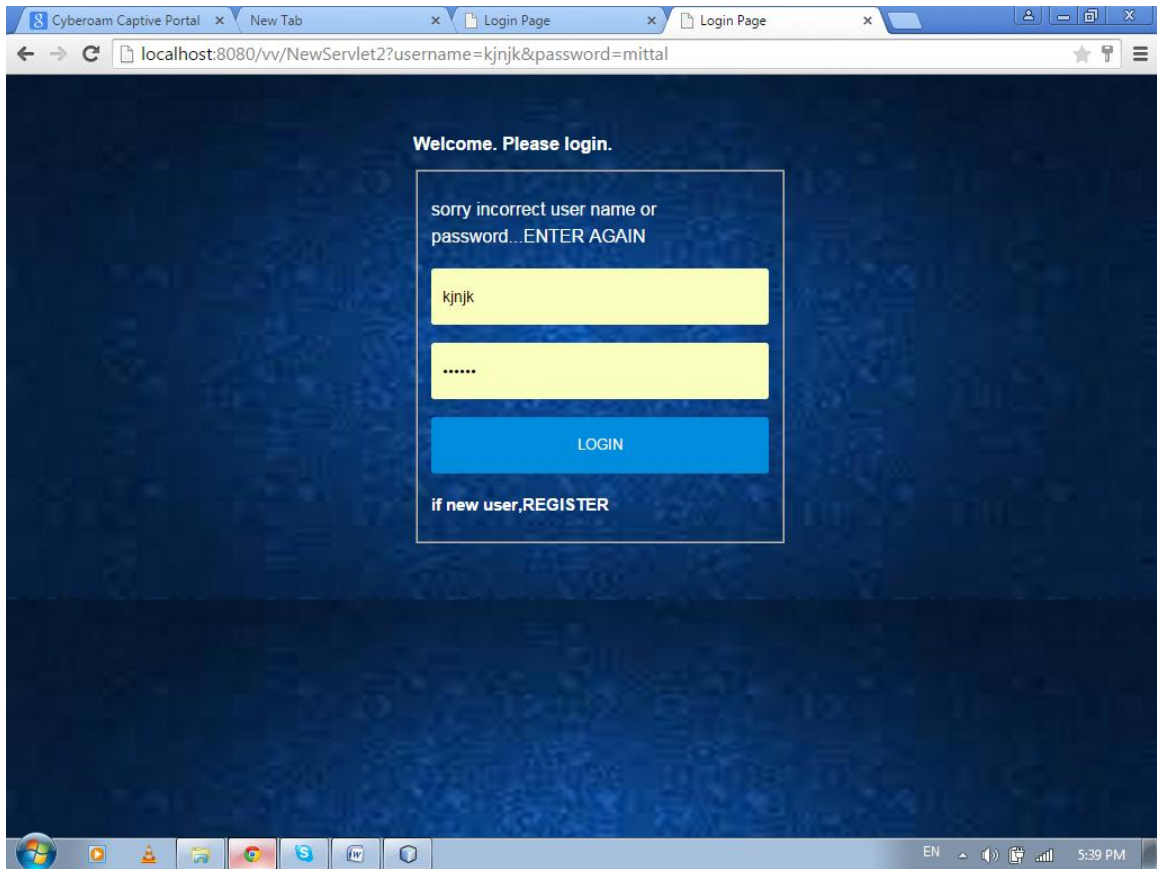


Fig. 4.7.1.3. UI of the incorrect Login

4.7.2 User registration

The user need to register with the system to play this game. The front end of this functionality is developed using HTML and CSS, and JavaScript is used for form validations. JAVA servlets is used for backend. JDBC is used for communication with the database.

- **Flow Chart of user registration**

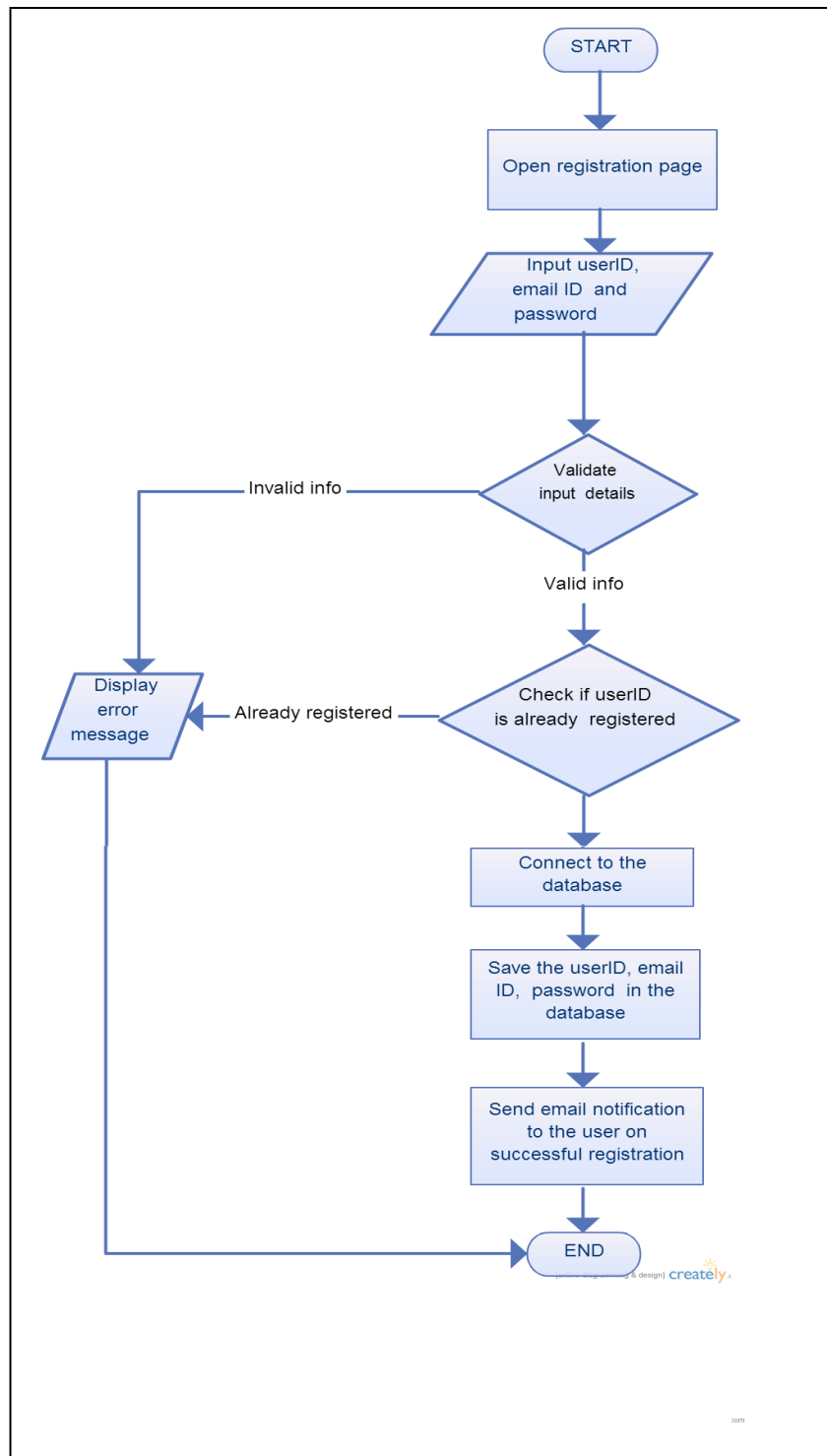


Fig. 4.7.2.1 Flow chart for User Registration

4.7.2.1 Email Notification

The system will send email notifications to the user on various events. Currently, the system sends an email notification to the user on successful registration and mentions the user ID and the password in the mail.

The mail is send to the email ID entered by the user while registering.

Some important points about the implementation of this functionality:

- **JavaMail API**

The module uses JavaMail API. The JavaMail API provides a platform-independent and protocol-independent framework to build mail and messaging applications. The JavaMail API is available as an optional package for use with the JavaSE platform and is also included in the Java EE platform. The javax.mail package defines classes that are common to all mail systems. The javax.mail.internet package defines classes that are specific to mail systems based on internet standards such as MIME, SMTP, POP3, and IMAP.

In the implementation, we use the SMTP Server provided by Gmail.

- **Steps to send email using JavaMail API**

There are following three steps to send email using JavaMail. They are as follows:

1. Get the session object that stores all the information of host like host name, username, password etc.
2. compose the message
3. send the message

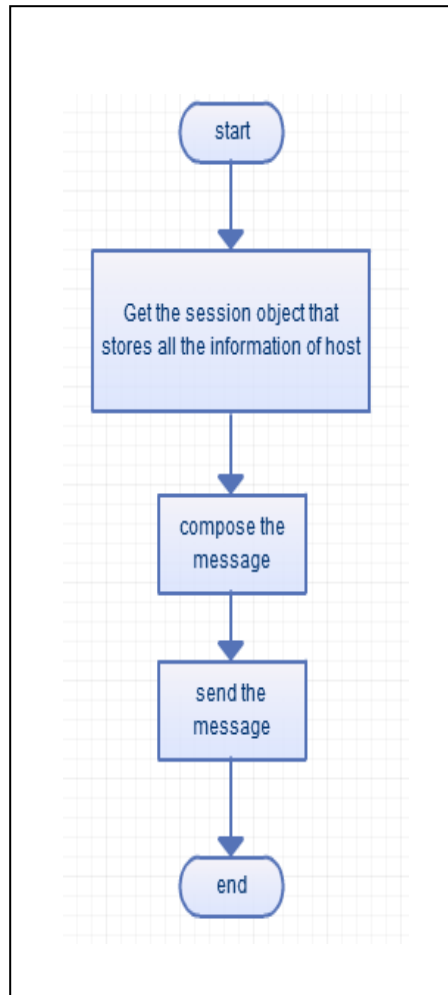


Fig 4.7.2.1.1 Flow Chart of the Email Notification

- **User Interface of Registration page**

.Given below is a screen shot of the user interface of the registration page.

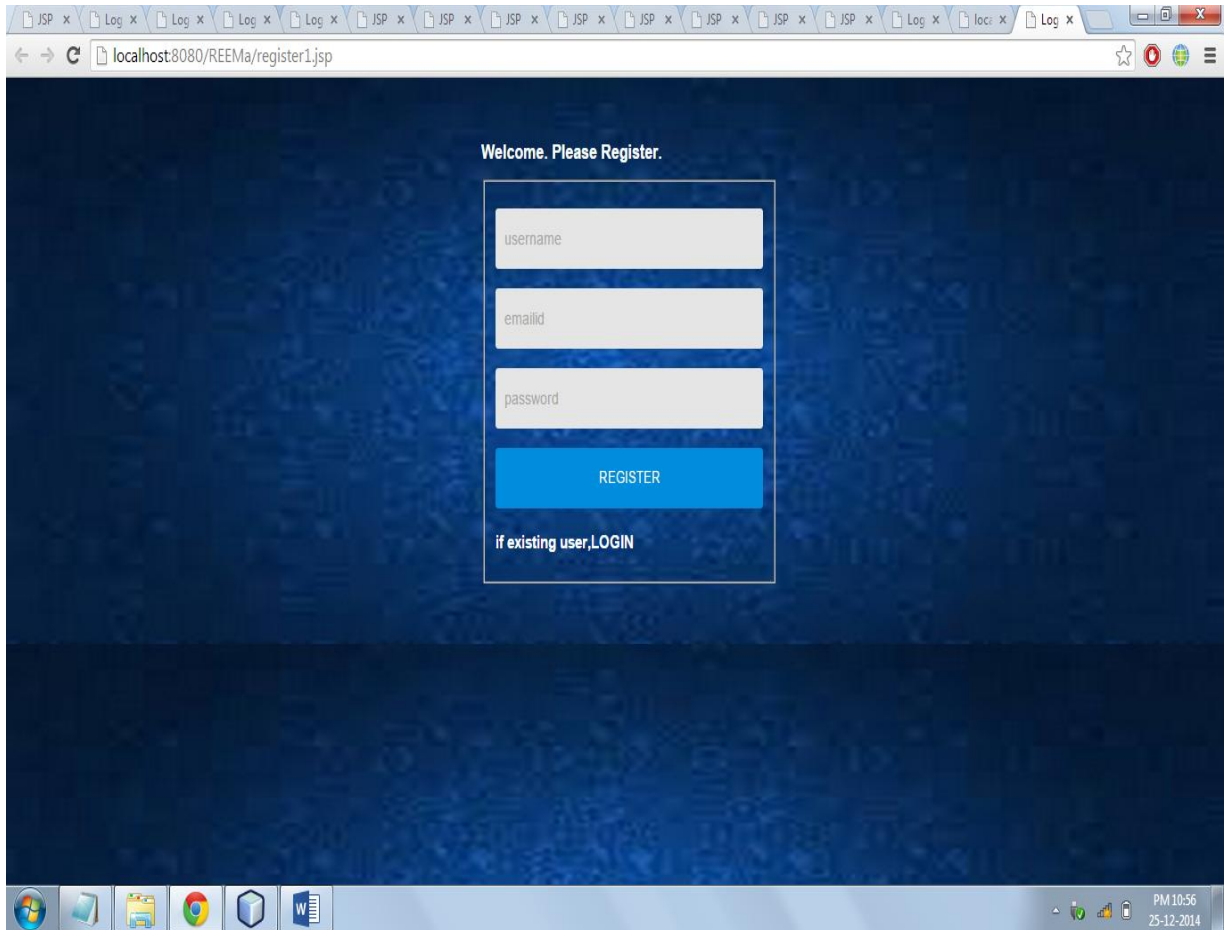


Fig. 4.7.2.1.2 UI for User Registration

1. User name: Numeric allowed , alphanumeric allowed , special characters not allowed
2. Email id: Numeric allowed , alphanumeric allowed , special characters allowed, length of characters should be less than 35, minimum one '@' and one '.' is mandatory
3. Password: : Numeric allowed , alphanumeric allowed , length must be less than 20 characters.

- **User interface for unsuccessful registration**

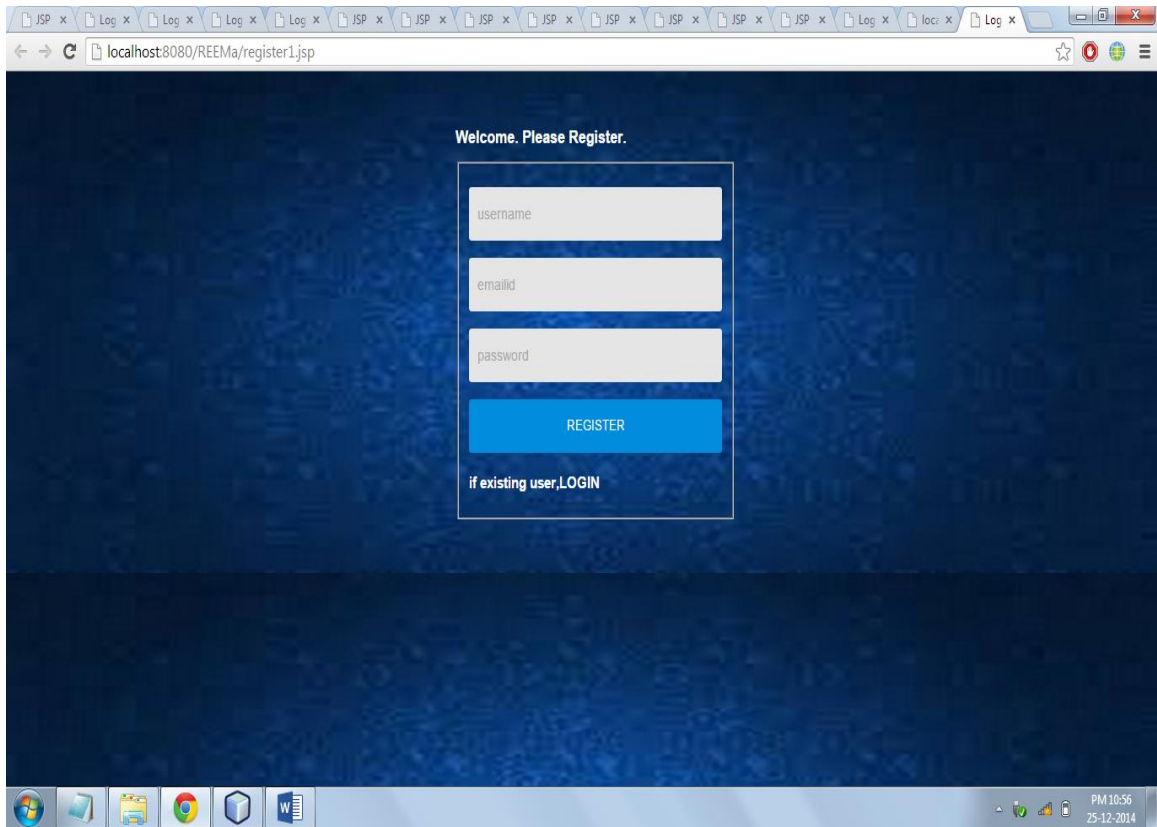


Fig 4.7.2.1.3 UI for unsuccessful User Registration

- **User interface to notify the user that he has been registered.**

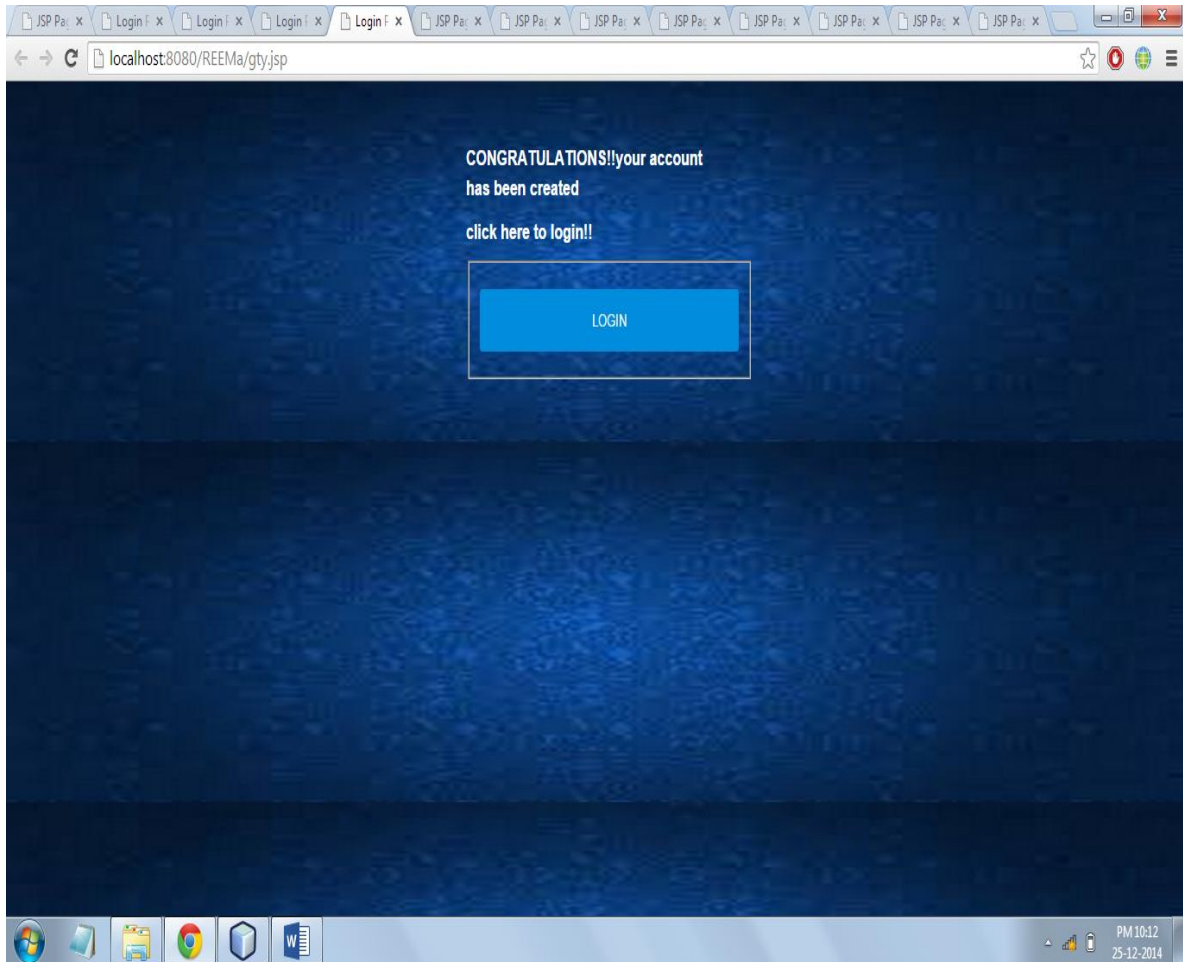


Fig. 4.7.2.1.4 UI for Successful Registration

On clicking login user will go the login page shown above.

4.7.3 Play Web Page

Once the user gets registered, he comes to the play page. The front end of this functionality is developed using HTML and CSS. JAVA servlets is used for backend. JDBC is used for communication with the database

- **User Interface of play page**

Given below is a screen shot of the user interface of play page.

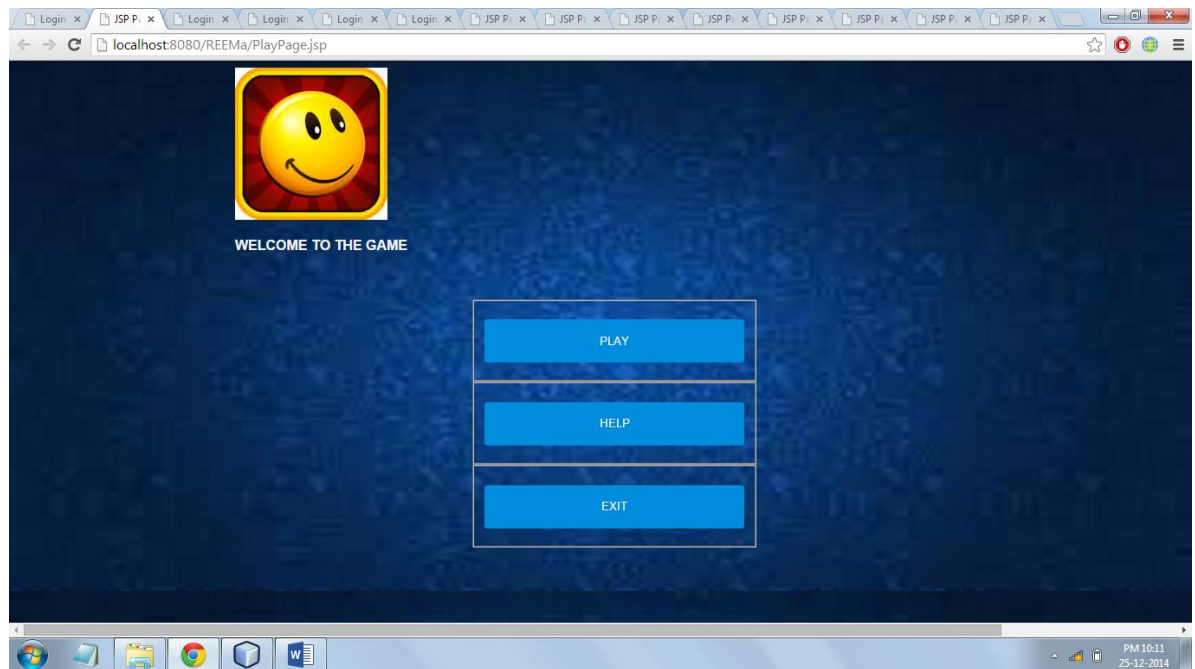


Fig. 4.7.3.1 UI for Play Game Page

1. Play: on clicking play button the user will go to the next page
2. Help: on clicking help button the user will be directed to the page having the instructions regarding the game
3. Exit : this button will exit the user

4.7.4 Select features

on clicking play button the user will come to the feature page including the following shown features . the user have to select his/her desired features

- **User interface of select features page**

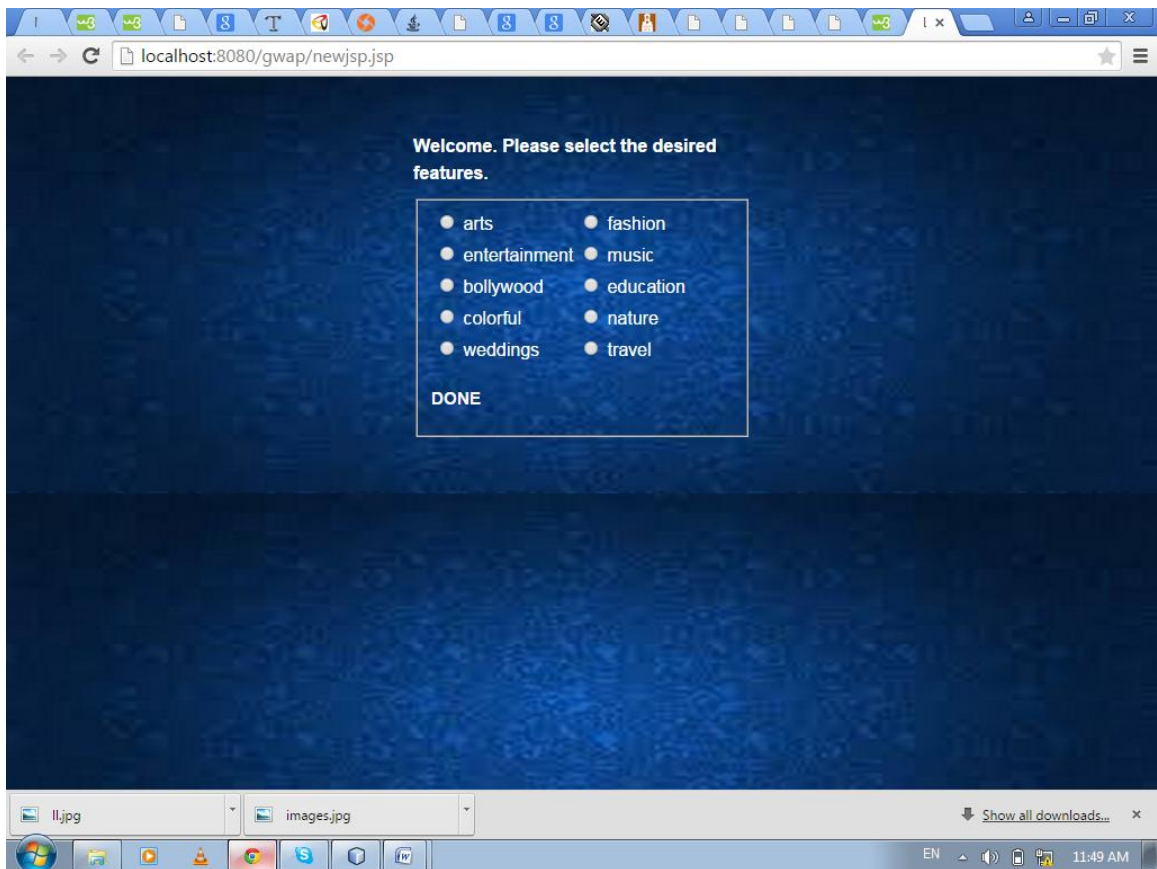


fig 4.7.4.1 UI for select features Page

4.7.5 Start Game Page

On clicking to done button the user comes to the start page. The front end of this functionality is developed using HTML and CSS. JAVA servlets is used for backend. JDBC is used for communication with the database

- **User Interface of start game page**

Given below is a screen shot of the user interface of play page.

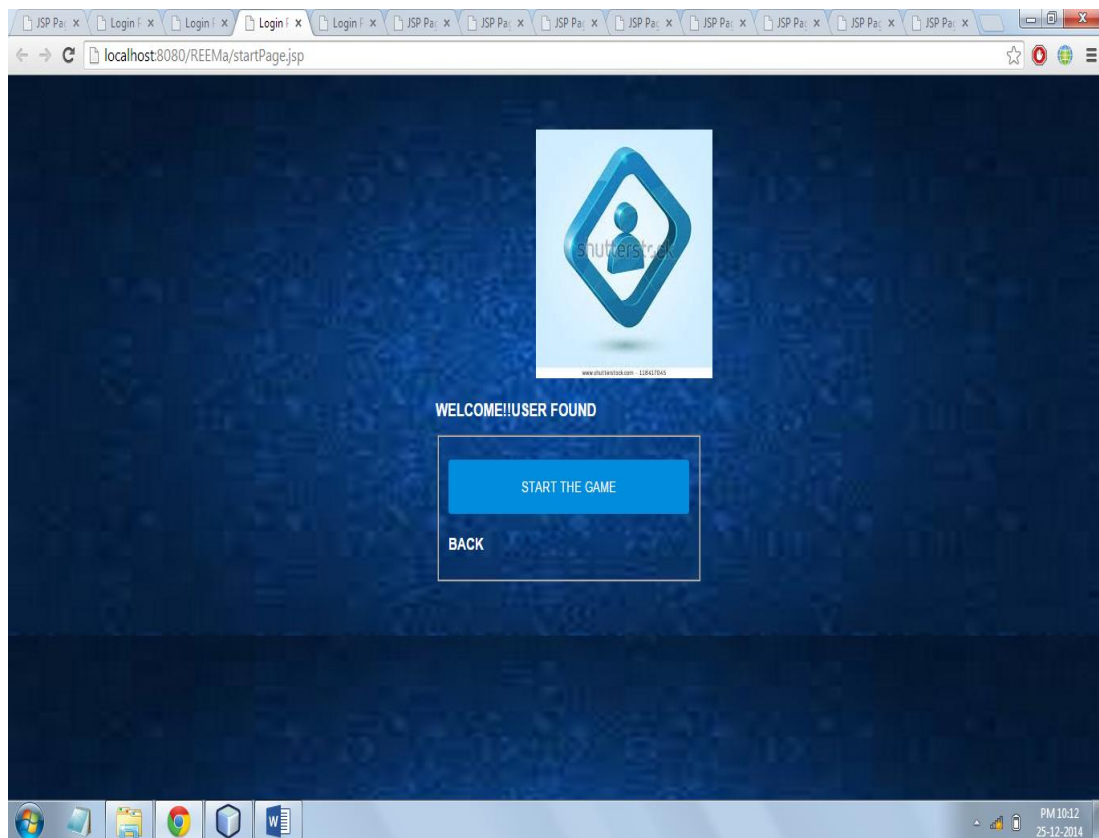


Fig. 4.7.5.1 UI for Start Game Page

4.7.6 Game pages

On clicking to start button, he comes to the game pages. The front end of this functionality is developed using HTML and CSS. JAVA servlets is used for backend. JDBC is used for communication with the database

- **User Interface**

Given below is a screen shots of the user interface of game page.

- **First page**

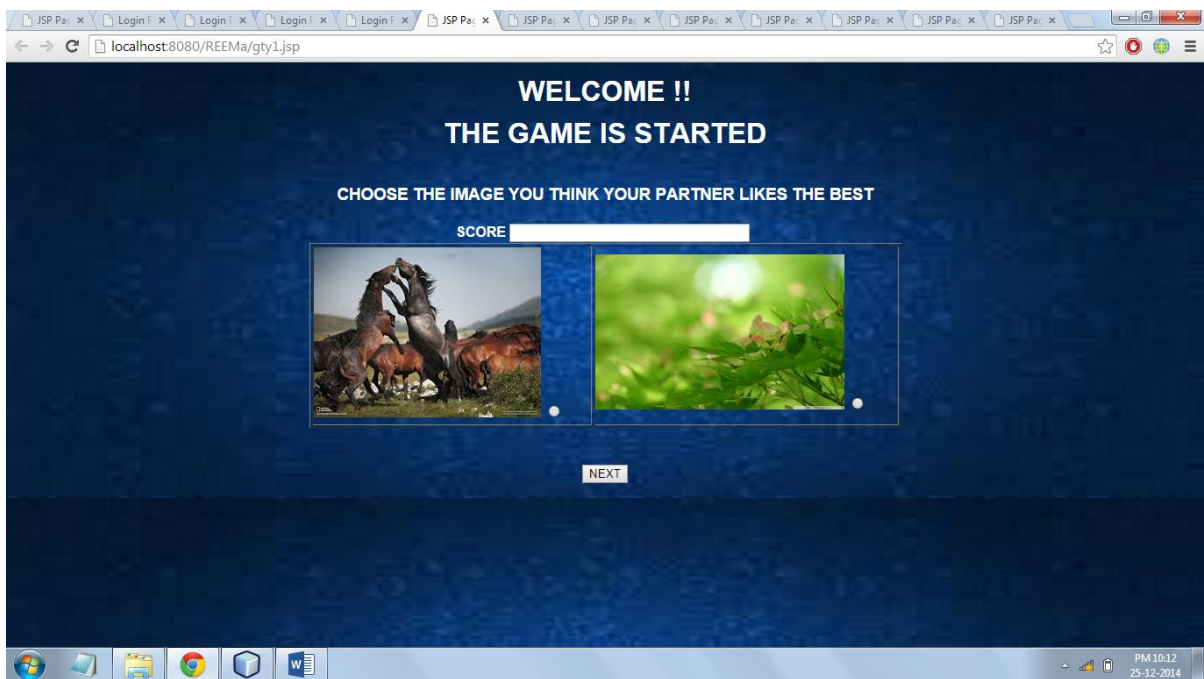


Fig. 4.7.6.1 UI for first page

➤ Second page

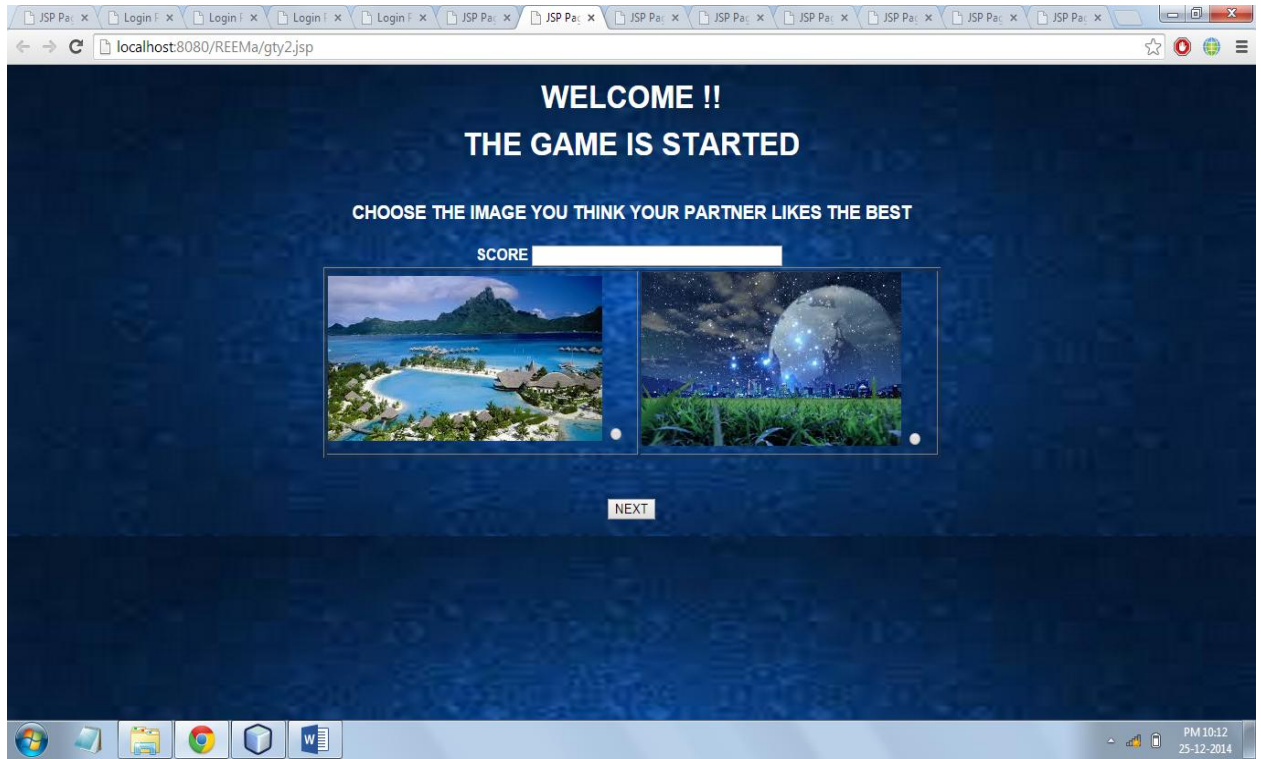


Fig. 4.7.6.2 UI for second page

➤ Third page

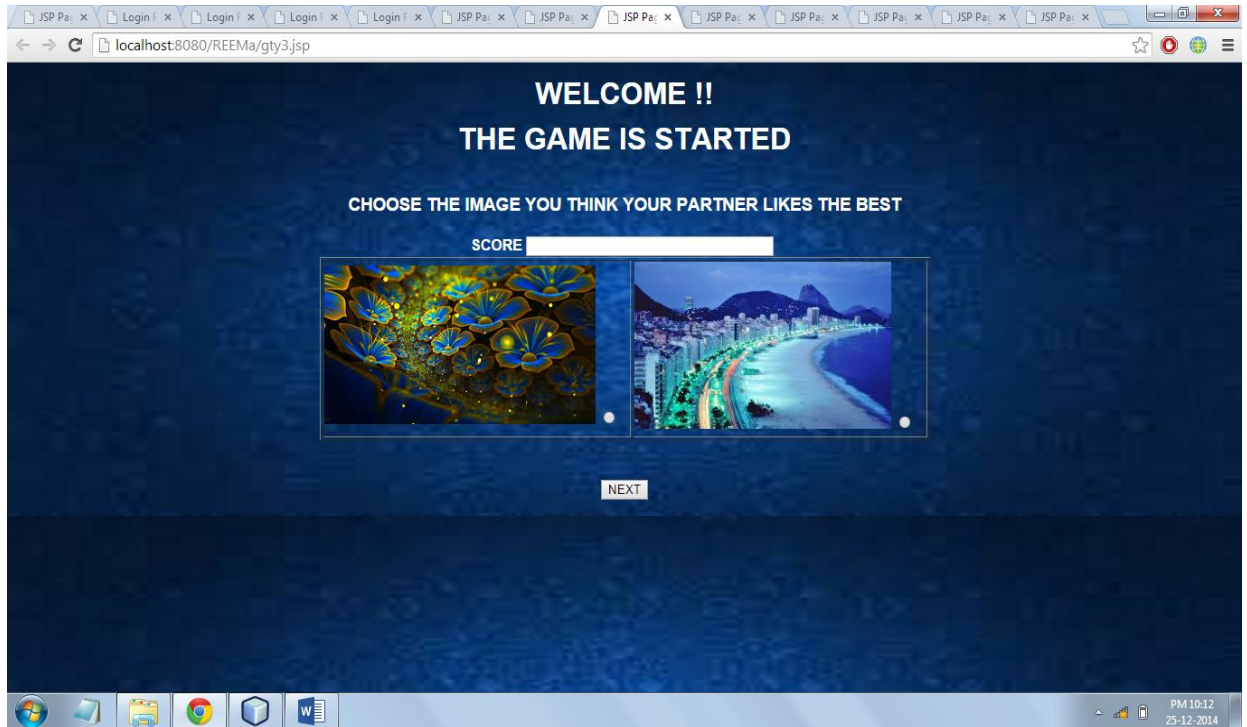


Fig. 4.7.6.3 UI for third page

➤ Fourth page

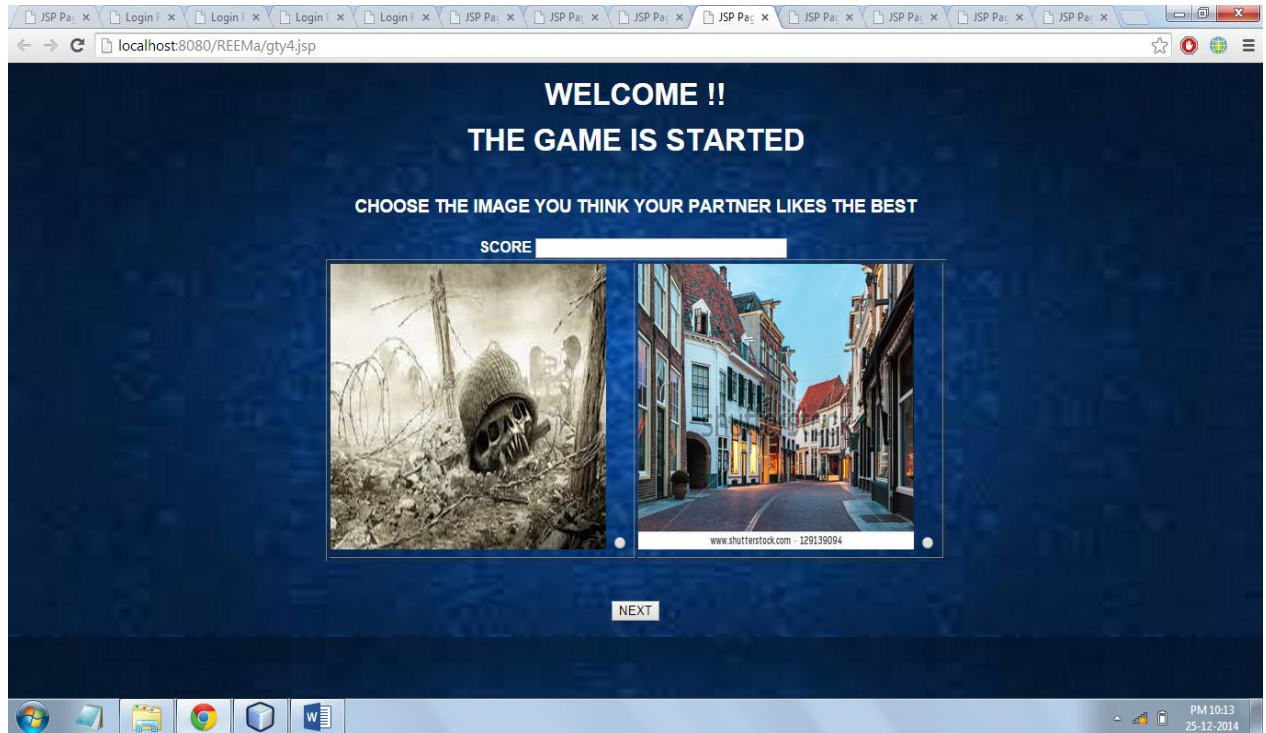


Fig. 4.7.6.4 UI for fourth page

➤ Fifth page

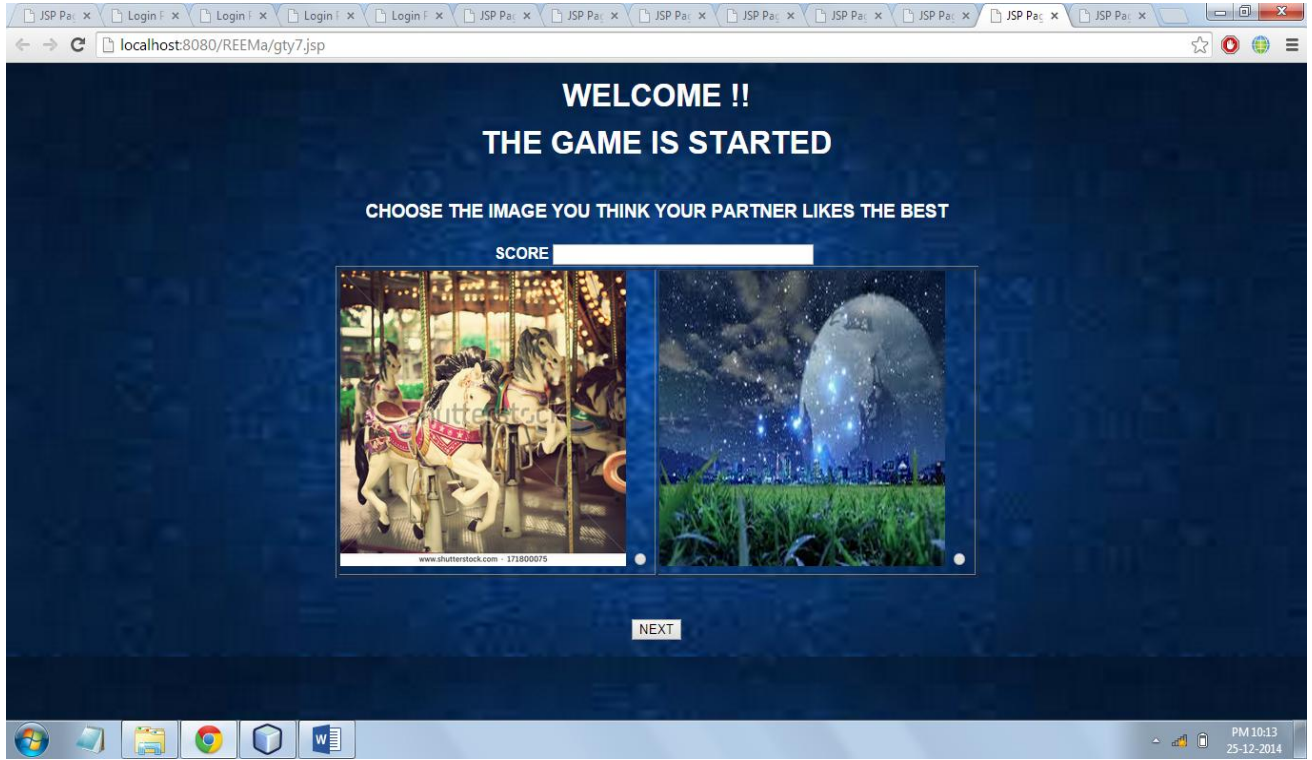


Fig. 4.7.6.5 UI for fifth page

➤ Sixth page

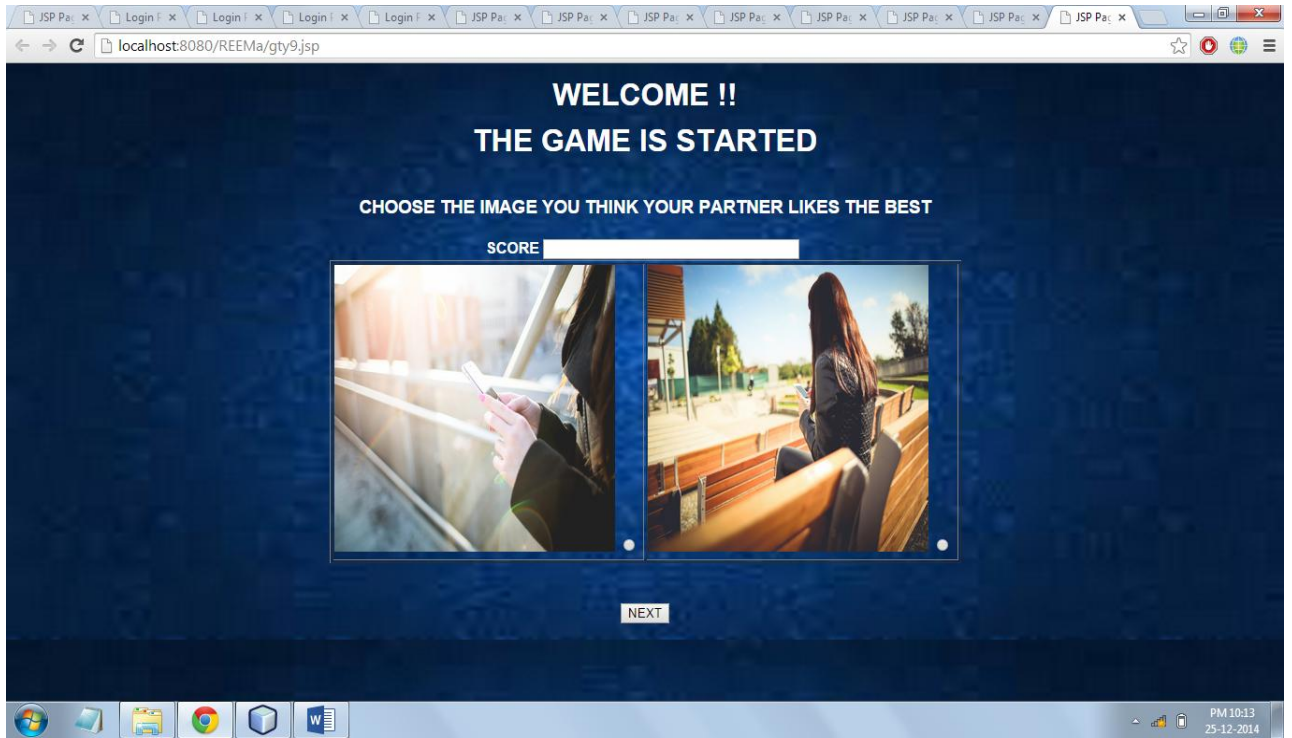


Fig. 4.7.6.6 UI for sixth page

4.8 Module 2: Finding the input data set

- **Finding gender- biased images**

For the gender prediction, we have to find pair of images (A, B) that has a strong gender bias. The strong gender bias images are selected on the basis of parameter conditional entropy $H[G|X]$ of the player's gender G given that we know the player's decision X , where $A > B$ means that image A was considered better than image B :

$$H[G|X] = \sum_{x \in \{A < B, A > B\}} \Pr(x) H[G|X = x]$$

A pair (A, B) has a large gender bias if the conditional entropy $H[G|X]$ is small

4.8.1 Algorithm to find gender biased images

Input-table gp (table in database storing the number of males and females who have chosen image A and image B over a pair of images (A,B), table gp1 (table in database storing the pair of Images having gender biasing)

Start

```
//count the number of pair of images
```

```
//create query
```

```
query1 = DB :: select (count(*)) -> from gp
```

```
//execute query
```

```
Results1 = query1->execute(null,false);
```

```
For i=1 to results1
```

```

// query to count no of females who have chosen image A over a pair of image(A,B)
    Query2 = DB :: select ('fa') -> from gp -> ('sno','=',i)
//execute query
    Results2 = query2->execute(null,false);
// query to count no of females who have chosen image B over a pair of image(A,B)
    Query3 = DB :: select ('fb') -> from gp -> ('sno','=',i)
//execute query
    Results3 = query3->execute(null,false);
// query to count no of males who have chosen image A over a pair of image(A,B)
    Query4 = DB :: select ('ma') -> from gp -> ('sno','=',i)
//execute query
    Results4 = query4->execute(null,false);
// query to count no of males who have chosen image B over a pair of image(A,B)
    Query5 = DB :: select ('mb') -> from gp -> ('sno','=',i)
//execute query
    Results5 = query5->execute(null,false);
//total number of females
    Tn= Results2 + Results3;
//total number of males
    Tm = Results4 + Results5;
    P1=Results2/(Results2+Results4)
    P2=Results3/(Results3+Results5)
    P3=P1+P2
        If p1>.9 and p1<1.05
// query to extract id of image A satisfying the above relation

```

```
Query46= DB :: select ('ida') -> from gp -> ('sno','=',i)
//execute query
Results6 = query6->execute(null,false);
// query to extract id of image B satisfying the above relation
Query7 = DB :: select ('idb') -> from gp -> ('sno','=',i)
//execute query
Results7 = query7->execute(null,false);
// query to insert these pair of images in new table
Query8 = DB :: insert ('ida','idb') -> into gp1
//execute query
query7->executeUpdate(null,false);
End for
```

End

OUTPUT- pair of gender biased images

4.9 Module 3: Gender prediction

After the player has played the game his/her preferences has been stored in the database and the gender of the player will be extracted on the basis of the preferences stored in the database

4.9.1 Algorithm for gender prediction

Input-table gp (table in database storing the number of males and females who have chosen image A and image B over a pair of images (A,B)), table userchoice (table in database storing the judgements/ preferences of the user over the pair of images (A,B))

Start

Set n=10 //number of pair of images

Initialize sum=0;

For i=1 to n

// query to extract the image selected by the user over a pair of image (A.B)

Query2 = DB :: select ('ch') -> from userchoice ->where ('sno','=',i)

//execute query

Results2 = query2->execute(null,false);

// query to find id of image A

Query3 = DB :: select ('ida') -> from userchoice -> where('sno','=',i)

//execute query

Results3 = query3->execute(null,false);

// query to find id of image B

Query4 = DB :: select ('idb') -> from userchoice-> where ('sno','=',i)

```

//execute query

    Results4 = query4->execute(null,false);

// query to find number of females who have chosen image A over a pair of image
(A,B) whose ids are(ida,idb)

    Query5 = DB :: select ('fa') -> from gp -> where('ida','=',results3 and
'adb','=',results4)

//execute query

    Results5 = query5->execute(null,false);

// query to find number of females who have chosen image B over a pair of image
(A,B) whose ids are(ida,idb)

    Query6 = DB :: select ('fb') -> from gp -> where('ida','=',results3 and
'adb','=',results4)

//execute query

    Results6 = query6->execute(null,false);
    Y=Results5+Results6

// query to find number of males who have chosen image A over a pair of image
(A,B) whose ids are(ida,idb)

    Query7 = DB :: select ('ma') -> from gp -> where('ida','=',results3 and
'adb','=',results4)

//execute query

    Results7 = query7->execute(null,false);

// query to find number of males who have chosen image B over a pair of image
(A,B) whose ids are(ida,idb)

    Query8 = DB :: select ('mb') -> from gp -> where('ida','=',results3 and
'adb','=',results4)

```

```
//execute query
    Results8 = query8->execute(null,false);
    Y1=Results7+Results8
End for
If ch==A
    Sum+=(results5/y)
    Sum1+=(results7/y1)
End if
Else
    Sum=+(results6/y)
    Sum1=+(results8/y1)
End else
If sum>sum1
Gender is gemale
Else
gender is male
END

Output-gender is male or female
```

4.10 Module 3: Collaborative Filtering

This allows us to recommend images to each user based on his/her preferences. Also, we can compare users and images with each other which users are similar and which images are similar.

In this model, each user i and each image j is described by a feature vector of length K . We store the user feature vectors in a $nusers \times K$ matrix \mathbf{U} , where each row is a user's feature vector. Similarly, we store the image feature vectors in a $nimages \times K$ matrix \mathbf{V} . We say that the amount by which user i likes image j is equal to the dot product of their feature vectors.

4.10.1 Algorithm for collaborative filtering

Input-table imgfeature(storing the information about features of the images)

Table userfeature(storing the information about preferable features of the user)

```
START  
  
Set n=no of users  
  
Set m=no of images  
  
Set y=no of features  
  
Initialize sum=0  
  
For each user i  
    For each feature k  
        Set U[i][k]= value stored in table userfeature for user i and feature k  
    End for  
  
End for  
  
For each image i
```



```

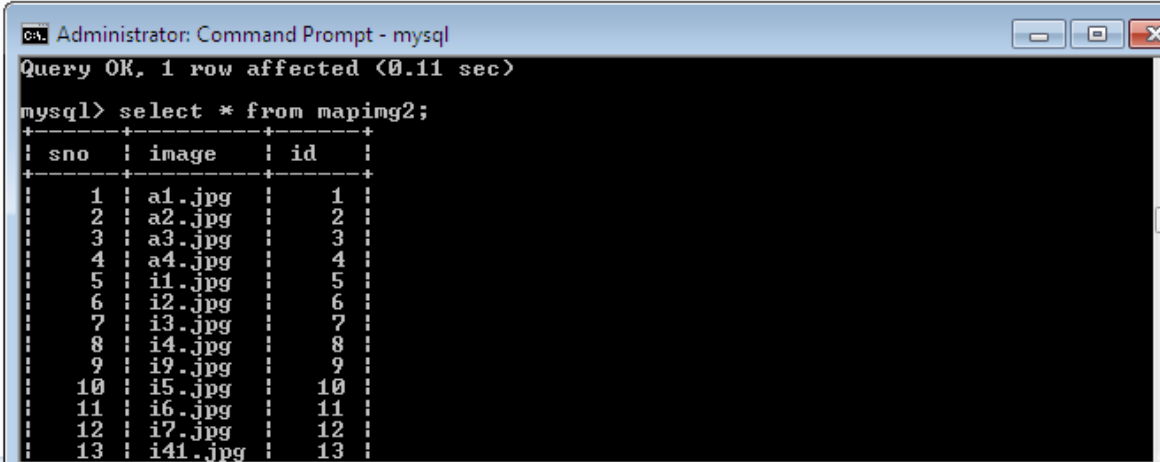
    For each feature k
        Set  $V[i][k]$ = value stored in table imgfeature for user i and feature k
    // find transpose of matrix U
    End for
End for
For i=1 to n
    For j=1 to k
         $D[k][i]=V[i][k]$ 
    //dot product of two matrices
    End for
End for
For i=1 to n
    For j=1 to m
        For k=1to y
             $Sum=sum + U[i][k]*D[k][j]$ 
        End for
         $N[i][j]=sum$ 
    End for
Set Sum=0
End for
OUTPUT – which images will be liked by the user and which are not .

```

This chapter would include the various inputs used in the above mentioned algorithms along with the desired output obtained.

5.1 Mapping of images

The mapping of images to the ids will make the usage of images in the code easier and short. The images used for the game has been given a specific id which is shown as below:



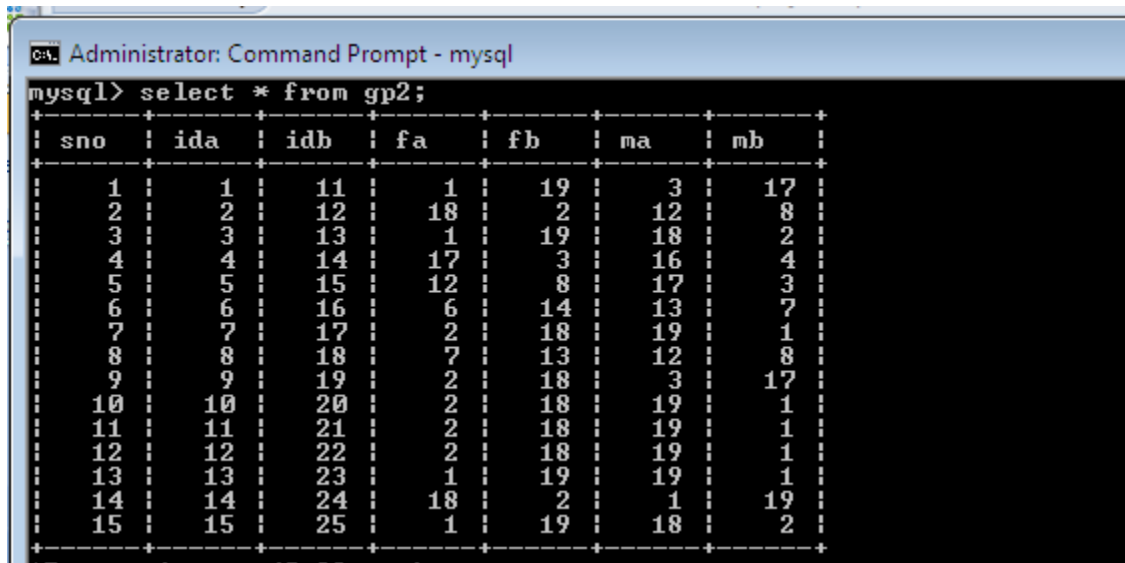
```
Administrator: Command Prompt - mysql
Query OK, 1 row affected (0.11 sec)
mysql> select * from mapping2;
+----+-----+----+
| sno | image | id  |
+----+-----+----+
| 1   | a1.jpg | 1   |
| 2   | a2.jpg | 2   |
| 3   | a3.jpg | 3   |
| 4   | a4.jpg | 4   |
| 5   | i1.jpg | 5   |
| 6   | i2.jpg | 6   |
| 7   | i3.jpg | 7   |
| 8   | i4.jpg | 8   |
| 9   | i9.jpg | 9   |
| 10  | i5.jpg | 10  |
| 11  | i6.jpg | 11  |
| 12  | i7.jpg | 12  |
| 13  | i41.jpg | 13  |
+----+-----+----+
```

Fig 5.1.1 Database view of mapping of images to the specific ids.

1. sno : serial number of the images
2. image : url of the image
- 3 id : id given to the image

5.2 Input for gender biased images algorithm

Here we will show the input which is being used for predicting gender biased images. The data of some users whose gender is already known is being stored and is being used for predicting gender biased images. Following is the table in database storing the number of males and females who have chosen image A and image B over a pair of images (A,B)



```
mysql> select * from gp2;
```

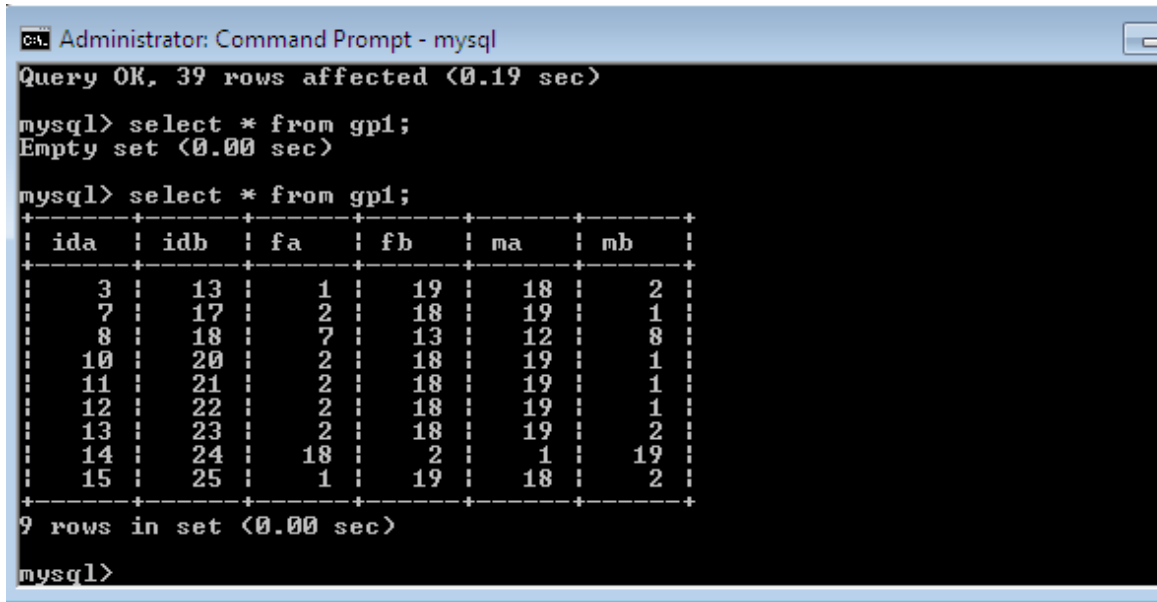
sno	ida	idb	fa	fb	ma	mb
1	1	11	1	19	3	17
2	2	12	18	2	12	8
3	3	13	1	19	18	2
4	4	14	17	3	16	4
5	5	15	12	8	17	3
6	6	16	6	14	13	7
7	7	17	2	18	19	1
8	8	18	7	13	12	8
9	9	19	2	18	3	17
10	10	20	2	18	19	1
11	11	21	2	18	19	1
12	12	22	2	18	19	1
13	13	23	1	19	19	1
14	14	24	18	2	1	19
15	15	25	1	19	18	2

Fig 5.2.1 Database view of table gp2

1. Sno : serial number in the table
2. Ida : id of image A over a pair of image (A,B)
3. Idb : id of image B over a pair of image (A,B)
4. Fa : number of females who have chosen image A over image B
5. Fb : number of females who have chosen image A over image A
6. ma : number of males who have chosen image A over image B
7. mb : number of males who have chosen image B over image A

5.3 Output for gender biased images algorithm

Here we will show the output of predicting gender biased images algorithm. Following is the table in database storing the pair of Images and related information having gender biasing.

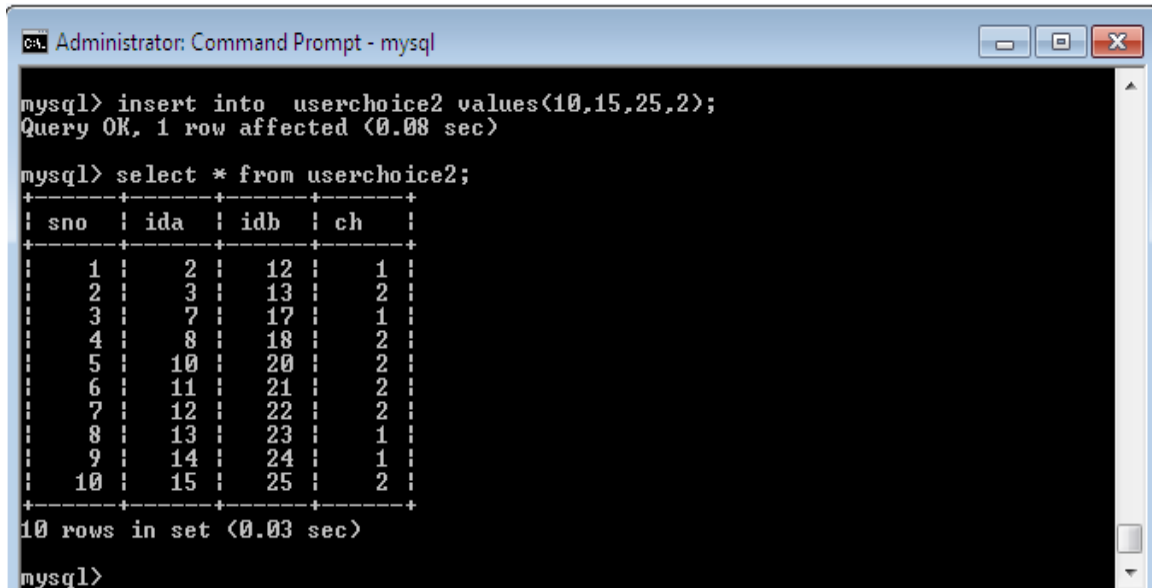


```
Administrator: Command Prompt - mysql
Query OK, 39 rows affected (0.19 sec)
mysql> select * from gp1;
Empty set (0.00 sec)
mysql> select * from gp1;
+----+----+----+----+----+----+
| ida | idb | fa  | fb  | ma  | mb  |
+----+----+----+----+----+----+
| 3   | 13  | 1   | 19  | 18  | 2   |
| 7   | 17  | 2   | 18  | 19  | 1   |
| 8   | 18  | 7   | 13  | 12  | 8   |
| 10  | 20  | 2   | 18  | 19  | 1   |
| 11  | 21  | 2   | 18  | 19  | 1   |
| 12  | 22  | 2   | 18  | 19  | 1   |
| 13  | 23  | 2   | 18  | 19  | 2   |
| 14  | 24  | 18  | 2   | 1   | 19  |
| 15  | 25  | 1   | 19  | 18  | 2   |
+----+----+----+----+----+----+
9 rows in set (0.00 sec)
mysql>
```

Fig 5.3.1 Database view of table gp1

5.4 Input for gender prediction algorithm

After the player has played the game, the preferences of the player will get stored in the database which will act as a input for gender prediction algorithm. Following figure shows the preferences of the user who has played the game.



```
Administrator: Command Prompt - mysql
mysql> insert into userchoice2 values(10,15,25,2);
Query OK, 1 row affected (0.08 sec)

mysql> select * from userchoice2;
+----+----+----+----+
| sno | ida | idb | ch |
+----+----+----+----+
| 1   | 2   | 12  | 1  |
| 2   | 3   | 13  | 2  |
| 3   | 7   | 17  | 1  |
| 4   | 8   | 18  | 2  |
| 5   | 10  | 20  | 2  |
| 6   | 11  | 21  | 2  |
| 7   | 12  | 22  | 2  |
| 8   | 13  | 23  | 1  |
| 9   | 14  | 24  | 1  |
| 10  | 15  | 25  | 2  |
+----+----+----+----+
10 rows in set (0.03 sec)

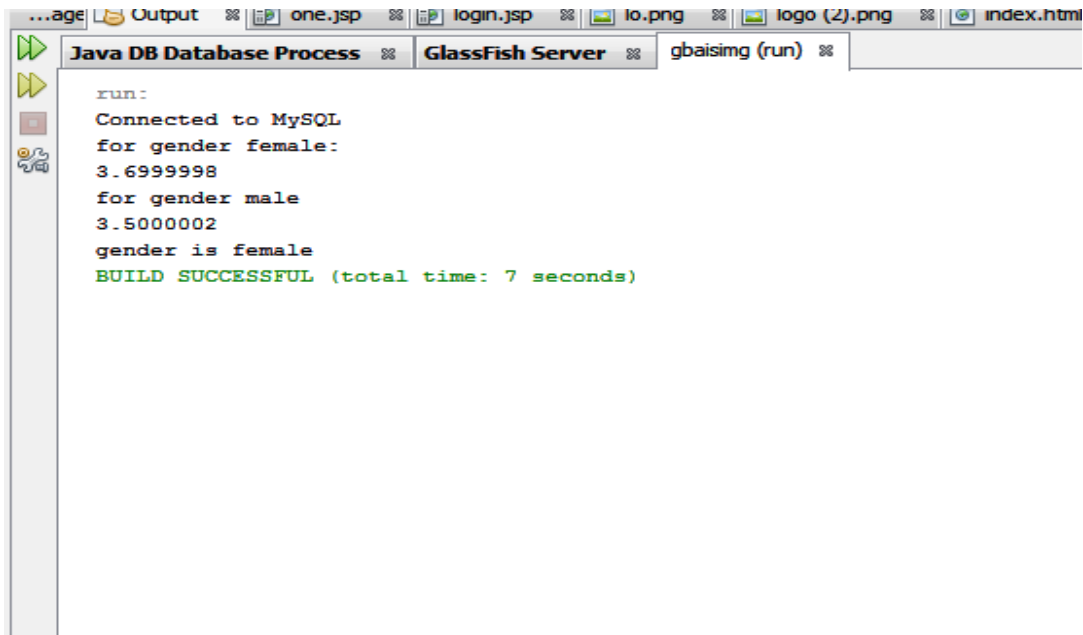
mysql>
```

Fig 5.4.1 Database view of table showing the preferences made by the user over the pair of image (A,B)

1. ida : id of image A
2. idb : id of image B
3. Ch : choice made by the player(1->A,2->B)

5.5 Output of gender prediction algorithm

The algorithm for gender prediction discussed in above chapter is being performed on the above shown preferences of the user and the gender is being predicted as shown in following figure.



```
run:
Connected to MySQL
for gender female:
3.6999998
for gender male
3.5000002
gender is female
BUILD SUCCESSFUL (total time: 7 seconds)
```

Fig 5.5.1 Database view of output of gender prediction algorithm

This figure shows the results of algorithm if we assume gender to be female and then male and ultimately compares the result of both and finally predicts the gender.

CHAPTER 6 CONCLUSION AND FUTURE SCOPE

This Chapter would include the Conclusion reached after creating few modules of the Software in regards to the functional requirements of the System. It would also list the future work that is intended to be accomplished.

6.1 Conclusions

In conclusion of this report, the various aspects covered in throughout would be discussed. The project is briefly introduced to make the audience aware of the basic idea behind it. The aim of the project is specified which describes the final result of the project in brief how the online games are useful in eliciting the user preferences is discussed and which results in the motivation behind the creation of this project.

The main contribution of this project is to provide a method to elicit user preferences. For It remains to be investigated how much other personal information can be gathered in the same way as our gender test does. We also discuss the various software requirement and the assumptions and dependencies. Out of the many functionalities that the project aims to implement the online game has been implemented The various details the theses implementations are discussed, along with some output screenshots.

6.2 Future scope of the project

- This project will be used in applications of image search like if in Google we want to search images of particular type that the images that are preferable more by the users will be shown first and so on according to their ranking.
- It also helps in determining the computer vision i.e. used in various algorithms that access the quality of image.
- Similarly rather than images we can even rank books , journals , songs , movies , restaurants , brands etc. and can their global ranking.
- This project will be helpful in various other applications like prediction of election that who will win
- This project will be helpful for e-commerce applications as the products will be shown according their preferences
- This project will help in predicting the type of annotations to be shown to the users while surfing the internet.

REFERENCES

[1] Luis Von Ahn “Matchin: Eliciting User Preferences with an Online Game”, Conf. on Web Games, Preference Elicitation, Human Computation, CHI 2009. pp 1-10

[2] Luis Von Ahn, “Eliciting User Preferences through an Online Game”, ACM Conf. on Human Factors in Computing Systems, CHI 2009. pp 1207-1216

[3] Alexander J. Quinn, Benjamin B. Bederson, “Human Computation: A Survey and Taxonomy of a Growing Field”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011. pp 1403-1412

[4] Laura Dabbish, “General Techniques for Designing Games with a Purpose”, Communications of the ACM, August 2008. pp 58-67.

[5] Luis Von Ahn, “Designing Games with a Purpose” Communications of the ACM, June 2006. pp 68-77.

[6]www.w3schools.com

[7]http://en.wikipedia.org/wiki/Luis_von_Ahn

