COMP 10071 Capstone Project

Final Project Report

# Developed By: Chao Zhang

# Student ID: 000306946

# Date Submitted: April 15th, 2016

Project Background

Many people enjoy casual gambling, but the money spent on gas along with time on the road could be put to better use. I myself love watching poker online, but my friends are not as enthusiastic as I am about it. So it is difficult for me to get any hands on experience of playing it at a beginner level without travelling hours to the casino. I envision project *Stacked Deck* to be a simple online gambling site for people to enjoy at the comfort of their homes. It will allow members to order credits to play Blackjack. Not everyone understands the rules of the game, that’s why my website will feature a guest mode where people can play without spending money. But of course they don’t get any winnings either. I want this project to be something my friends and I could use to play together some day, also to just show them what I’ve learned at school. Creating something with real world applications is what I want to achieve with my capstone project.

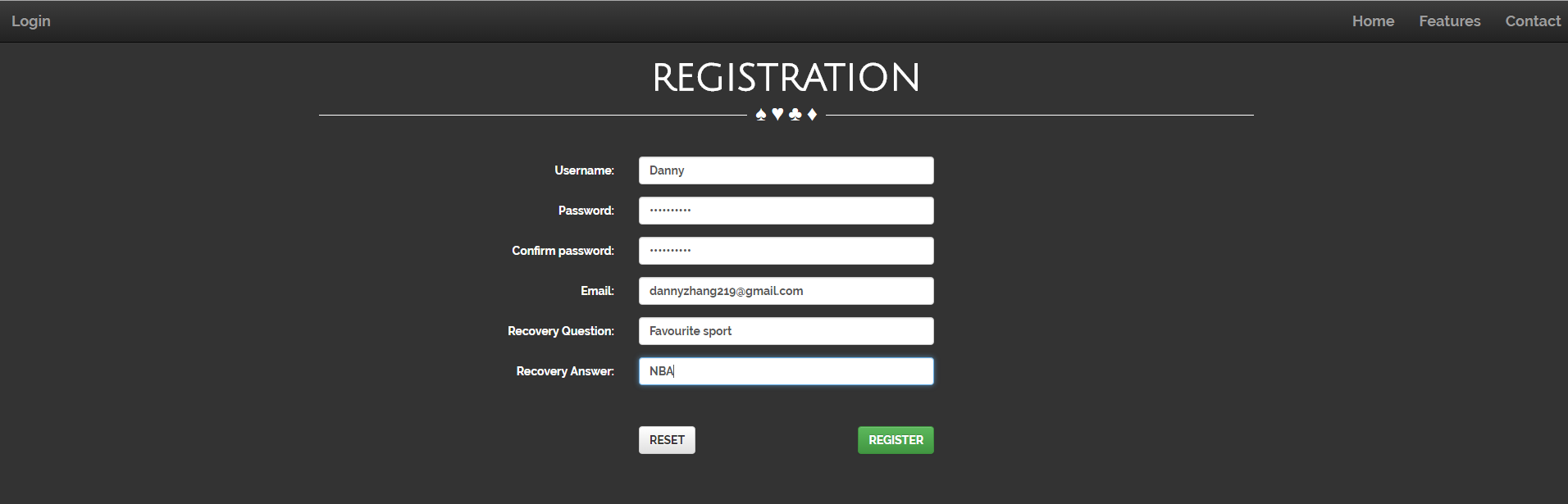
Functionality Summary

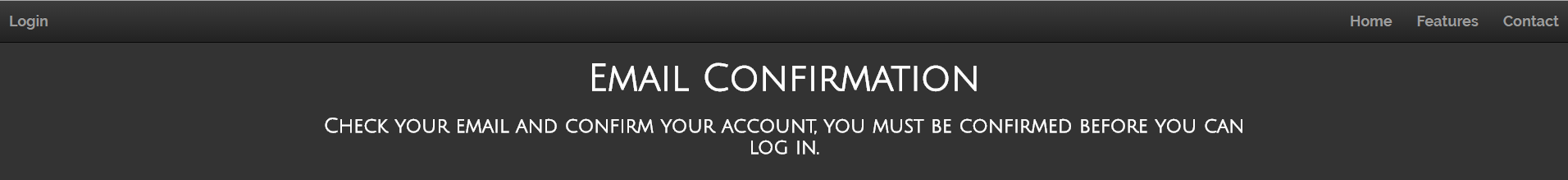
Project Stacked Deck

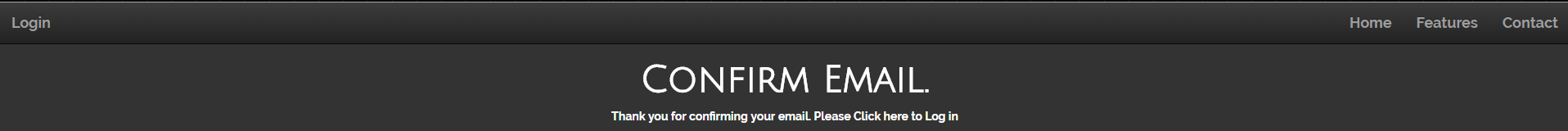
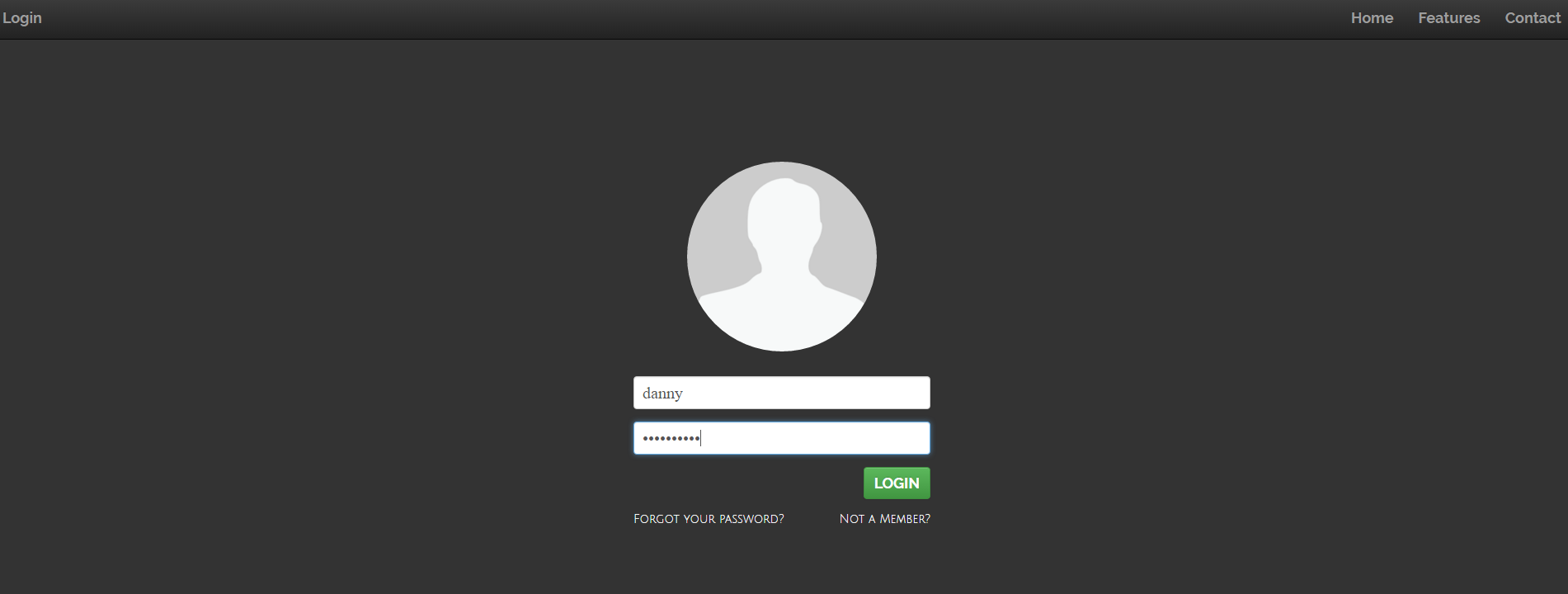
|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Admin | User | Anonymous |
| Register | ✓ | ✓ | ✓ |
| Login | ✓ | ✓ |  |
| Password Recovery | ✓ | ✓ |  |
| Credit Request | ✓ | ✓ |  |
| Messaging | ✓ | ✓ | ✓ |
| Discussion Threads | ✓ | ✓ |  |
| Discussion Posts | ✓ | ✓ |  |
| Search Discussion Threads | ✓ | ✓ |  |
| Report User Posts | ✓ | ✓ |  |
| Delete Posts | ✓ | ✓ |  |
| Delete Threads | ✓ | ✓ |  |
| Password Change | ✓ | ✓ |  |
| User Information Update | ✓ | ✓ |  |
| User Search | ✓ |  |  |
| Ad Management | ✓ |  |  |
| Credit Request Approvals | ✓ |  |  |
| User Post Report Management | ✓ |  |  |
| Play Game | ✓ | ✓ |  |

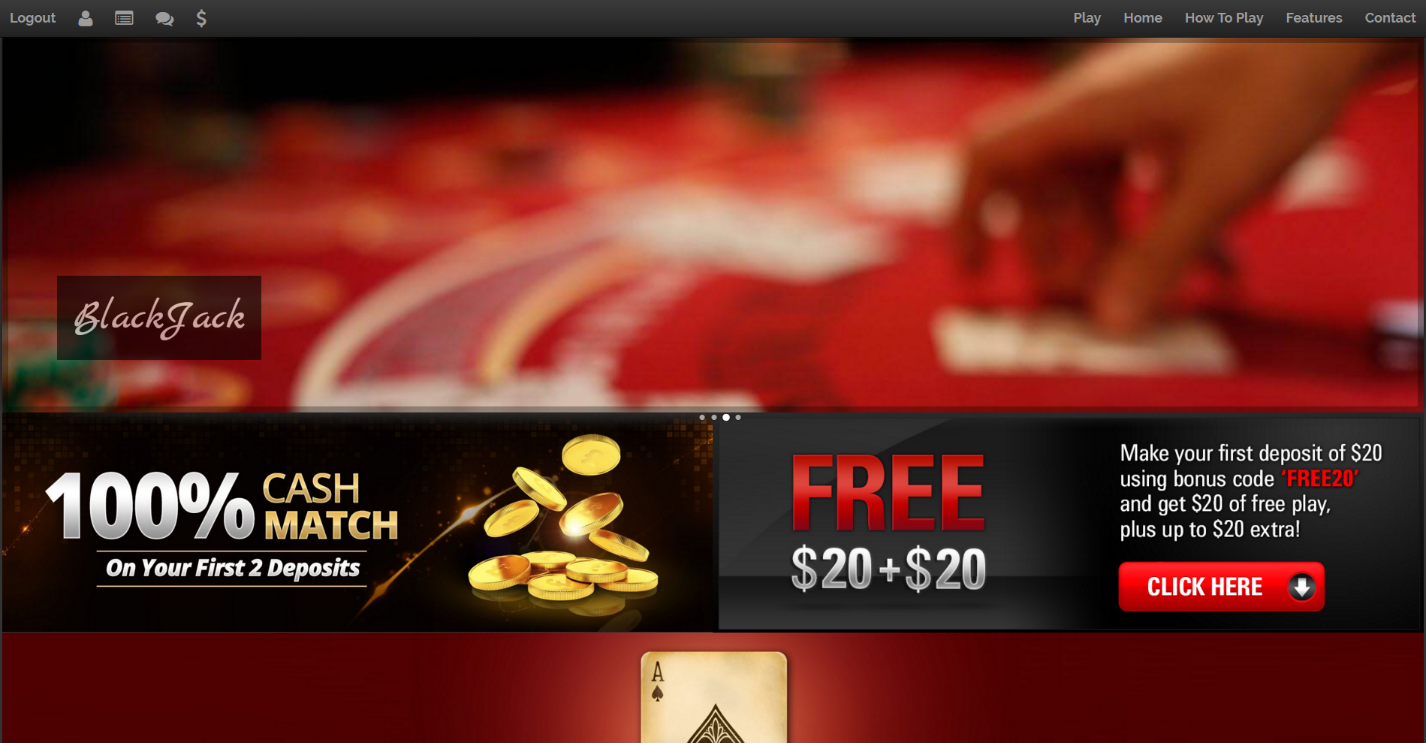
Major Functionalities

# Registration and Login



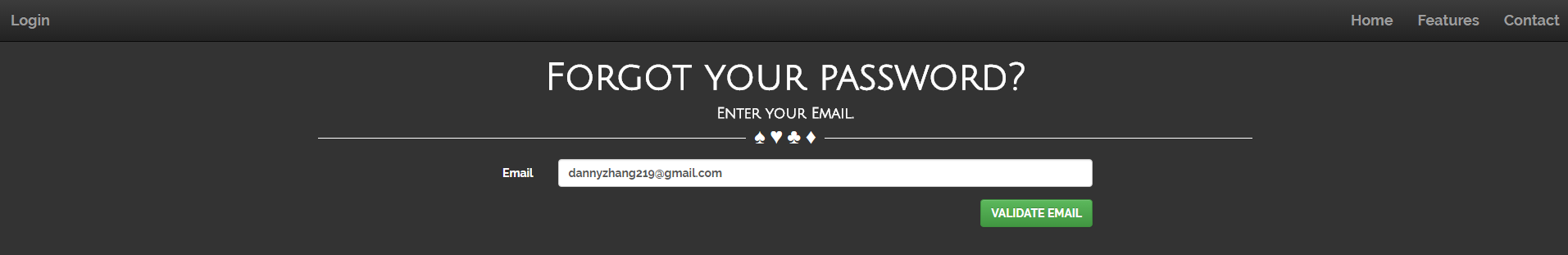


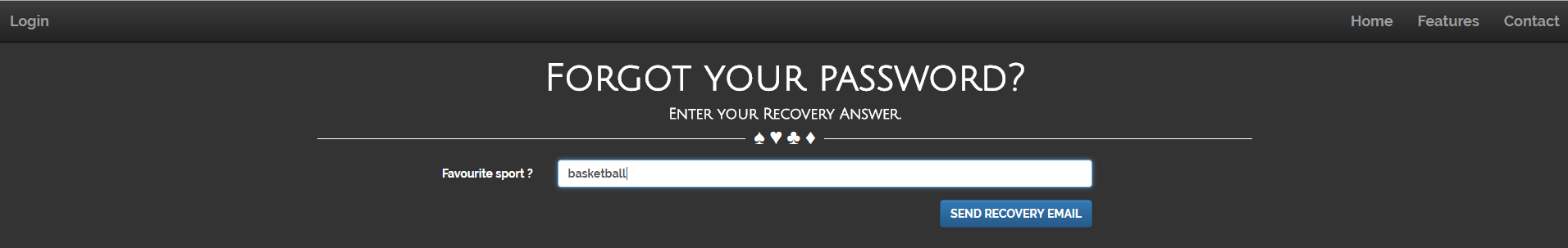


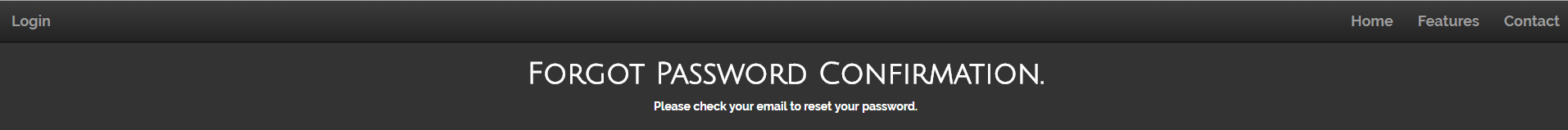


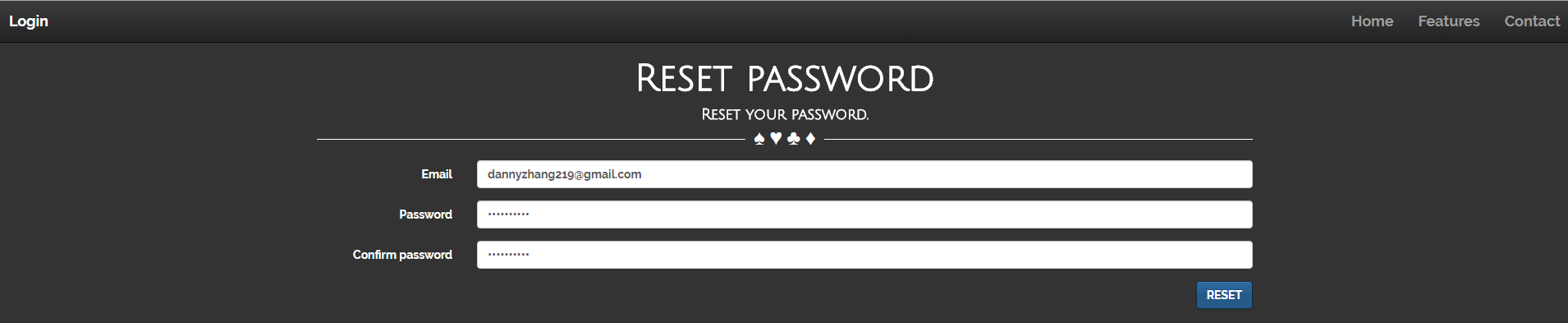
A user must go through the process of registration and log in to the application to use all of the functionalities. After filling out the required information on the registration page, the user must check their email for a confirmation email. After they have confirmed the registration, the user may then log in.

# Password Recovery





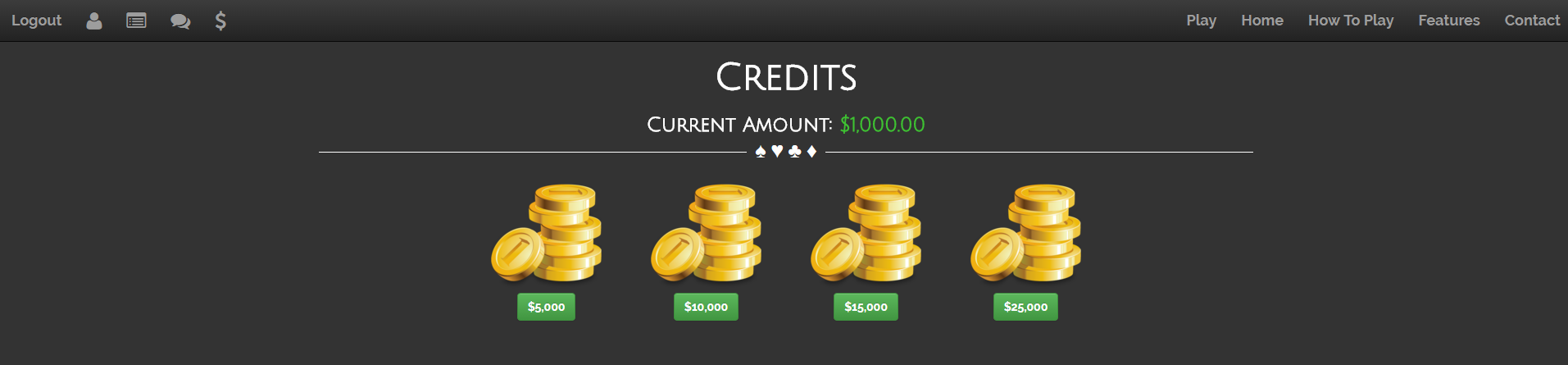






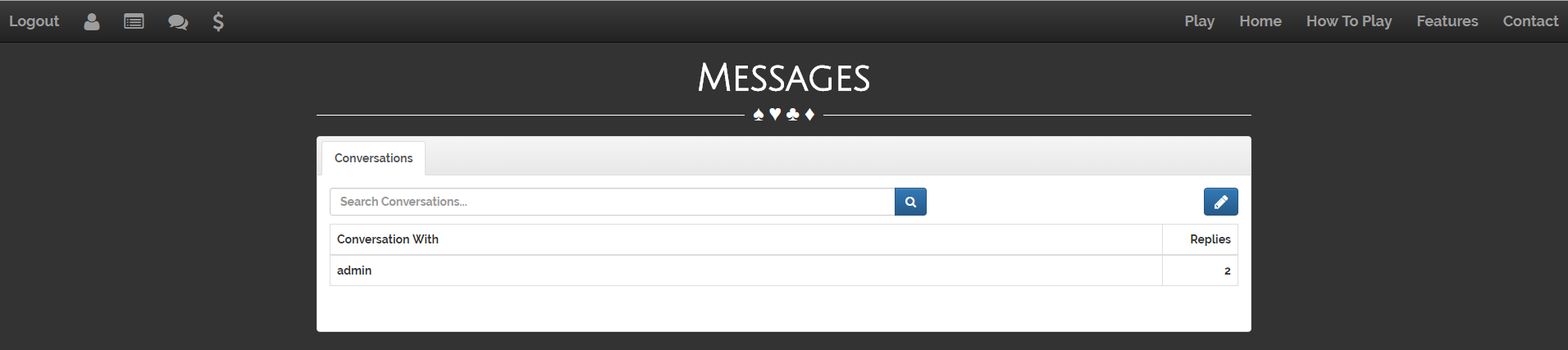
If a user forgets their password, they can recover their password through the password recovery process. After confirming the email address, the user must then confirm their recovery question. If the email address matches an existing address, a confirmation mail will be sent. After confirmation, user can then create a new password.

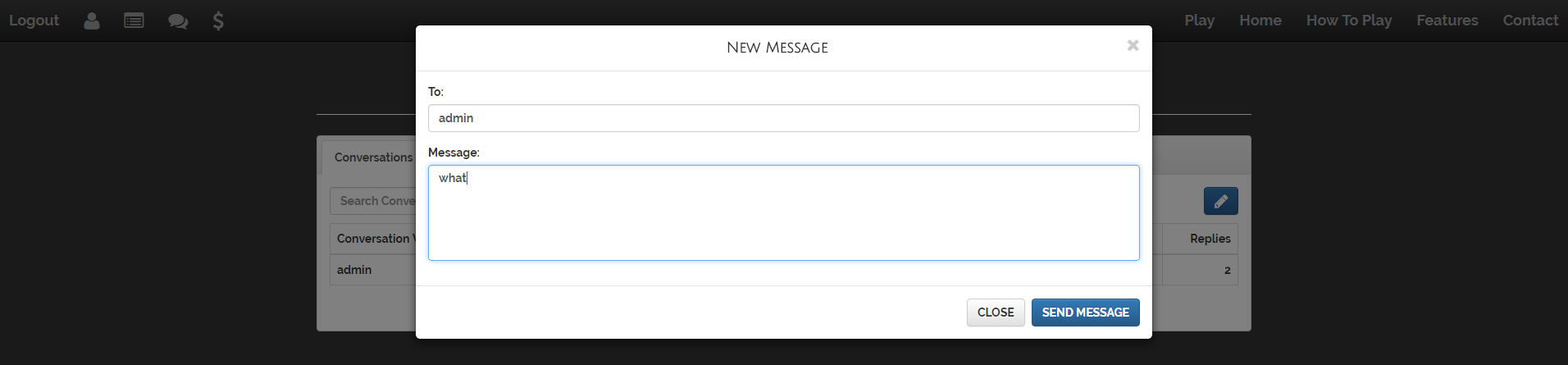
# Credit Request

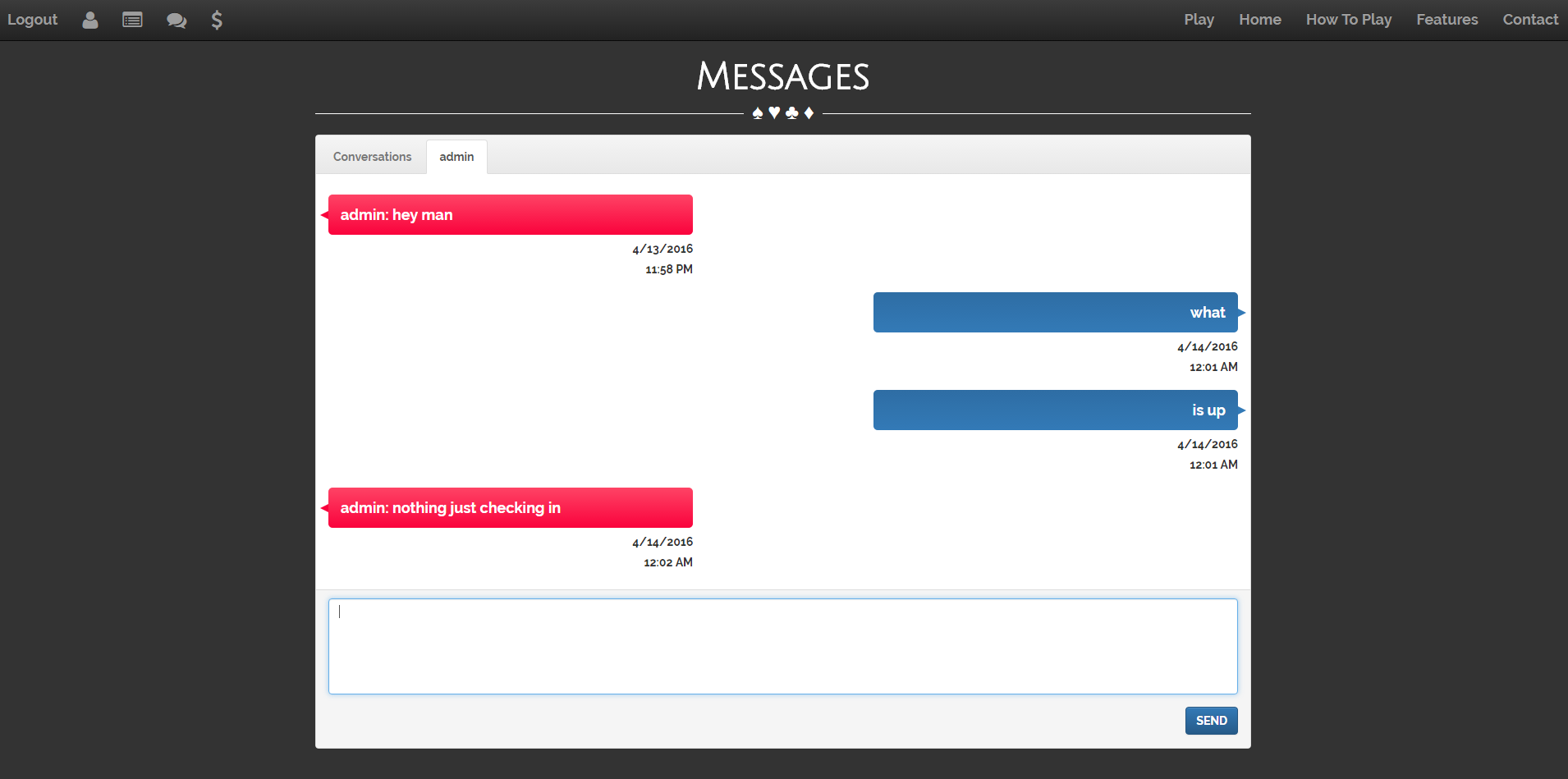


User can request any of 4 different amount of credits at any time for the admin to approve. Credits are used to play the featured game.

# Messaging

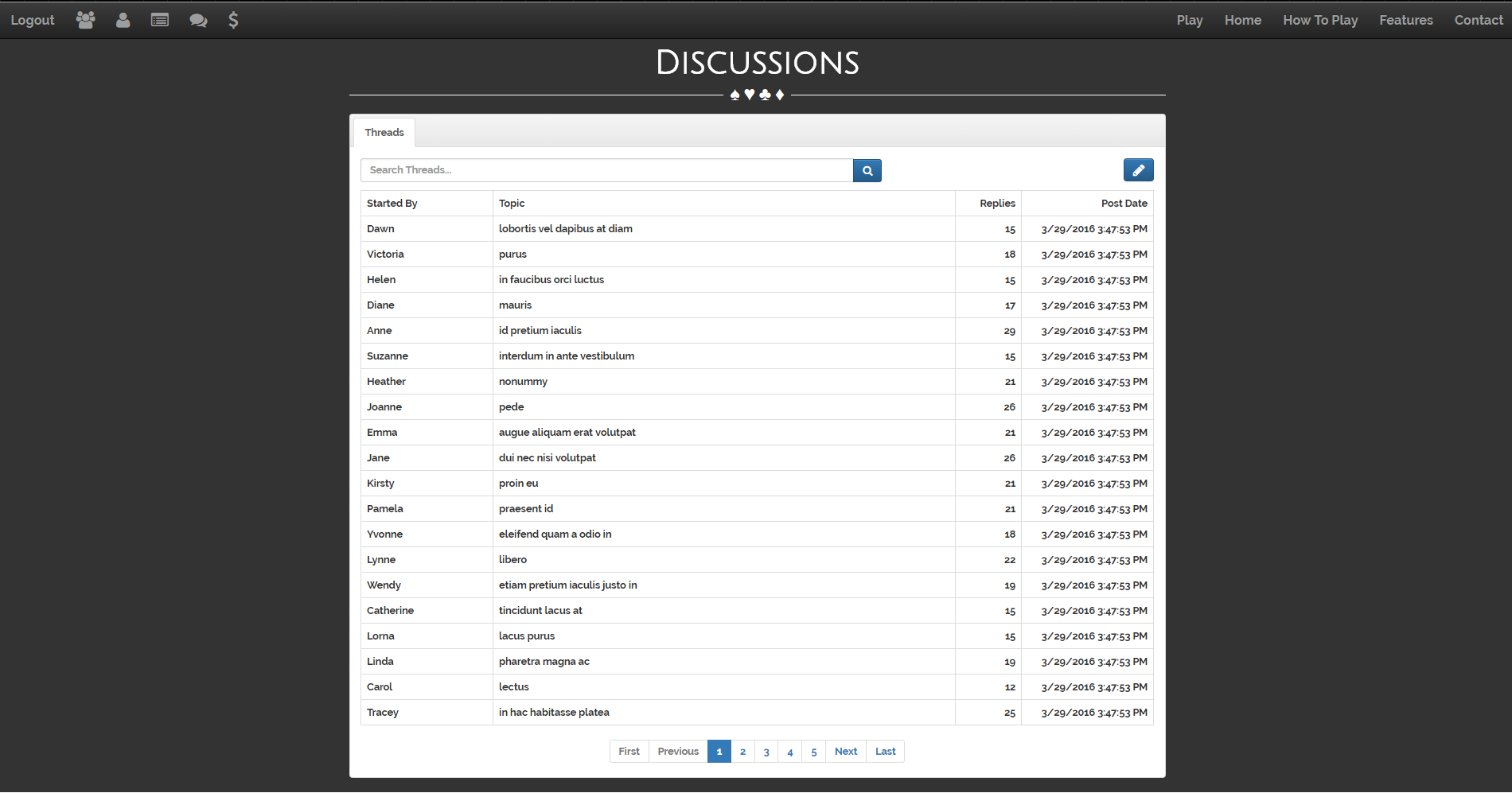


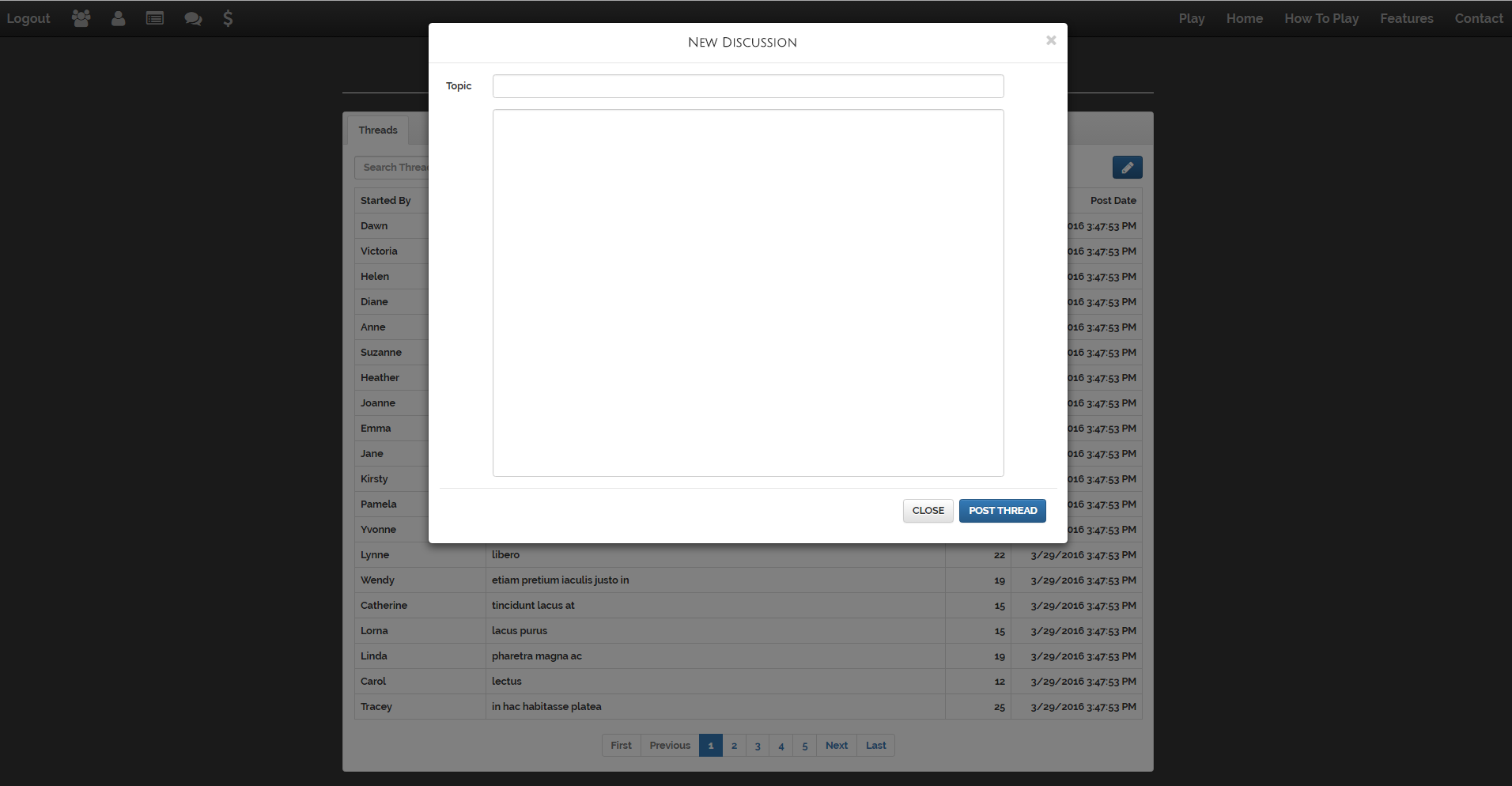


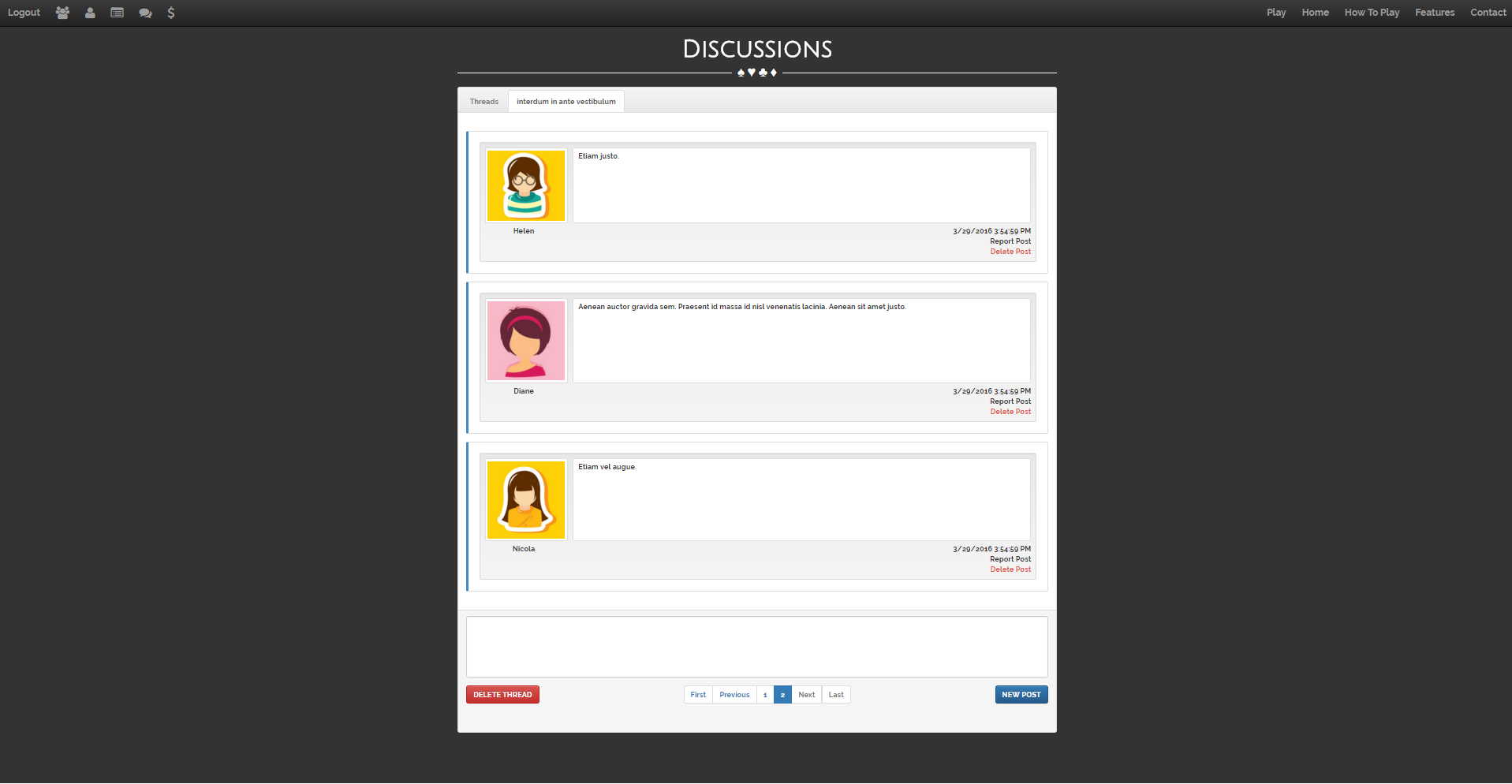


Users can send messages to each other in this page. There is a search conversation function to help if there are too many conversations.

# Discussions

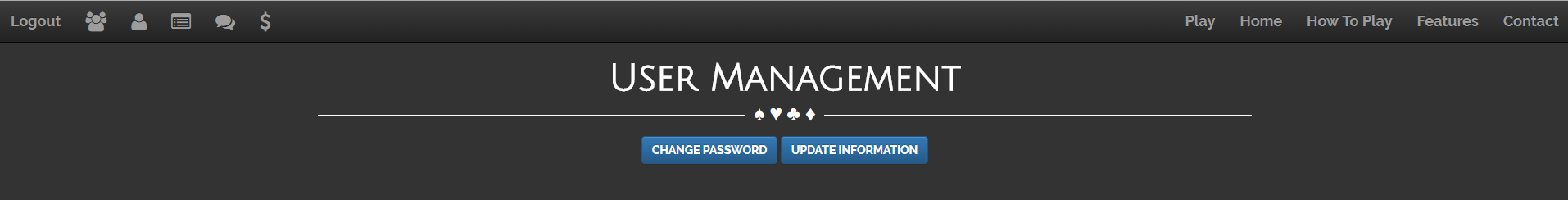






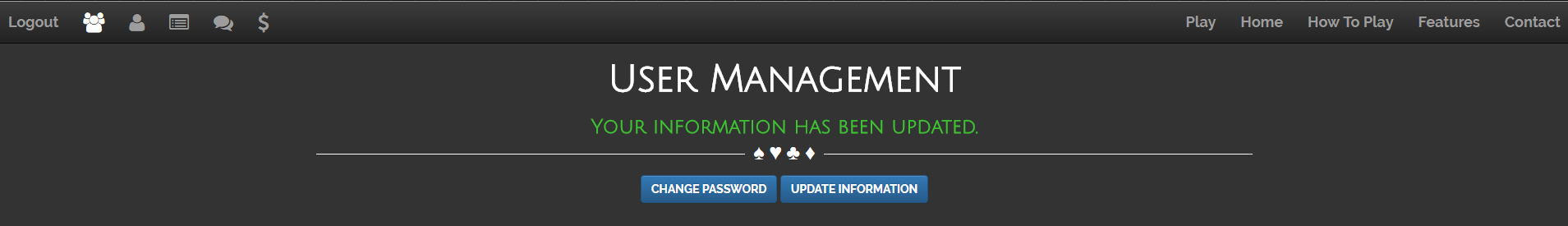
Any user can start a discussion thread, and can post comments into any thread. The threads and posts may only be deleted by admin or the author of the thread/post. There is a report user functionality associated with the post in the discussion thread. Any reported users will show up in the admin page.

# User Management





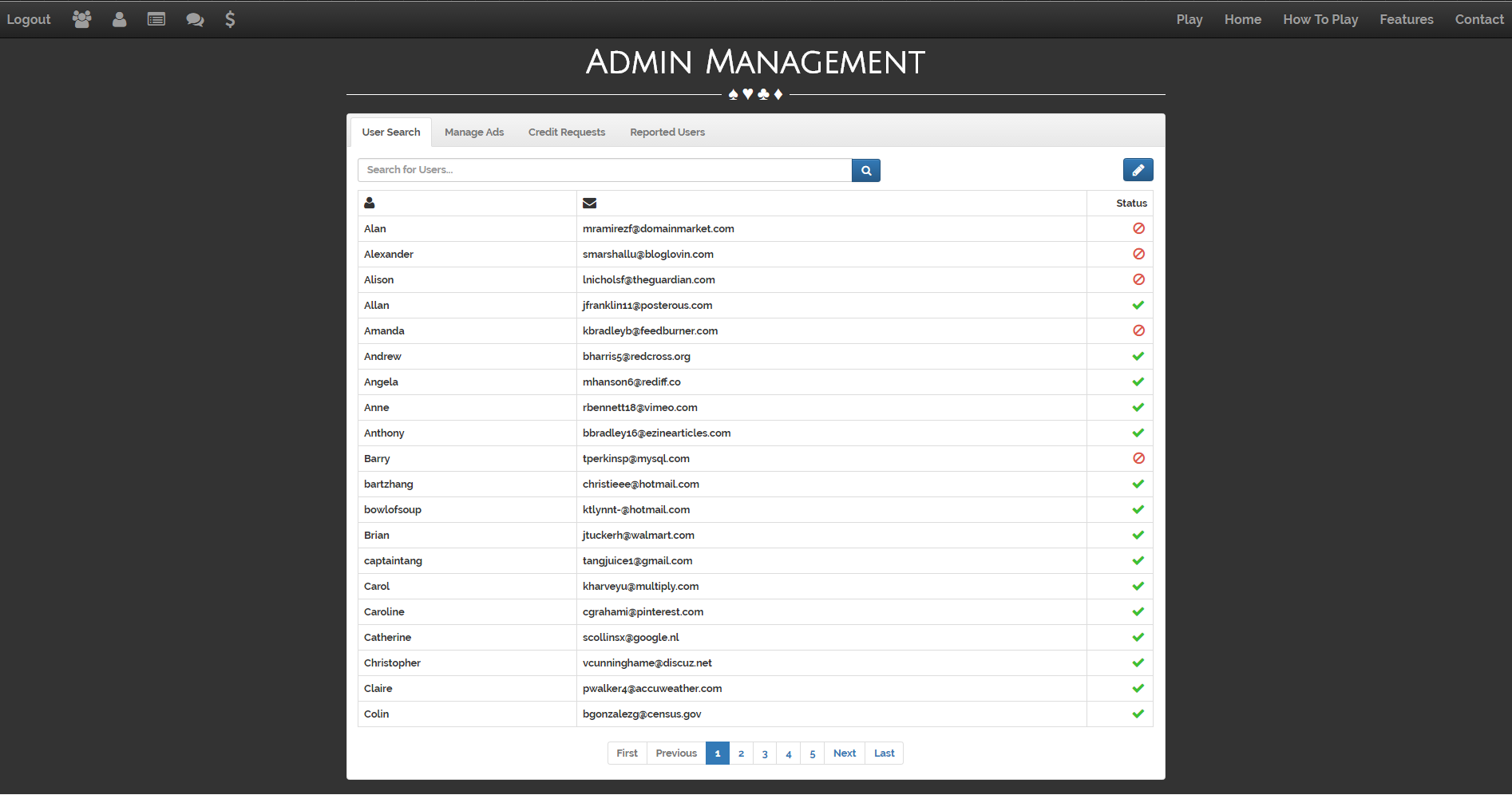


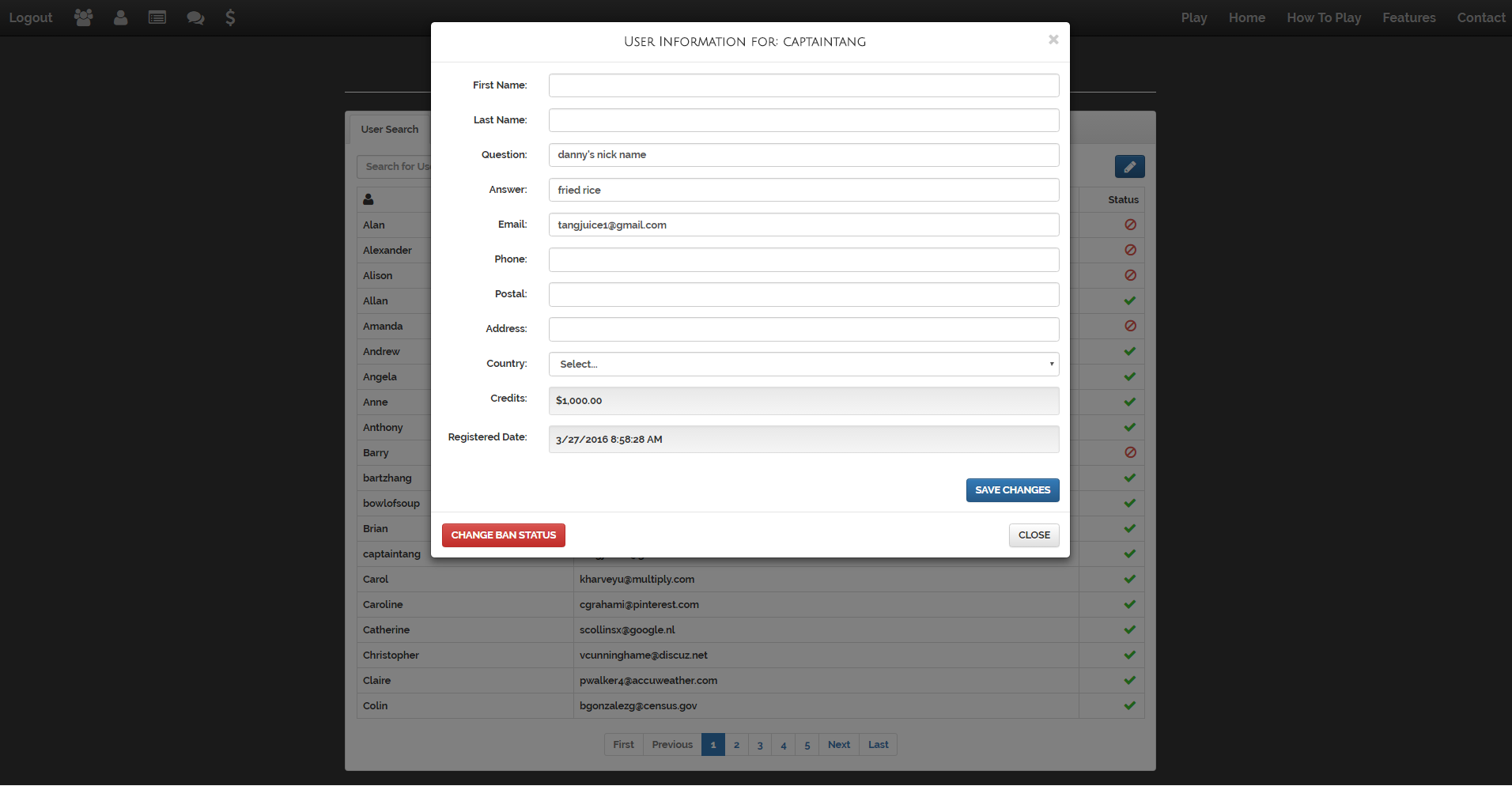


User can enter extra information about themselves, change password and select an avatar for display.

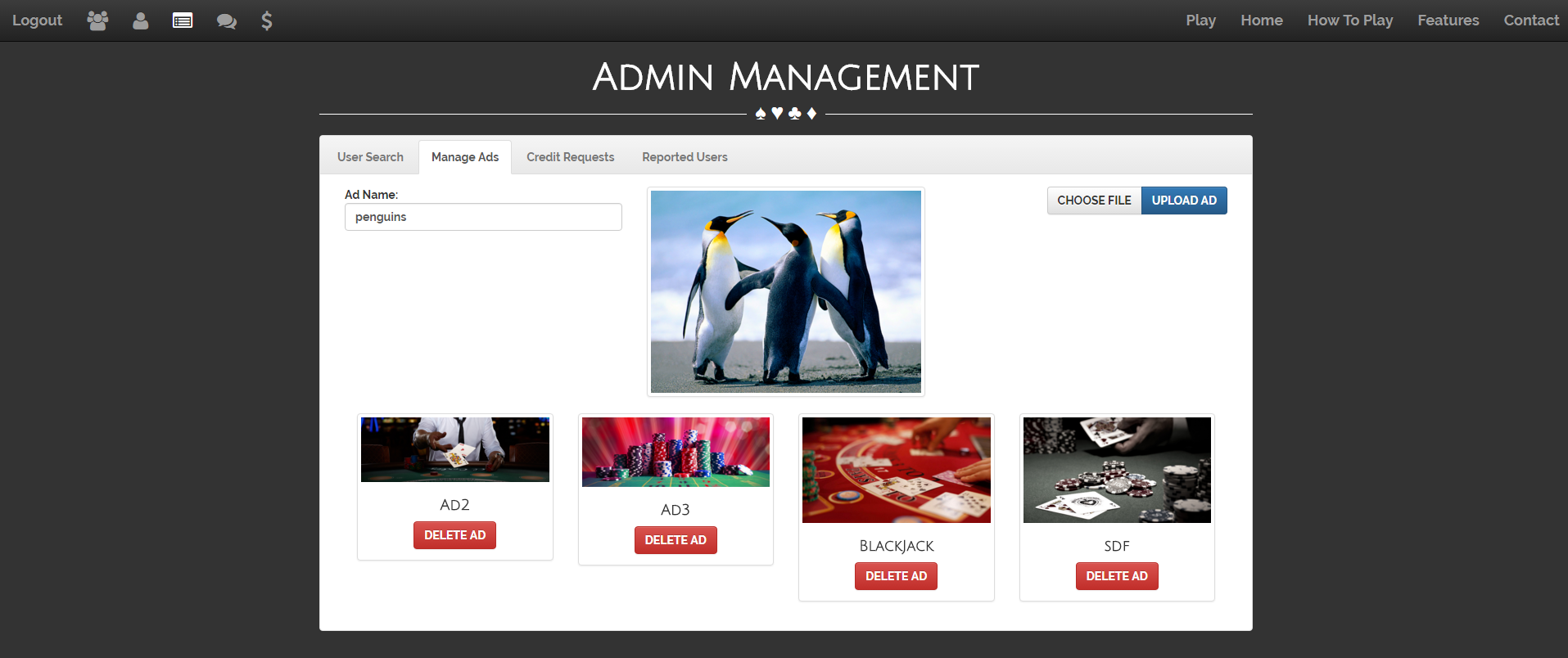
# Admin Management

Admin management consists of the application’s core functionality. The admin page has four major functionalities: User management, advertisement management, process credit requests and report management.

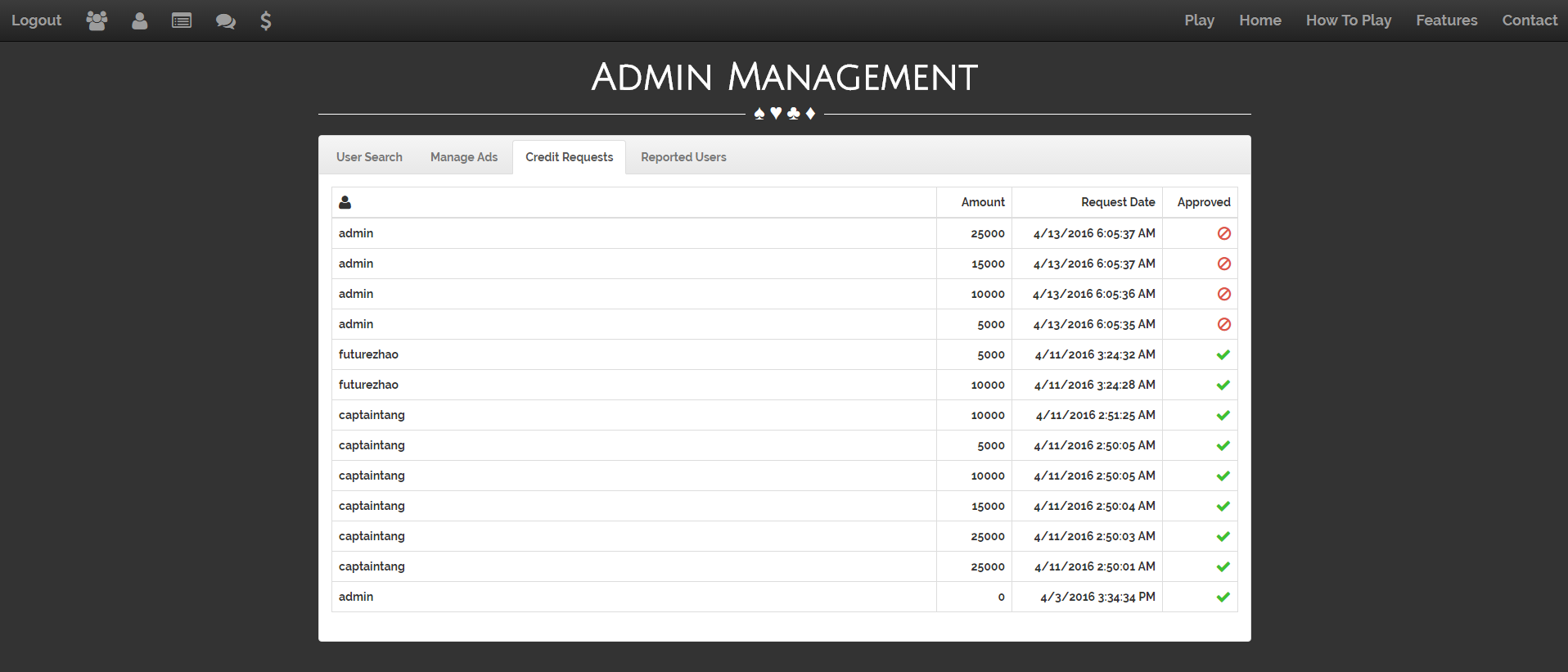




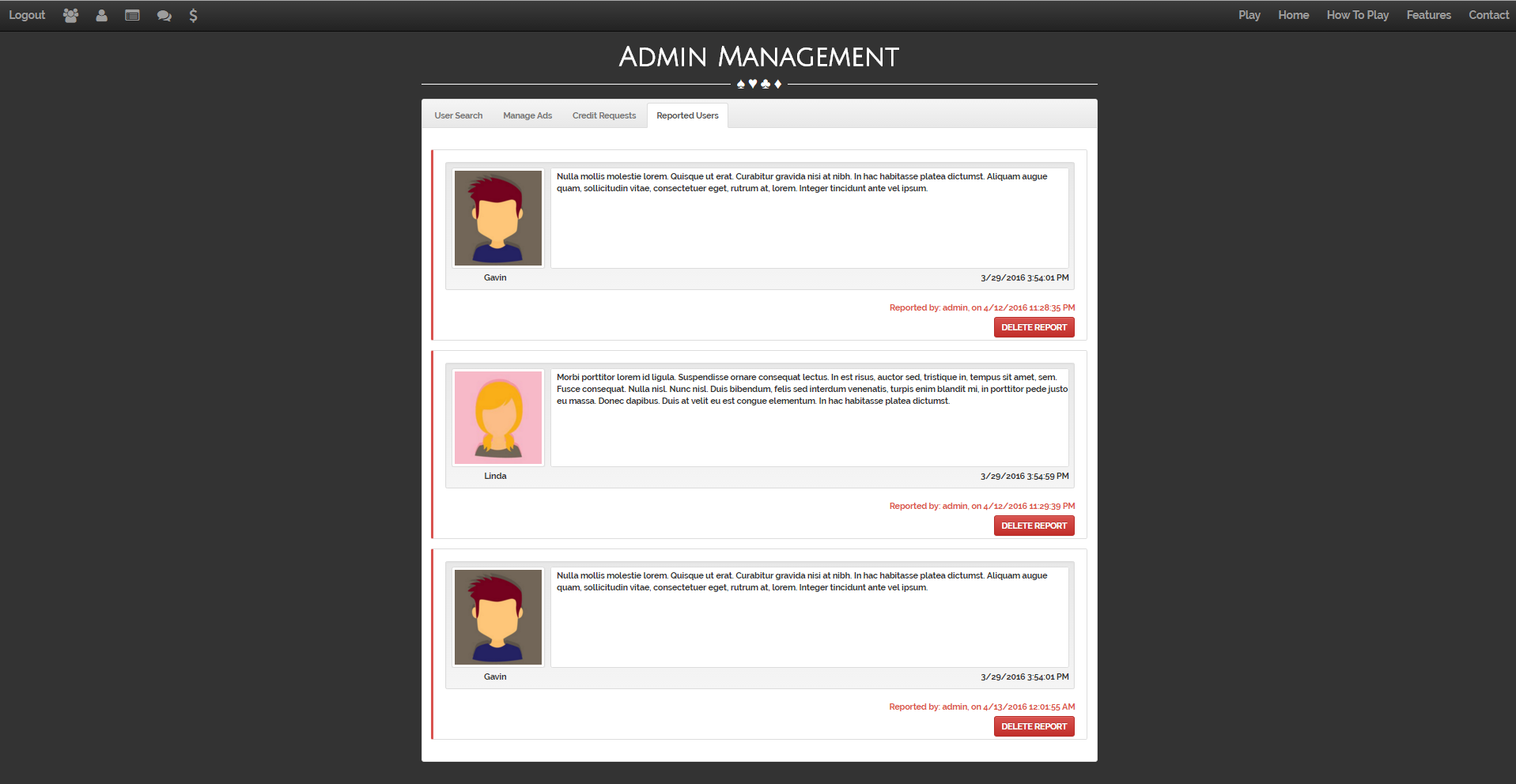
In user management, the admin can search any user based on username or email address. When a user row is clicked, the information associated with the user is brought up. The admin may edit some of the information fields. The admin can also toggle the ban status of a selected user from this page.



In the manage advertisements section, the admin can upload/delete images on the server. The images are used on the home page as a carousel.



By clicking a credit request, the admin approves the amount of credits requested by a user. The amount of credits will be added to the user’s account.



In the reported user section, the admin can review all the posts that have been reported by users.

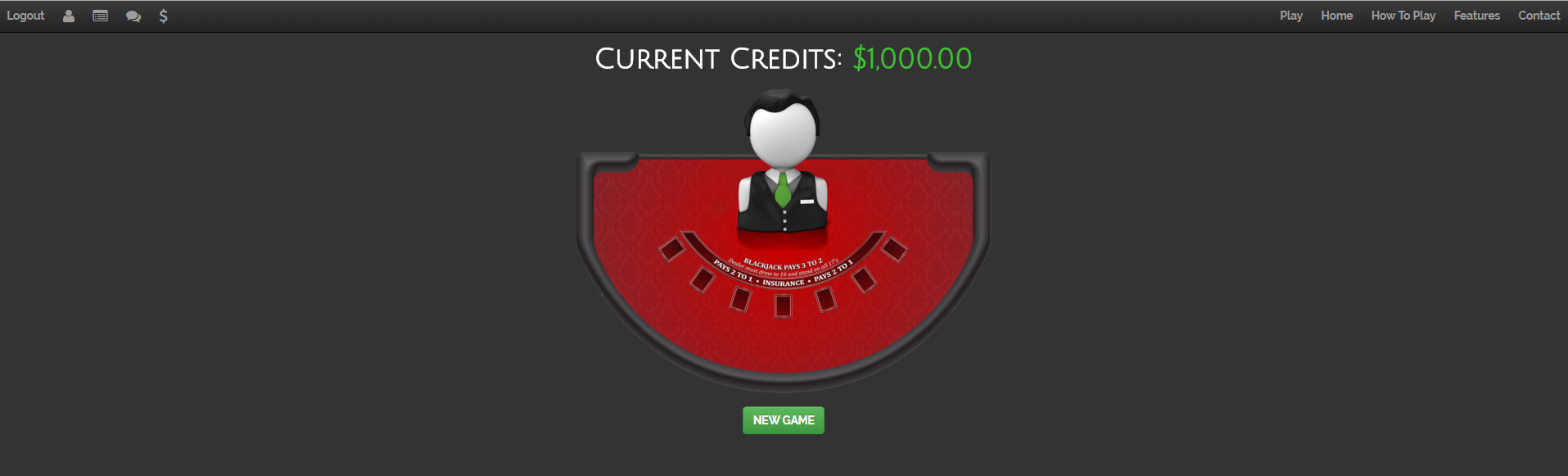
# How to Play

This page has two embedded YouTube videos on Black Jack tutorials, and rules of the game in the description.

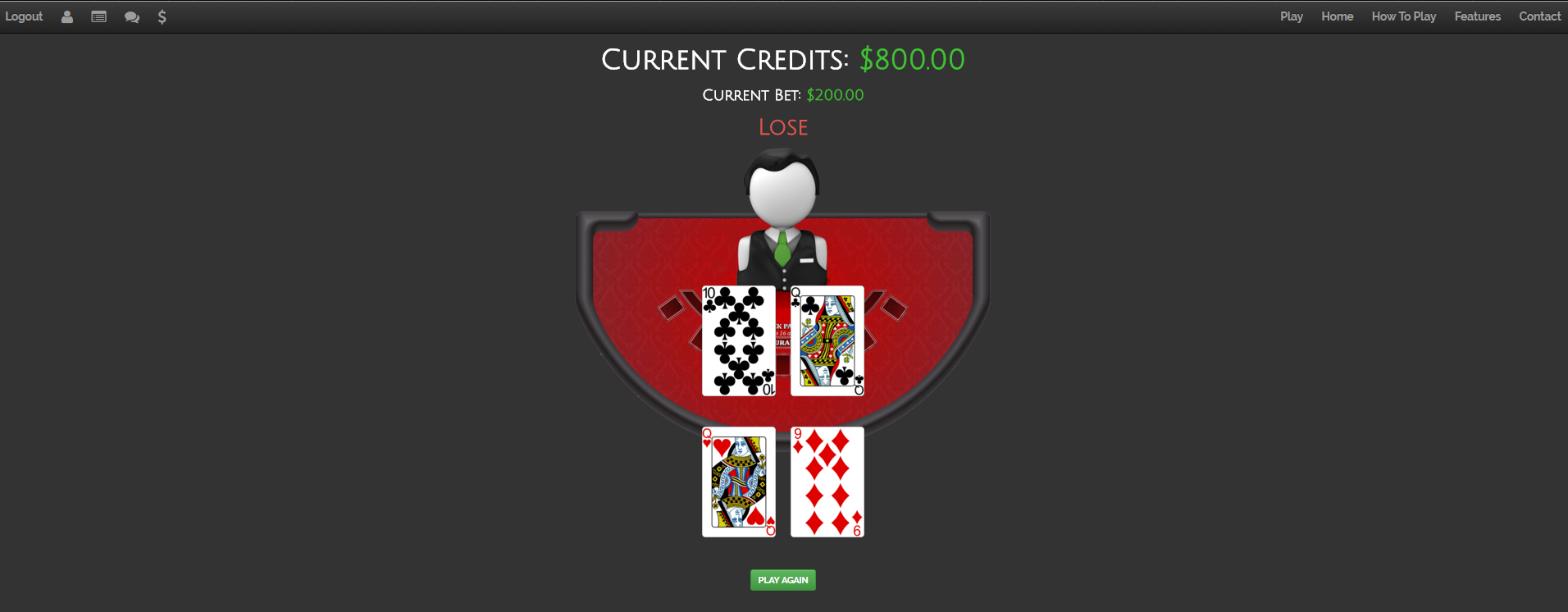


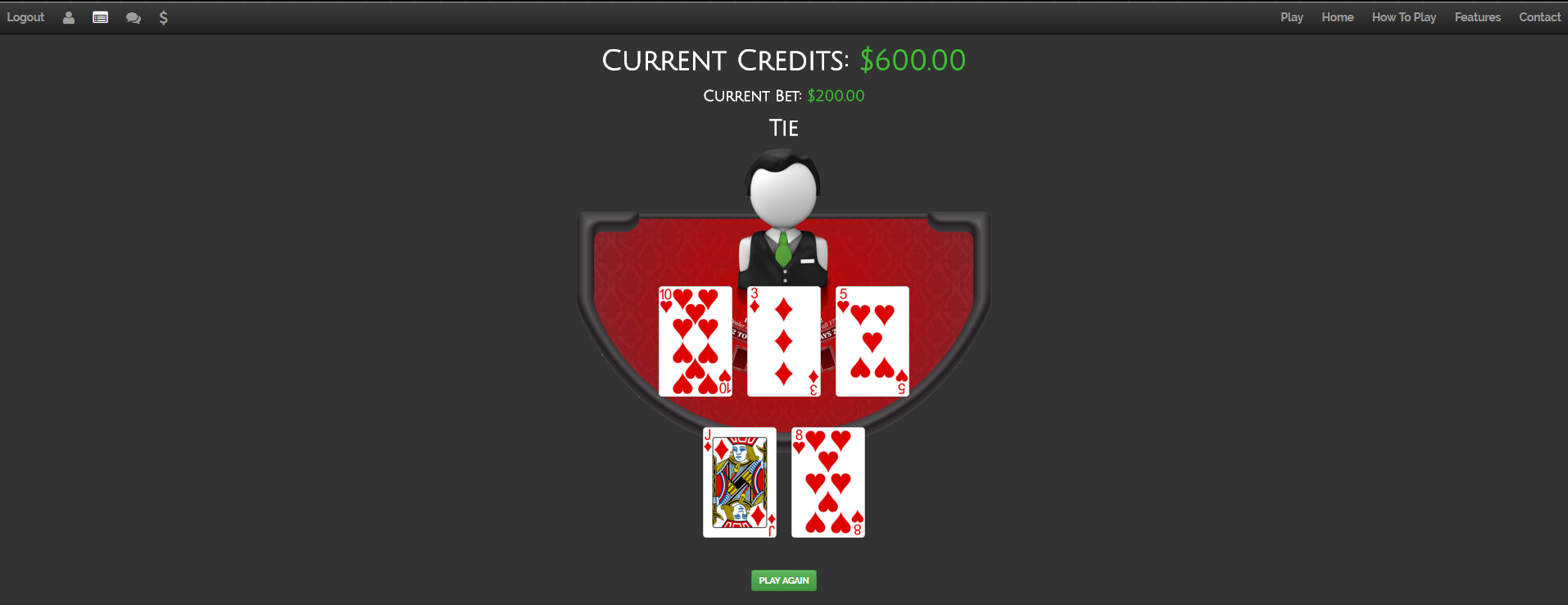
# Play Game

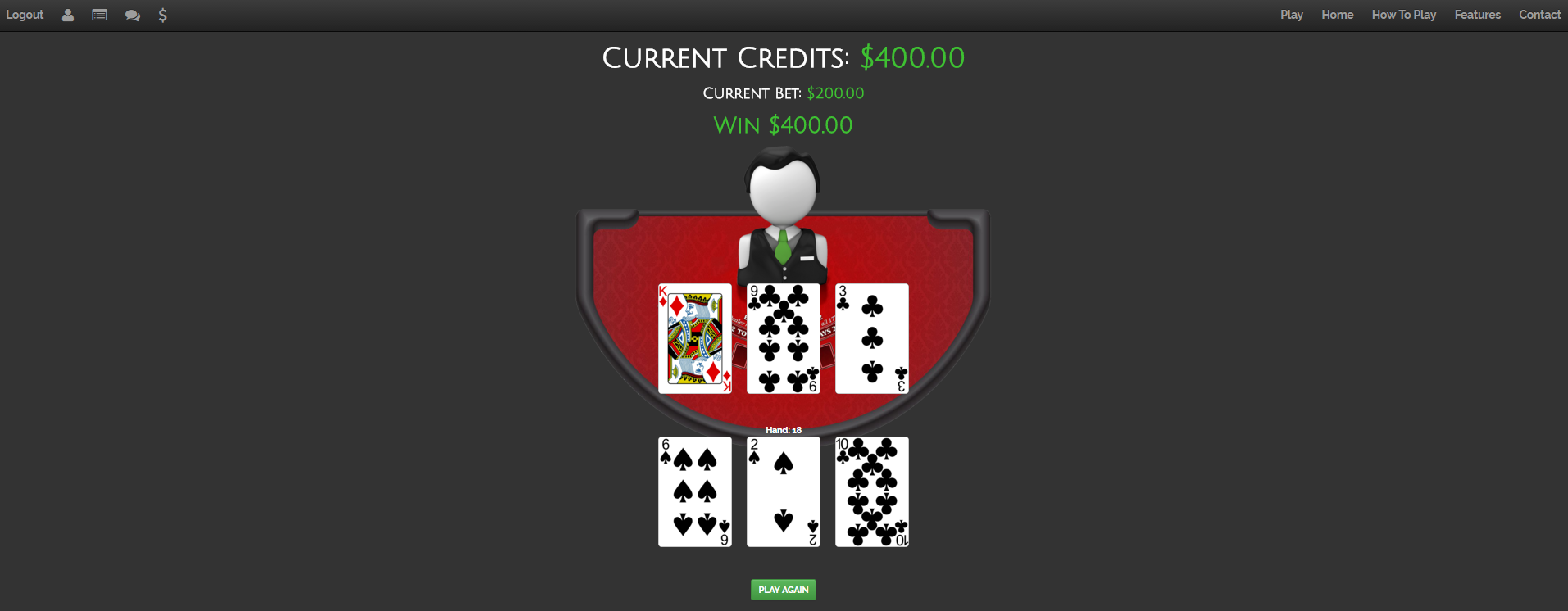
The featured game here is Black Jack, the user may bet any amount of credits that they own. They will win double the bet amount, and lose the current bet amount. If the user runs out of credits, they will be redirected to the credit request page. There is no access to this page without credits.











Development Stack, Framework, Plug-ins

|  |  |  |
| --- | --- | --- |
| Technologies used:   * HTML5 * CSS3 * Twitter Bootstrap * JavaScript * C# * Microsoft SQL Server * AspNet Razor Dynamic webpages | Frameworks/Libraries used:   * Microsoft AspNet Identity Entity Framework * Code First Migrations and Deployment with the Entity Framework * JQuery * Modernizr | Plug-ins/Services   * Jasny Bootstrap * WOWSlider Carousel * SendGrid Emailing Service |

Code Notes

The two main frameworks I used was provided by Microsoft AspNet. Their Identity Entity Framework and Code First Migrations Deployment with Entity Framework. I estimate about 70% of the entire project is my own original code. I also used libraries, plug-ins and services for this project. The AspNet Identity Entity Framework is used for user authentication and creation. I built upon the original boilerplate code to customize my application’s user fields. The database migration entity framework is used for updating database automatically if the schema was changed in the classes. The emailing service was provided by the SendGrid Emailing Service, a suggested provider from Microsoft AspNet.

Extensions

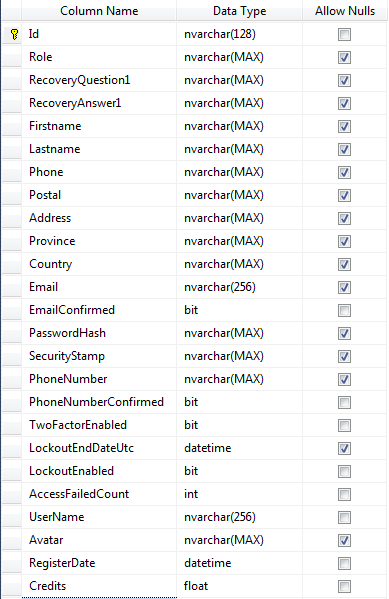
In the future, I would like to clean up the typography, colour scheme and images in my project. By doing this I am hoping to make it look more professional and less amateurish. This project is a good indication of my capabilities as a new developer that just graduated college. So I would like to use it for reference when I apply for jobs.

Installation

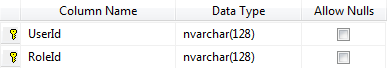
Extract the zipped visual studio 2015 web application project, open it in visual studio and run.

# Database Schema

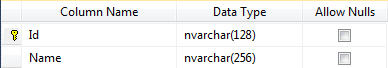
## dbo.AspNetUsers



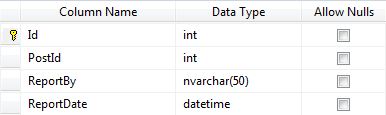
## dbo.AspNetUserRoles



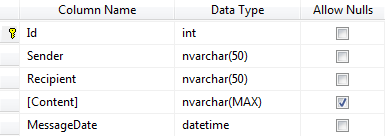
## dbo.AspNetRoles



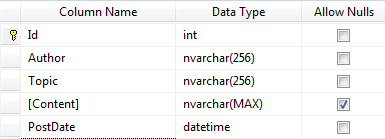
## dbo. AspNetReportedUsers



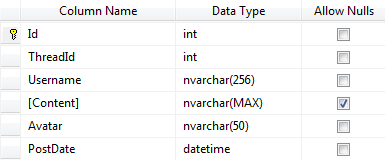
## dbo. AspNetMessages



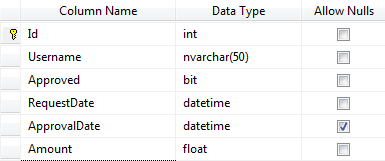
## dbo. AspNetDiscussionThreads



## dbo. AspNetDiscussionPosts



## dbo.AspNetCreditRequests



## dbo. \_\_MigrationHistory

This is used by the MVC5 data migration entity framework in visual studio to automatically update database.

