EzGift

A Easy Gift Finder Appliction for all

Introduction

EzGift is a web application build in purpose of easily managing gifts you want from others as well as buying gifts for your dear ones. You can keep track of your own wish list in your mobile as well as let others know about it, so that they don’t have to do the guessing and often repeating the same gift.

The following sections will cover the design details of the application.

1. Design
   1. Architecture Diagram
   2. Important Classes/Objects
2. Features
   1. My Wish List
   2. Friend List
   3. Events
   4. Buy Gift
   5. Personal Profile
3. Web Services
   1. Own Services
   2. Third-party services
4. Database
   1. Important tables/relations
   2. Data access methods
5. Mobile User Interface
   1. Requirement
   2. Screen shots
6. Limitations
7. Future Works
8. Reference

## 1. Design

The following architecture diagrams can best describe the total architecture of the system:

[diagram]

## 2. Important Classes/Objects

| Class | Responsibility |
| --- | --- |
| IService1 | The Service Interface class contains all the services provided by the system. All services return JSON formatted data. The request methods are GET and POST mainly. |
| Service1.svc.cs | The implementation class of the IService1 interface. |
| DataHandle.cs | All the data access/SQL methods are defined in this class. |
| UIDFactory.cs | A Utility class responsible to generate random numbers which can be used as user session tokens. |
| Package: ResponseClasses | List of all the Response classes that are used as container and serialized before sending as JSON response. |
| Package: DAO | Data Access Objects used as data structures. |
| Web.cofig | Configuration file for the project. |

## 3. Features

**My Wish List**

The Wish list contains all the items a user wants. The list contains the Items description, Barcode # etc. A user can add a new item in the list as well as remove an item from the list.

**Friend List**

The friend list contains all the friends of the user. S/he can add/remove friend. While adding the friend, a notification should go to the requested person (not implemented yet) and when he accepts it, they become friends. The friends can see the user’s wish list.

**Events** (not implemented yet)

A user can create events like birthday etc. Then he can add his friends to the events. A user would have liberty to choose guests for the event among his friends. The guests will be notified via message.

The Event page should contain event data, place, status etc.

A user can see forth coming events, this includes events he created as well as the events he is invited. A user can accept/deny the invitation. If it is his event, he may cancel or reschedule it. In both case the guests will be notified.

**Buy Gifts** (not implemented yet)

The user can buy gift online and request to deliver it specified place. For example, a guest can buy a wish list item and request the delivery should be made in hosts’ house.

**Personal Profile** (partially done)

A user can update his personal information (e.g. address, phone etc.) at any time.

## 4. Web Services

The following web services are implemented so far:

|  |  |  |
| --- | --- | --- |
| Method Name | Return Type | Description |
| login(string username, string password) | LoginResponse | Login user |
| register(string username, string password, string email) | RegisterUserResponse | Create account |
| addfriend(string userid, string fid) | AddFriendResponse | Add a friend. |
| acceptReqeust(string userid, string reqid) | AcceptRequest | Accept friend request |
| [POST] listFriends (string userid) | ListFriendsResponse | List of friends |
| [POST] removefriend(string userid, string FriendId) | RemFriendResponse | Remove a friend from list |
| createEvent(string userid, string eventName, string place, string type, string startTime, string endTime) | CreateEventResponse | Create an event |
| inviteFriends(string userid, string eventid, string friends) | InviteFriendsResponse | Invite a friend to event |
| searchUser(string q) | SearchUserResponse | Search a user by keyword |
| wishlist(string userid) | GetWishListResponse | Get wish list |
|  |  |  |

**Third-party Web service**

The UPC database[http://www.upcdatabase.com] web service has been used for collecting product information from the online repository using barcode #. The provider allow remote procedure call to its server and the return data in XML format. A XML-RPC client project (RPCClient1) is a added in github source code folder which does this job.

## Database Design

The following tables are used in the application so far:

|  |  |
| --- | --- |
| Table Name | Attributes |
| userInfo | * userid * username * password * fname * lname * dob * phone * email * address |
| Friendlist | * userid * fid * status |
| eventBasic | * eventid * ename * etype * hostid * place * startTime * endTime * event-status |
| eventGuestList | * eventide * guestid * status |
| Wishlist | * userid * ucd * description |

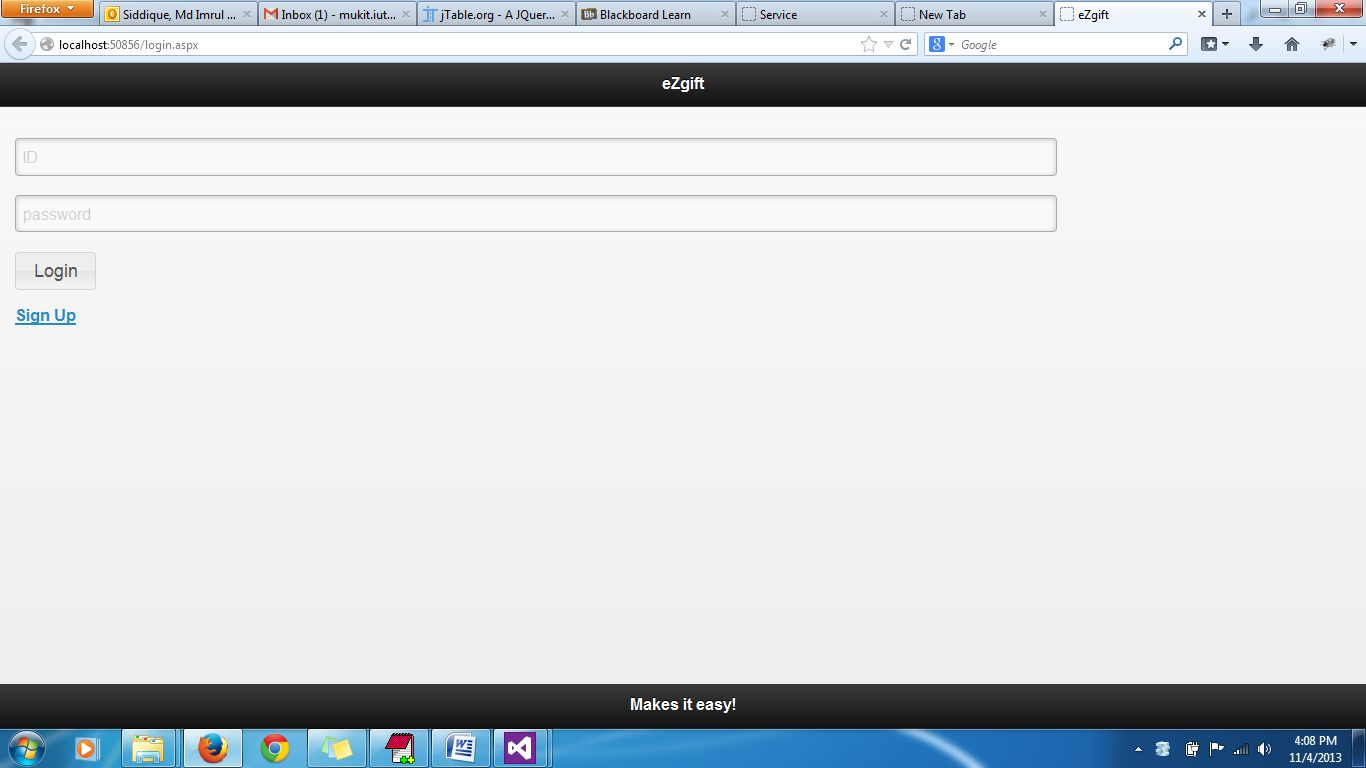
## Mobile User Interface

Requirement:

1. Jquery,
2. Jquery UI,
3. JTable.

## Screenshot

Login Page:



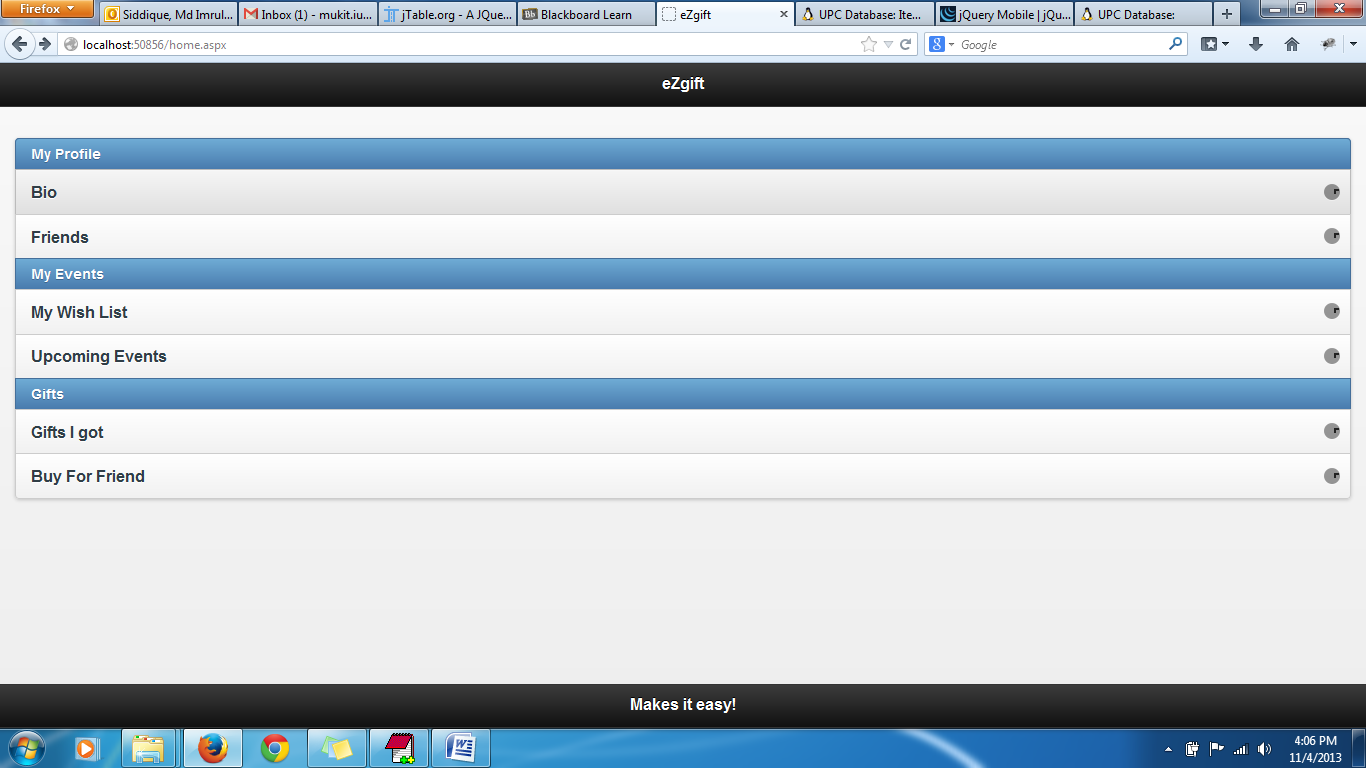
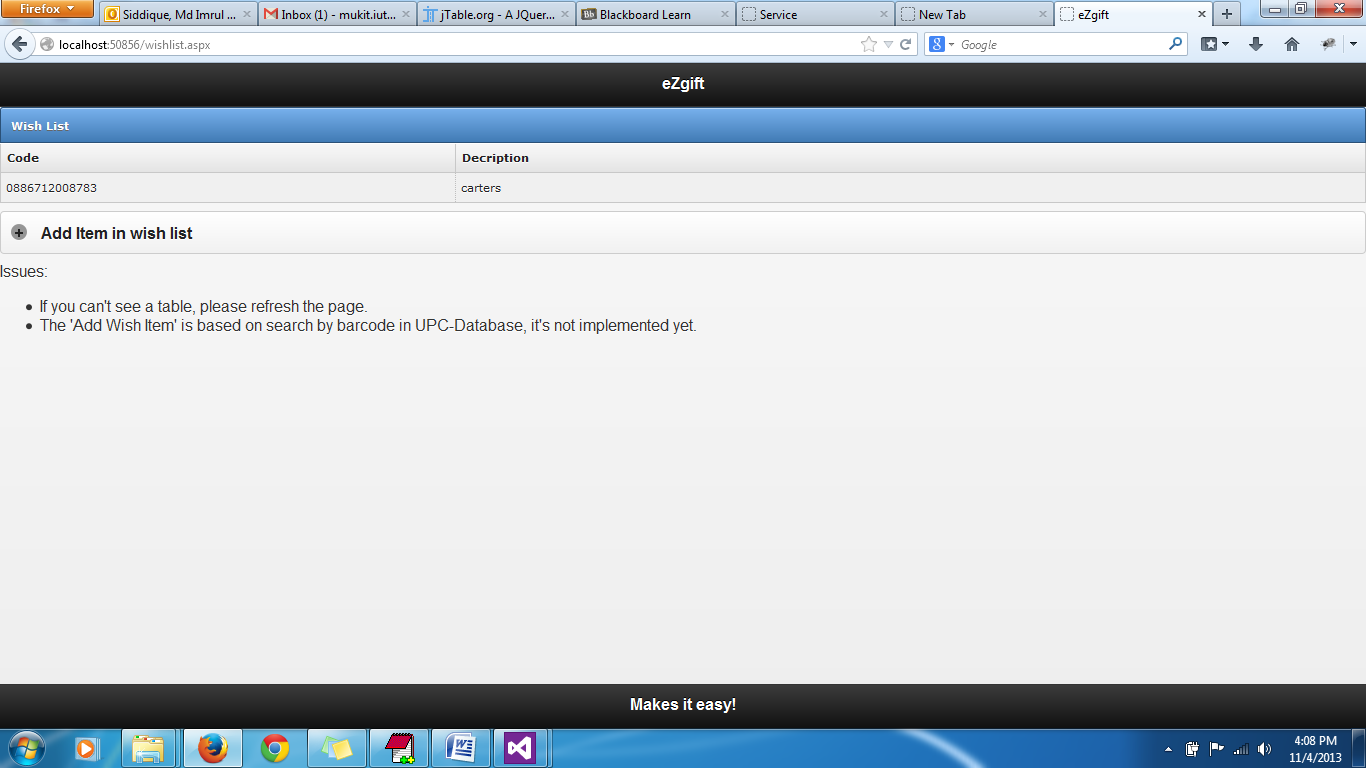
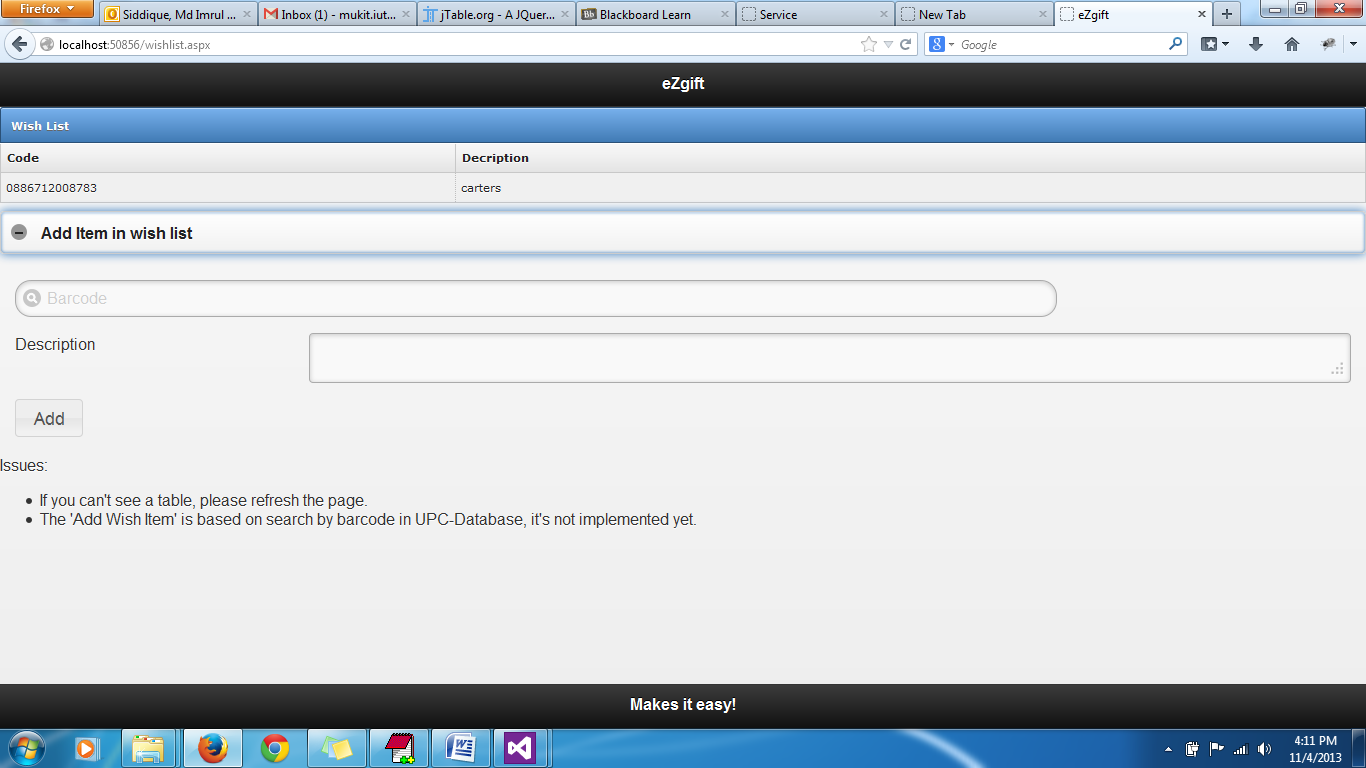
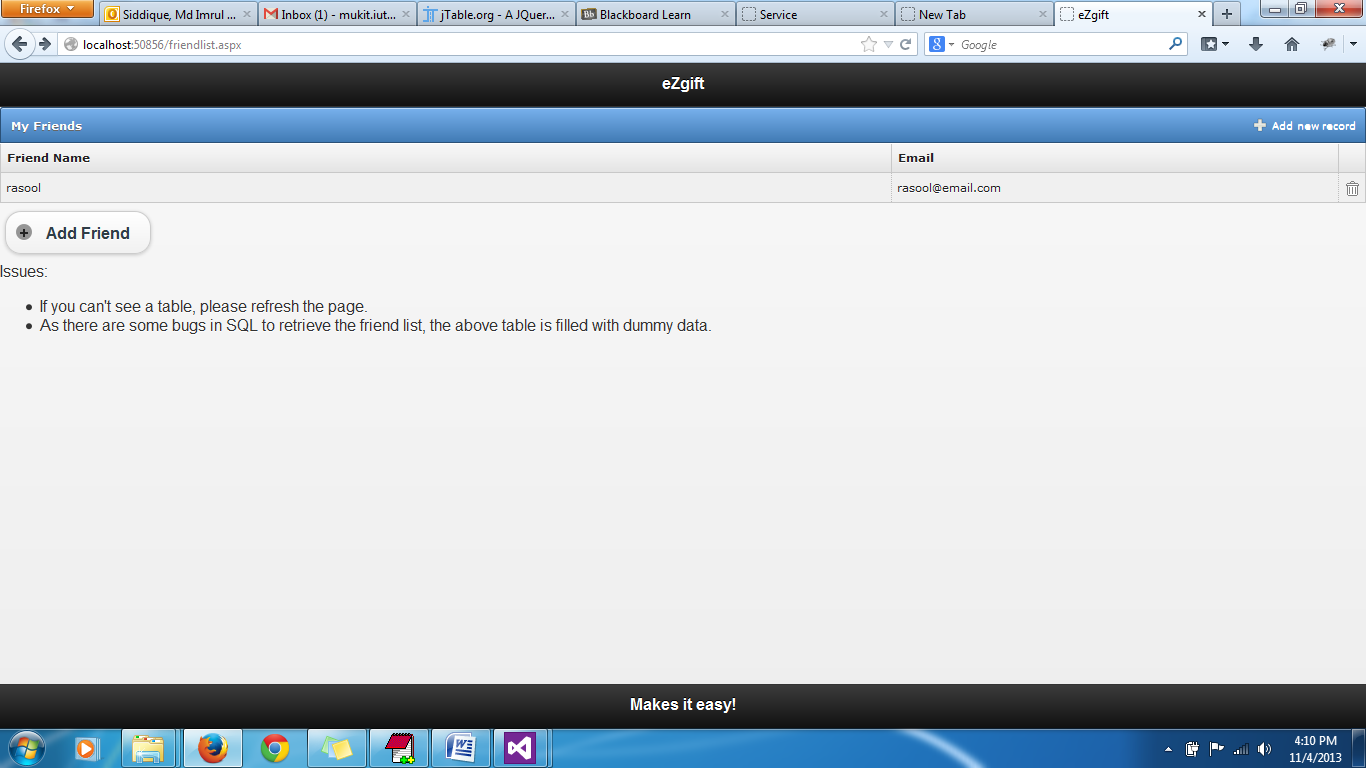
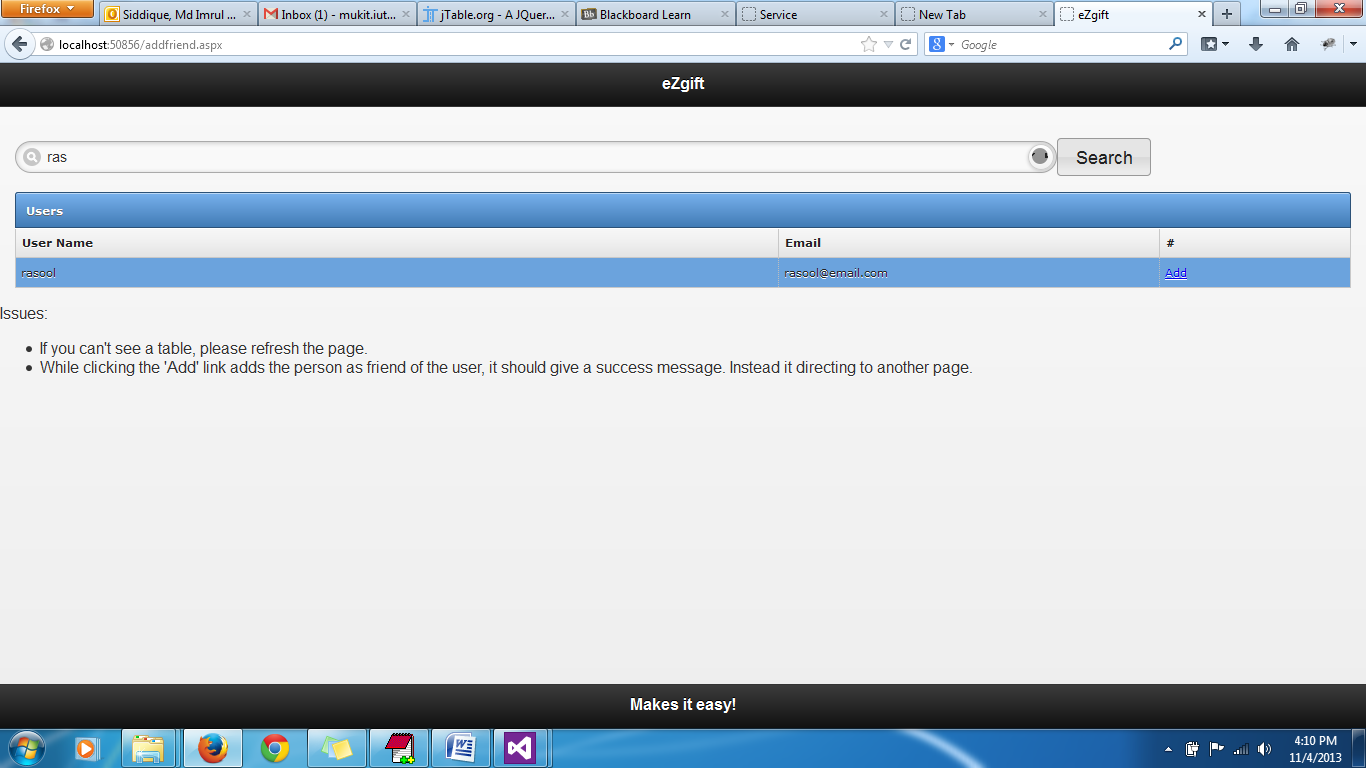
Home Page:

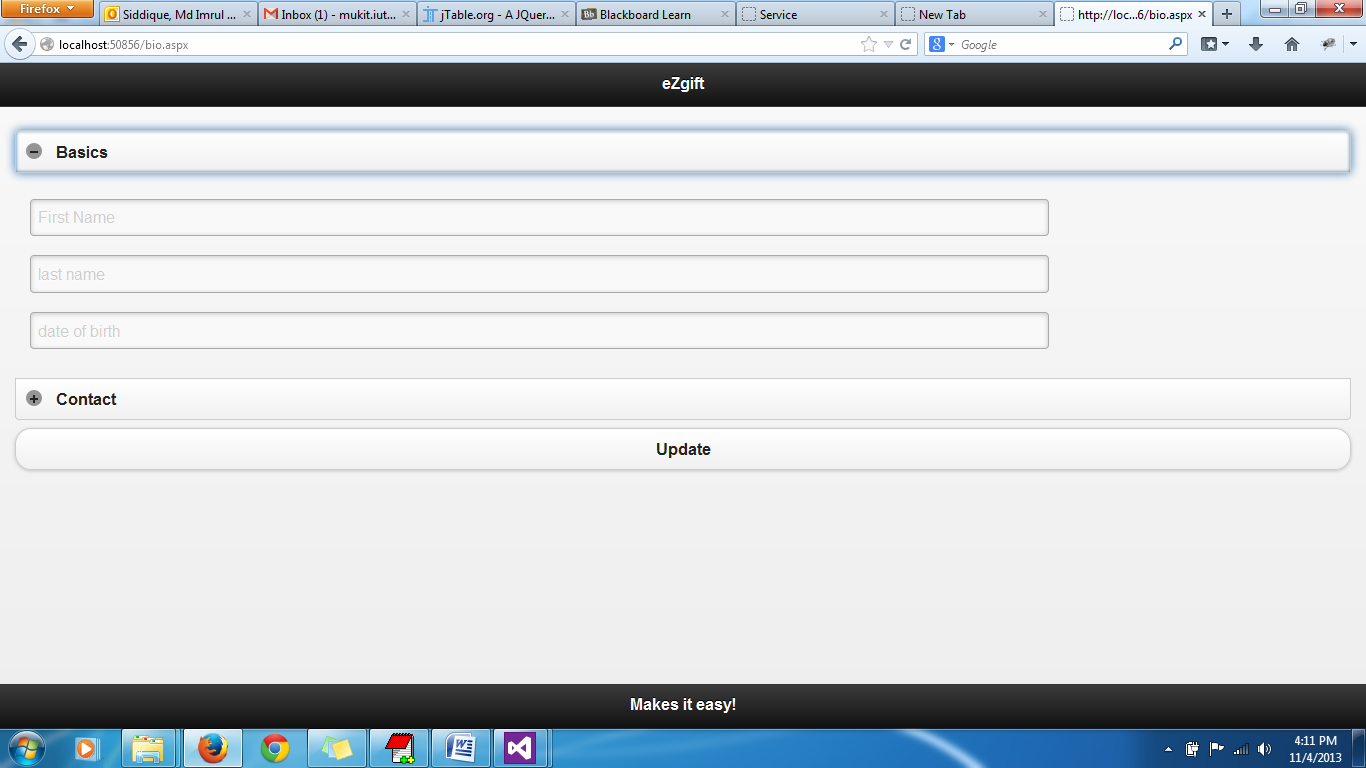
Figure : The Home page

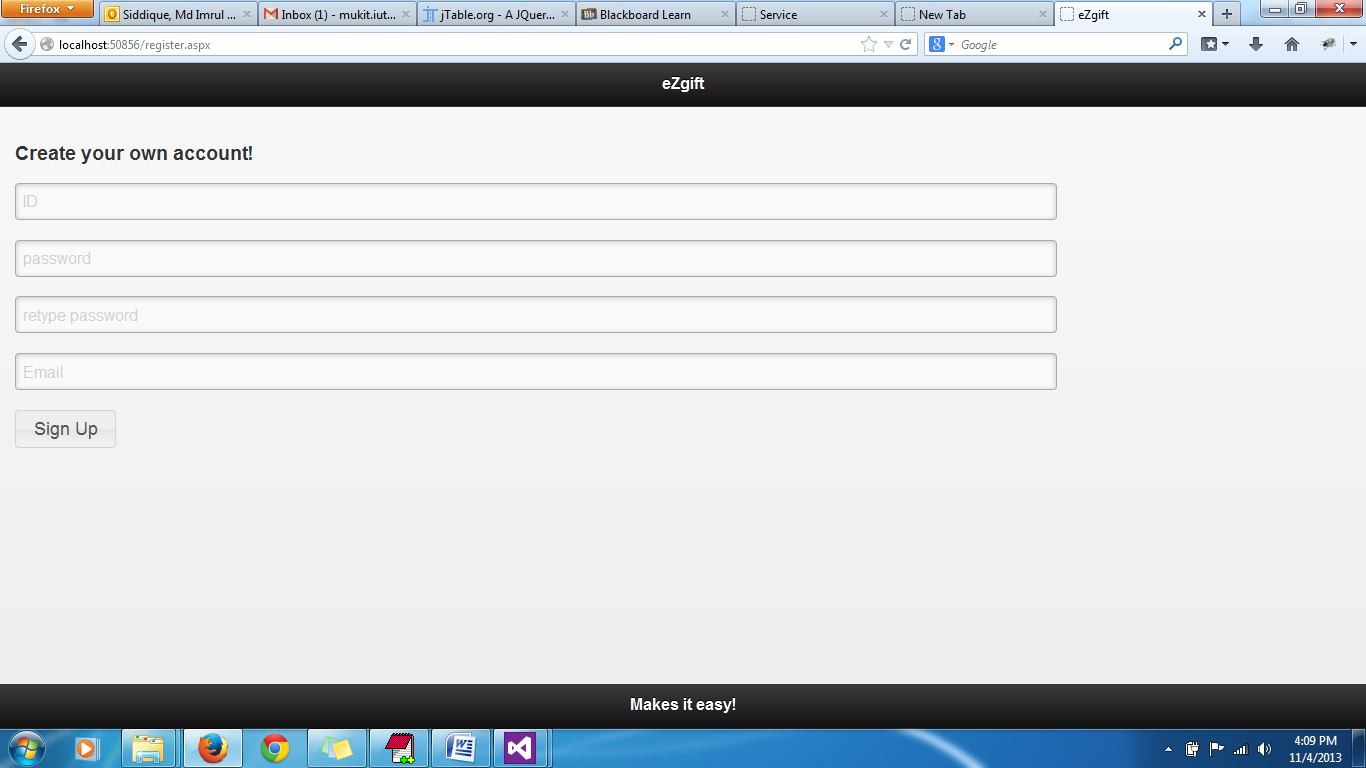
Wish List: 

Add Items in wish list: 

Friend List: 

Add Friend Page: 

Biography page: 

Register Page: 

## Limitation

The Mobile Web site has a few limitations which I couldn’t fix due to time constraint. However the overall structure is done.

## Future Work

This application is developed as required for midterm. However it gave me a good insight how to quickly develop a web application using different readily available web services. I would like to work on it in future and fix all the issues not done yet.