

MaidEasy

Admin Controller:

1. Admin_home ():

Parameters: void

Return type: Action Result

Loads Home page for admin.

Admin User Controller:

1. UserList ():

Parameters: void

Return type: Action Result

Used Method: RegisterController.getThanaID(Thana string)

Fetch user list from Database. Convert every users thana attribute data to thana name. In database we store thana attribute as a binary string.

2. MakeAdmin (int id):

Parameters: User Id

Return type: Action Result

Return To: AdminUser/UserList

Take user id as parameter and change type of user "general" to "admin", save changes to database. Then return to userlist.

3. RemoveAdmid (int id):

Parameters: User Id

Return type: Action Result

Return To: AdminUser/UserList

Take user id as parameter and change type of user "admin" to "general", save changes to database. Then return to userlist.

4. BlockUser (int id):

Parameters: User Id

Return type: Action Result

Return To: AdminUser/UserList

Take user id as parameter and change type of user "general" to "blocked", save changes to database. Then return to userlist.

5. UnblockUser (int id):

Parameters: User Id

Return type: Action Result

Return To: AdminUser/UserList

Take user id as parameter and change type of user "blocked" to "general", save changes to database. Then return to userlist.

6. CancelBlockedUserContracts (int id):

Parameters: User Id

Return type: void

Return To: BlockUser()

Fetch all current contracts from database regarding parameter id. Cancel all those contracts from current month. Set workers status free regarding contracted time span.

7. Convert(string end):

Parameters: string time

Return type: int

Used Method:

Convert time string to int as it will be used as an index of a binary string. And we use the binary to store information in database.

Example:

convert ("6:00 AM")

output: 0

this 0 means 0 index in the binary string.

8. getStatus (string status, int start, int end):

Parameters: current worker status, start of time, end of time

Return type: string

Takes starting and ending time as parameter and sets this time free.

Example:

```
getstatus("000000001100000", 0, 2)
status before function call: "000000001100000"
search time: 6:00 am - 7:00 am
status after function call: "111000001100000"
output: "111000001100000"
```

Admin Worker Controller:

1. Add_worker ():

Parameters: User Id

Return type: Action Result

Used Method:

Redirects to Add_worker page.

2. WorkerList ():

Parameters: void

Return type: Action Result

Used Method: getThanaList(thanastring),
getWorkerTypeList(workertypestring)

Fetch worker list from Database. Use getThanaList() and getWorkerTypeList() methods to show data in our desired format. Then shows the worker list.

3. Edit_Worker (int workerID):

Parameters: Worker Id

Return type: Action Result

Used Method: getWorkerTypeList(workertypestring)

Make worker profile page editable. Take worker id as parameter to fetch data from database. Use `getWorkerTypeList()` method to show data in our desired format. Then redirect to editable worker profile page.

4. **AddNewWorker (HttpPostedFileBase file):**

Parameters: file

Return type: Action Result

Used Method: `getThanaString (Thana name)`, `getTypeString ()`

This method is called when an admin submits worker info through `add_worker` page. The parameter in this method is that file which is uploaded as worker image. Takes form information using form request. Convert area and type information to the desired type of database table attribute. Rename uploaded file with workerID (if any). If all information is valid then store the info's in database and redirect to `workerlist.cshtml`. Otherwise redirect to `add_worker.cshtml`.

5. **SaveWorkerData (HttpPostedFileBase file):**

Parameters: file

Return type: Action Result

Used Method: `getThanaString (Thana name)`, `getTypeString ()`

This method is called when an admin edits worker info through `edit_worker` page. The parameter in this method is that file which is uploaded as worker image. Takes form information using form request. Convert area and type information to the desired type of database table attribute. Rename uploaded file with workerID (if any). If all information is valid then store the info's in database and redirect to `edit_worker` page in noneditable version.

6. **DeleteWorker ():**

Parameters: void

Return type: Action Result

Used Method:

Get workerID from Session variable. We save this when we enter into the worker profile. By calling this method we delete all information of this worker from database. Then remove session and redirect to worker list.

7. **getThanaList (string thanastring)**

Parameters: a binary string for thana attribute in database

Return type: string

Used Method:

Example:

```
getThanaList ("00010010");  
output: Bangsal  
        Chowkbazar
```

here, "00010000" is the binary string for thana which we pass in this method as parameter. This method iterates over the passed string. When gets '1', fetch thana name from database for thana id = pos +1. Here, pos is the position of '1'. Store thana name in a string and add new line after each name except the last one. Then return the string.

8. getWorkerTypeList (string typestring):

Parameters: a binary string for type attribute in database

Return type: string

Used Method:

Example:

```
getThanaList ("0001");  
output: Elderly Care
```

here, "0001" is the binary string for type which we pass in this method as parameter. This method iterates over the passed string.

9. getThanaString (string list):

Parameters: a string which contains list of area of a worker

Return type: string

Used Method:

Example:

```
getThanaString ("Badda\nDemra");  
output: "001000001"
```

we pass thana list of a worker (where he/she is available to hire) as a string in this method. Method creates a binary string for the parameter and return this.

10. getTypeString (bool a, bool b, bool c, bool d):

Parameters: Each parameter represents different types. If hr/she is this type of worker then its true otherwise false.

Return type: string

Used Method:

Example:

getTypeString (false, true, false, false)

output: "0100"

Home Controller:

1. Index ():

Parameters: void

Return type: Action Result

Used Method:

This function is called when we load the home page of our project. Depending current date of month, it updates worker experience. Then if you are a logged in user, depending on your user type it redirects to user home page or admin home page.

2. About ():

Parameters: void

Return type: Action Result

Used Method:

Loads about us page.

3. Contact ():

Parameters: void

Return type: Action Result

Used Method:

Load contact page.

4. EntryContactInfo ():

Parameters: void

Return type: Action Result

Used Method:

This method is called when a user submits contact us page. Then it directs to Home page.

Maid Controller:

1. Maid profile ():

Parameters: void

Return type: Action Result

Used Method:

This method is called from 2 pages. One from service page view profile button and another one is hired worker page view profile button. Then it goes to maid profile page and show maid's information. Then if you go through maid search then you find a hire button to hire this maid, otherwise not.

2. Hire ():

Parameters: void

Return type: Action Result

Used Method:

This method is called from maid profile hire button. When a user goes to a maid profile and wants to hire this maid, then user hits the button and this method is called. It's just redirect user to hire page.

3. getEndMonth (string StartMonth, int ConLength):

Parameters: Selected starting month and contract length for hiring a maid

Return type: string

Used Method:

Example:

getEndMonth ("06/20", 3)

output: "09/20"

4. getStatus ():

Parameters: void

Return type: string

Used Method:

Fetch status of worker from database, set new status and return it. We need new status as hired maid will not available be in the booked time. Here we get search time from session variable. So, we call this method when we need to hire a maid in our preferred time.

Example:

getstatus ()

status before function call: "000000001100000"

search time: 6:00 am - 7:00 am

status after function call: "111000001100000"

output: "111000001100000"

5. checkStatus ():

Parameters: void

Return type: Boolean

Used Method:

Before make a booking confirm we have to check that is this maid booked already in this time. So, before confirmation we recheck the status of the worker and call this method.

6. Booking ():

Parameters: void

Return type: ActionResult

Used Method: getEndMonth(), getStatus(), checkStatus()

This method is called, when we need to book a maid. Except temporary type maid it is redirect to BookingFull(). It is called from hire page when a user hits hire button. This method gets all kinds of booking information and try to make a contract if it is possible. If the maid is already booked, then it redirects user to service page. Otherwise, make a contract and redirects to hired workers page.

7. BookingFull ():

Parameters: void

Return type: ActionResult

Used Method: getEndMonth(), getStatus(), checkStatus()

This method is called, when we try to book a maid except temporary maid. All other work is as same as Booking().

Register Controller:

1. SignUp():

Parameters: void

Return type: ActionResult

Used Method:

Loads the signup page.

2. LogIn():

Parameters: void

Return type: ActionResult

Used Method:

Loads the login page.

3. Logout():

Parameters: void

Return type: ActionResult

Used Method:

Done logout and remove sessions.

4. VerificationCode():

Parameters: void

Return type: ActionResult

Used Method:

This method is called when we have to verify our phone no.

5. getThanaString(string thana):

Parameters: name of thana

Return type: string

Used Method:

Take thana name as parameter then build a binary string. String is build based "1" for the id index of parameter, else "0".

6. getThanaID (string thana):

Parameters: binary string of thana

Return type: int

Used Method:

Thana binary string is "1" for only our desired thana. So, we can easily get thana id by iterating over the string.

7. **CheckInfo ():**

Parameters: Image file

Return type: ActionResult

Used Method: getThanaString(), hash()

This method is called when a new user tries to register in our website. User input his/her data and tries to register. We take data match given username in our database. If given username match with any other user's username then show popup and redirects to signup page. Otherwise hash the given password and redirects to phone no verification page.

8. **hash(string password):**

Parameters: users given password

Return type: string

Used Method:

Takes a string as a given password, hash it and return it.

9. **checkPassword (string a, string b)**

Parameters: stored password in database, users given password

Return type: boolean

Used Method:

Check 2 passwords which was given in the parameter., as one in the hashed format. Then return true/ false depending on the match result.

10. **AddUser():**

Parameters: void

Return type: ActionResult

Used Method:

When mobile verification is done then this method is called to insert new user to database. After inserting data, it redirects to user home page.

11. **VerifyUser ():**

Parameters: void

Return type: ActionResult

Used Method:

When a user tries to log in to the website, this method is called. This method takes user input from and verify password. After verifying save session variables and redirect to user home page. Else show log in error.

12. **getEditprofileData ():**

Parameters: Image file

Return type: ActionResult

Used Method:

In edit profile page, user input their information to edit his/her profile. Then when hits a button to proceed this method is called. In this method we match given old password with our stored password in database. If it fails then redirect to the same page otherwise goes for phone no verification.

13. **saveEditProfile ():**

Parameters: void

Return type: ActionResult

Used Method:

After phone no verification in edit profile this method is called. It saves updated information in database and redirects to user profile.

Service Controller:

1. **Service():**

Parameters: void

Return type: ActionResult

Used Method:

Loads out website service page when user clicks service in navbar.

2. **Convert(string end):**

Parameters: string time

Return type: int

Used Method:

Convert time string to int as it will be used as an index of a binary string. And we use the binary to store information in database.

Example:

convert ("6:00 AM")

output: 0

this 0 means 0 index in the binary string.

3. findType (string s):

Parameters: string time

Return type: int

Used Method:

Take maid type as string and convert to an int index, then return that index.

4. Searching():

Parameters: void

Return type: ActionResult

Used Method:

When user insert his/her desired user type as temporary maid and input time period, he has to be hit search button to see the available maids in his/her area. When he/she hits search button this method calls. This method fetches available workers for the input constraints of the user and again redirects to the service page.

5. SearchingFull ():

Parameters: void

Return type: ActionResult

Used Method:

When user insert his/her desired user type as full-time maid, he has to be hit search button to see the available maids in his/her area. When he/she hits search button this method calls. This method fetches available workers for the input constraints of the user and again redirects to the service page.

User Controller:

1. Feedback():

Parameters: void

Return type: ActionResult

Used Method:

To load feedback page. If user wants to give feedback to any of his/her hired worker then click feedback button in hired workers page and this method is called.

2. getThana(string s):

Parameters: thana information in binary string

Return type: string

Used Method:

Returns thana name for "1" in index no of thana string.

3. **user_profile (int id)**

Parameters: user id

Return type: ActionResult

Used Method:

Fetch user information from database goes to user profile view.

4. **hired_worker_profile ():**

Parameters: user id

Return type: ActionResult

Used Method:

Fetch hired workers information from database goes to hired workers page for user.

5. **Edit_profile ():**

Parameters: user id

Return type: ActionResult

Used Method:

From user profile if user wants to edit his/her profile, click edit profile button then this method is called. This method fetches user information from database and goes to edit profile.

6. **Entry feedback ():**

Parameters: user id

Return type: ActionResult

Used Method:

When user gives feedback to a hired maid click feedback button then this method is called. This method saves user's feedback to database and redirects to user profile.

DB Helper:

1. **DBConnection ():**

Parameters: void

Return type: void

Creates mysql connection.

2. **getDB ():**

Parameters: void

Return type: DBHelper

Return an object of DBHelper class to do database operations. Singleton pattern is used here.

3. **getData ():**

Parameters: string query

Return type: SqlDataReader

Take a query as parameter. This query is a type of fetching some data from database. Then execute the query and returns the result.

4. **setData ():**

Parameters: string query

Return type: void

Take a query as parameter. This query is a type of set/update some data of database. Then execute the query .

Number Authentication JS:

1. **phoneAuth(num):**

Parameters: phone number

Send message to the number which takes as parameter.

2. **Codeverify(cv):**

Parameters: flag which decides its destination

Take code number from field which was input by user in code field. After verification, by its parameter it can understand why this function is called.

cv = 1 means it calls to add a new user.

cv = 2 means it calls to edit user profile