

**ReviUL - UL Proof Reading Website**

# 1.0 Purpose of Website

ReviUL provides a medium for students and staff at the University of Limerick to find others to proofread their academic documents as well as allowing the proofreading of other user’s documents. Proofreading can be one of the more tedious tasks to complete when writing an academic paper, and when done by the writer alone it often leads to uncaught typos and some poorly constructed sentences. This site will help ensure that their paper is being graded for their ideas and concepts, not their grammar. It can also be difficult to find others to help with proof-reading tasks, as friends and family may not be informed with the specific terminology of the writer’s discipline and often colleagues are busy reviewing and writing their own papers. This website provides a means of finding individuals with experience within similar disciplines that are available to proof read papers and other documents.

At present there is no platform like this within UL – even outside UL these types of resources are limited to sites requiring payment which do not utilise crowdsourcing. While this platform may not be as high quality as these paid alternatives, it is a free service which should attract a range of people with experience from a variety of disciplines for reviewing.

A logged in user will be able to create a task and will also be able to accept tasks created by other. A task will specify a deadline for its acceptance as well as a deadline for the return of the reviewed document. The requester will then be able contact the accepter and send them the full document for review.

## 1.1 User Types

Below are the types of users who have access to the system as well as the functionality they have access to.

### 1.1.1 Visitor

A visitor will have access to very little of the site’s functionality – they will only be able to access the register / login page and will be prompted to either login or register. They will not be able to see any of the sites content including profiles or tasks.

### 1.1.2 Standard User

A standard user will be able to login, view and update their profile information (excluding their email and UL ID). They can also create, view and accept tasks. A user can look for new tasks by checking the Tasks Stream, which will show tasks most relevant to that user (based on their tags). A user that has accepted tasks (Claimant) can view these tasks on a dedicated page and request for the full file for review through an email template (claimants will not have access to the task creator’s email). They will also be able to mark a task as completed once they have finished proof reading the document. If they are unable to complete the task and mark it cancelled their reputation score will be reduced by 15 marks. If they do not complete or cancel the assignment by the deadline they will be penalised by having 30 marks removed from their reputation score. The user who requested the task (Creator) will be able to view the claimant’s name and email once they have claimed their task. Once the claimant has completed the task, the creator will be able to review the claimer for their work by stating whether they were ‘Happy’ or ‘Not Happy’ with the review. If they choose ‘Happy’ 5 marks are added to the claimers score, if they choose ‘Not Happy’ 5 marks are removed.

Users can also flag tasks created by other users if they feel they include inappropriate material. They will also be able to post a review of users who were assigned to tasks they had created. A standard user has a reputation score of under 40, otherwise they will be upgraded to a moderator.

### 1.1.3 Moderator

A moderator will have access to all of the functionality of a standard user (above) as well as additional privileges including ability to review flagged tasks which will be visible on a dedicated page as well as the option to ban the creators of these tasks. A moderator will also be able to remove these flagged tasks from the available list of tasks.

### 1.1.4 Banned User

A banned user will no longer be able to login or register again with the same email or UL ID. As this system is based on providing a UL email they will be unable to join the site again.

## 1.2 Justification of Functionality

Once a user selects the website on the browser, they will be brought to a page in which they will have the ability to register or sign into ReviUL. If the user decides to register if they have not previously done so, they will be requested to enter their email address, first name, last name, relevant discipline, tags, password and also to re-enter their password; this will all be completed in text boxes provided, similar to this (see Appendix 1: Figure 2 Webpages). If the password does not match in both password text boxes, the user will be asked to re-enter their password as it is incorrect. Once all the fields are entered they can register for the website. If the user has registered for the website on a previous occasion they can go to the login section. This section will have two text boxes asking for the user to enter their email and password. There will also be an option to click "forgotten password" if the user does not recall their password. Once all fields have been completed the user can click the button to login, and they will be granted access to the system if the information is correct (see Appendix 1: Figure 1 Webpages).

Once logged in, the user will be brought to their user page, which would contain user details such tasks they have uploaded (see Appendix 1: Figure 3 Webpages). There will be a menu bar, which will include the option to log out, see claimed tasks by that user, see what users have claimed their tasks and to search for other tasks which they may have an interest in proof reading.

The task section of the page will show information in relation to the article that a user has uploaded for other users to view. When users upload information about their task, they will have a screen similar to this page (see Appendix 1: Figure 4 Webpages). The information provided with this function will allow other users to contemplate if they will to claim this task to proof read. There will also be the option to message the user in order to claim tasks on the user page.

The proposed website will look similar to this figure in which the user will log in or register for the website (see Appendix 1: Figure 5 Webpages).

## 1.3 Potential Ramifications

The potential ramifications for a website of this nature on users would that it would provide an easy and accessible platform for users to share their work for proof reading and help other users by proof reading their work. It would be easy to use as users could upload details of their work on the website for other users to search through and gather information on, to ensure that capable of proof reading such a document. This proposed website would enable users to increase their skills and knowledge by aiding others but also build a community in which users give and receive help.

There is also a potential increase for scope of this website for users to be able to upload more than one document at a time. Scope could also be increased by offering services to other colleges and universities not only around the country but possibly worldwide.

However, there are some potential issues which may occur with such a website. Users leaving tasks which have not been claimed for a long time. This has potential to clog the system and database with outdated tasks if left for a long period of time. To address this issue, there would be an expiry date set in the database to every task in order to ensure tasks are up to date. Users would also be aware of expiry date setting when uploading a task to the website.

There is also the possible issue of users not removing "claimed tasks" from the website, thus allowing other users to attempt to claim them. This would make the search capabilities of the system overloaded with unavailable tasks. To combat this issue, there would be an option for the user to select that their task has been claimed. Once the user has selected this option, the task will appear as such on the website for other users to see. It will also not appear when other users are searching for tasks to proof read. Users will be notified each week to update if tasks have been claimed or not.

# 2.0 Detailed Description

When the user launched the proposed website, the homepage will be similar to that in figure 5 (see Appendix 1: Webpages).No processes will be needed to generate this page. The user can navigate to other pages from this home page and processes can be triggered from these further pages.

P1 (Registration) will be executed when a user wishes to sign up to the website to avail of proof reading help. Users will need to provide information to submit to the form displayed on the webpage. See Figure 5 (see Appendix 1: Webpages**).** The input will be validated and once it is deemed that the information is correct and in the correct format for inputs such as email and password, the information will be sent to the database. In the database, the new entries will be added to the correct tables. The user will then be brought to their profile page. If the information provided in the web form is incorrect, an error will be displayed and the user will be informed what fields were completed incorrectly.

P2 (Login) is executed if the user selects login on the registration page. Users who have previously registered will be required to fill in their email and password. This information will then be validated and the information in the database will be queried in order to complete this step. If the information submitted is correct, the user will be brought to their profile page. If not, an error will be displayed informing the user that the information inputted is incorrect. See Figure 6 (see Appendix 1: Webpages).

P3 (Forgotten Password) is executed once a user selects the forgotten password option on the login screen. The user would input the email they wish the computer generated password to be sent to. The user would then submit and the password would be sent to the user. See Figure 7 (see Appendix 1: Webpages).

P4 (Profile Page) is executed once the user has logged in to the website and their credentials have been validated. The user page displays information on the tasks uploaded by the user. This form would include their name, email and message. The message can then be sent to the user. See Figure 8 (see Appendix 1: Webpages).

P5 (Task Upload) is executed when a user selected to upload details on a task. This would bring them to a web form which would require them to fill out details around their task. These details would include file type, a brief description of their task, tags to describe the task, task type, page count, word count and a sample of the document. Once the user selects submit, the information would be sent to the database and the information would be entered into the correct tables in the database.

P6 (Tasks) is executed when a (user) potential user selects a task. This would display the information that the task creator (user) has uploaded in the task upload page. This information would be queried from the database and displayed in the correct format on the web page. It would also contain the option for other users to message them in relation to tasks they have uploaded. This form would include their name, email and message. The message can then be sent to the user.

P7 (Flagged Tasks) executes if another user has flagged a task if they deemed it inappropriate. Users who have enough reputation points to be a moderator will be notified that there is a flagged task. A moderator will then review the flagged task and decide if the task should be removed and the user banned. If they deem it as inappropriate, the information will then be sent to the database in order to delete the task form the task table and input the user as banned in the user type.

# 3.0 Benefits and Limitations

The benefits of the proposed website include having a community of individuals with the same goal of giving and receiving aid in relation to proof reading. Users can upload details of their task with ease and provide information for other users about the task, which will also aid the claiming process.

Limitations may include that users can only upload one document at a time. The user cannot upload their actual file to the website and the user must be contacted in order for other users to receive the file in order to proof read it. This may make actually proof reading a task a long process.

# 4.0 Technologies

Collection of technologies will be used to accomplish various tasks in creating the proposed website.

1. Operating System:

Microsoft Windows 7 / Linux will be used as OS for developing the proposed website.

1. **IDE:**   
    Notepad++ and Adobe Dreamweaver will be used as Integrated Development Environment for the proposed project.
2. Browsers:  
   Google Chrome, Safari, Firefox and Internet Explorer   
   Theses browsers will be used to view the web pages.
3. **HTML5:**  
   This mark-up language will be used to provide the structure of the website.
4. **CSS3:**  
   The Cascading Style Sheet will be used to change colours, fonts, animations, and transitions on the proposed project to make the web look good.
5. **Bootstrap:**  
   Bootstrap is one of the most popular front-end frameworks and open source projects. It is an HTML, CSS and JavaScript framework that will be used as a basis for creating the proposed website.
6. **JavaScript**:  
   JavaScript is an object-oriented programming language which will be used in the project to create interactive effects within web browsers.
7. **PHP:**  
   PHP is open-source general-purpose scripting language which will be used in web development and can be embedded into HTML.
8. **JQuery:**  
   JQuery is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) and open source [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) designed to simplify the [client-side scripting](https://en.wikipedia.org/wiki/Client-side_scripting) of [HTML](https://en.wikipedia.org/wiki/HTML). JQuery allows for changes to be made to HTML pages in a similar way to JavaScript.
9. **MySQL:**  
   MySQL is a database management system which will be used on the proposed project. It runs on server and is very fast, reliable and easy to use.
10. **Git (**[**www.github.com**](http://www.github.com/)**) :**

Git is a version control system for tracking changes in the computer files and coordinating work on those files among the team. Git will be used in the proposed project to keep track of changes in any files.

**Appendix 1: Web Pages**

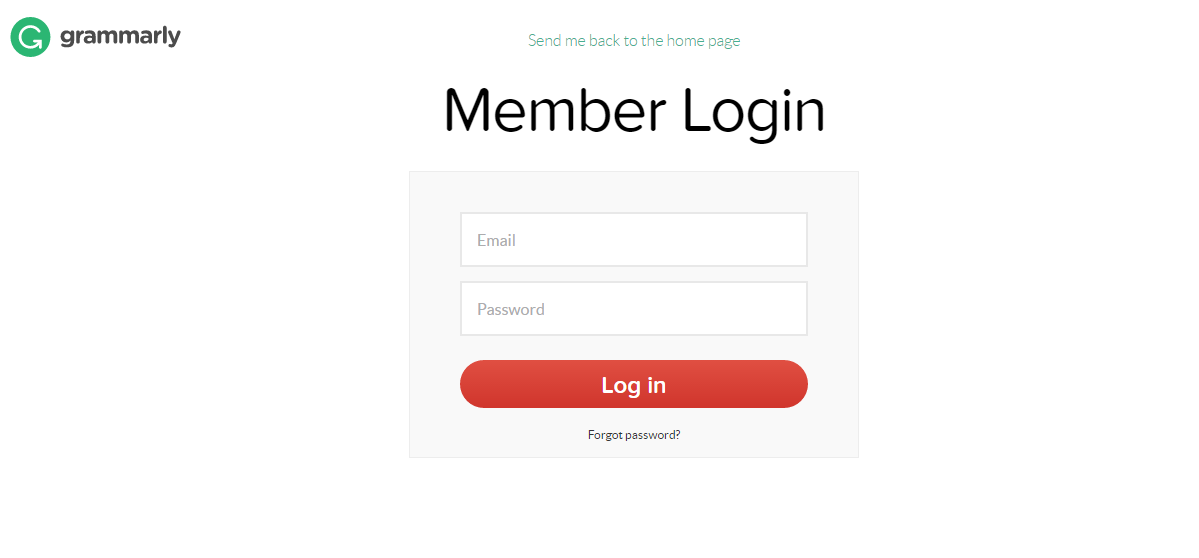


Figure 1: Log in screen from [www.grammarly.com](http://www.grammarly.com/)

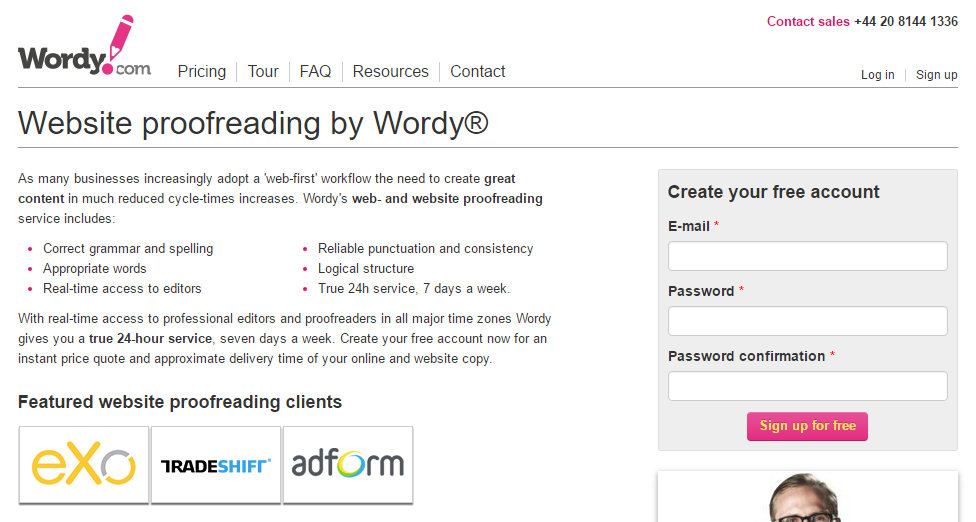


Figure 2: Registration screen from [www.wordy.com](http://www.wordy.com/)

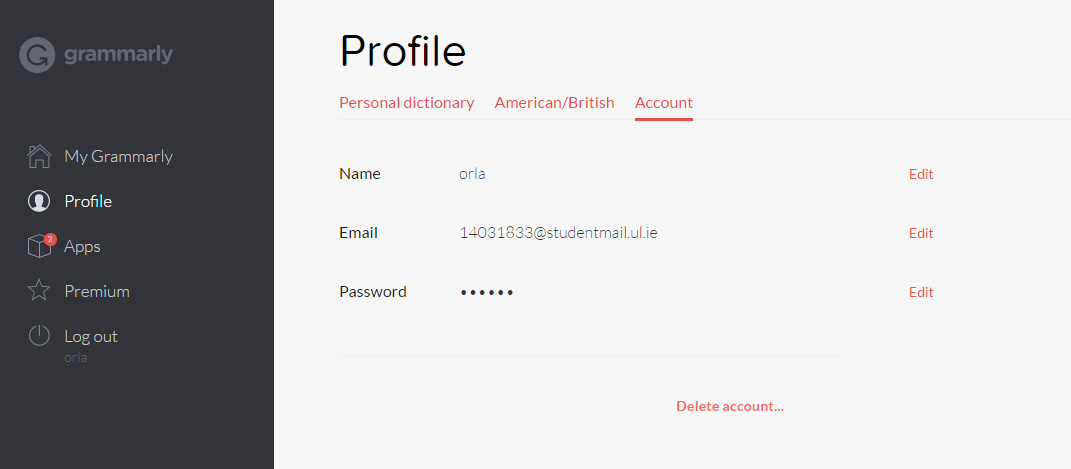


Figure 3: Profile page from [www.grammarly.com](http://www.grammarly.com)

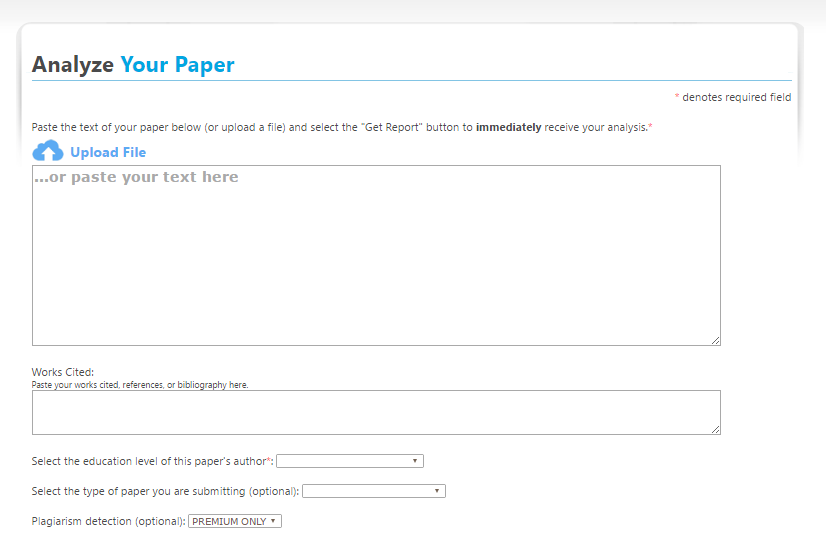


Figure 4: Task Information Upload Screen from [www.paperrater.com](http://www.paperrater.com/)

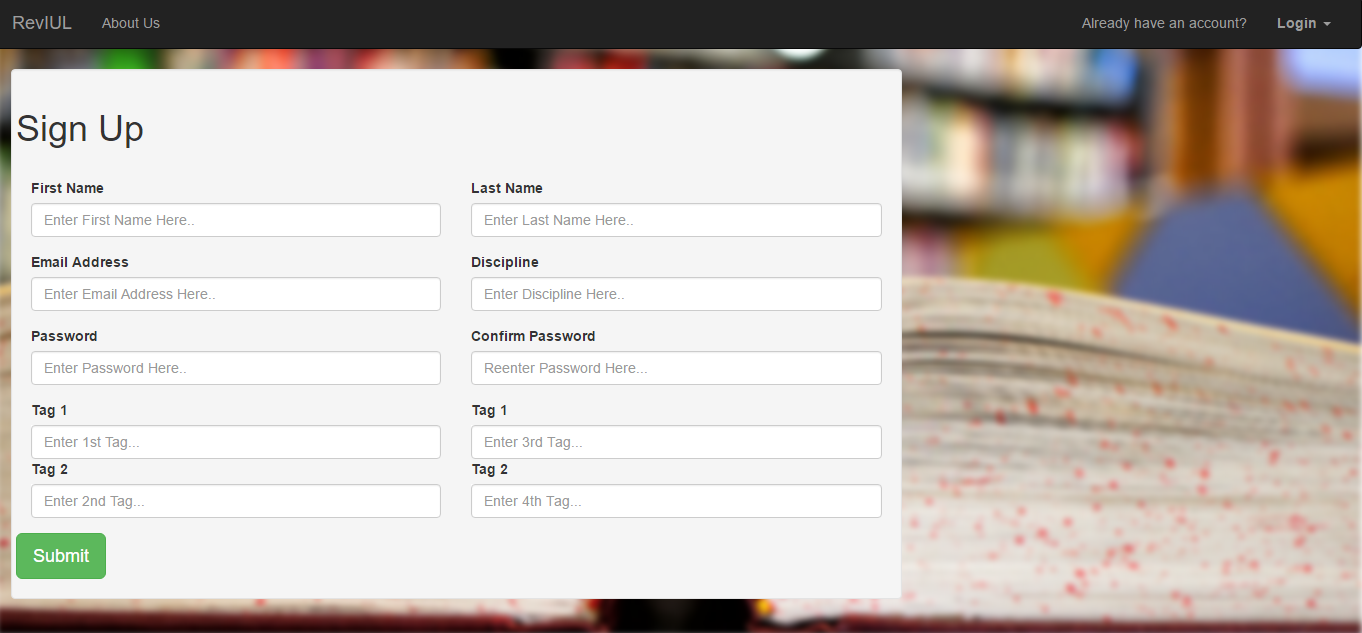


Figure 5: Registration page for ReviUL

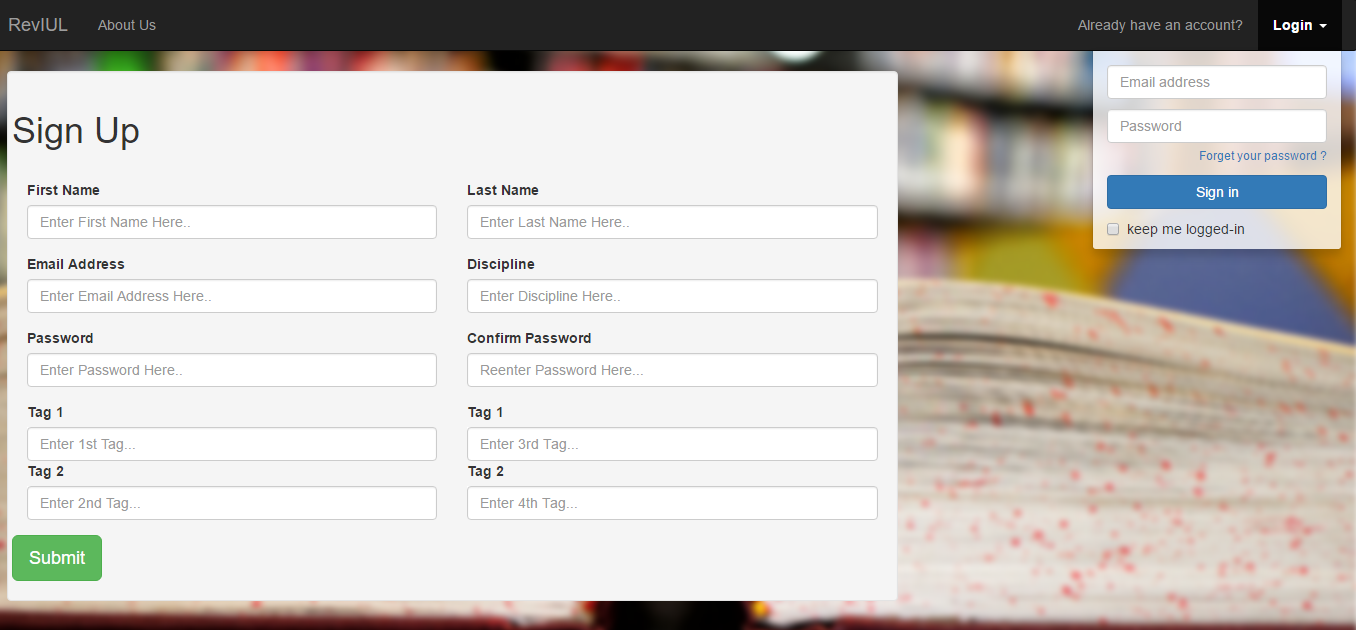


Figure 6: Login section on top right hand corner of website

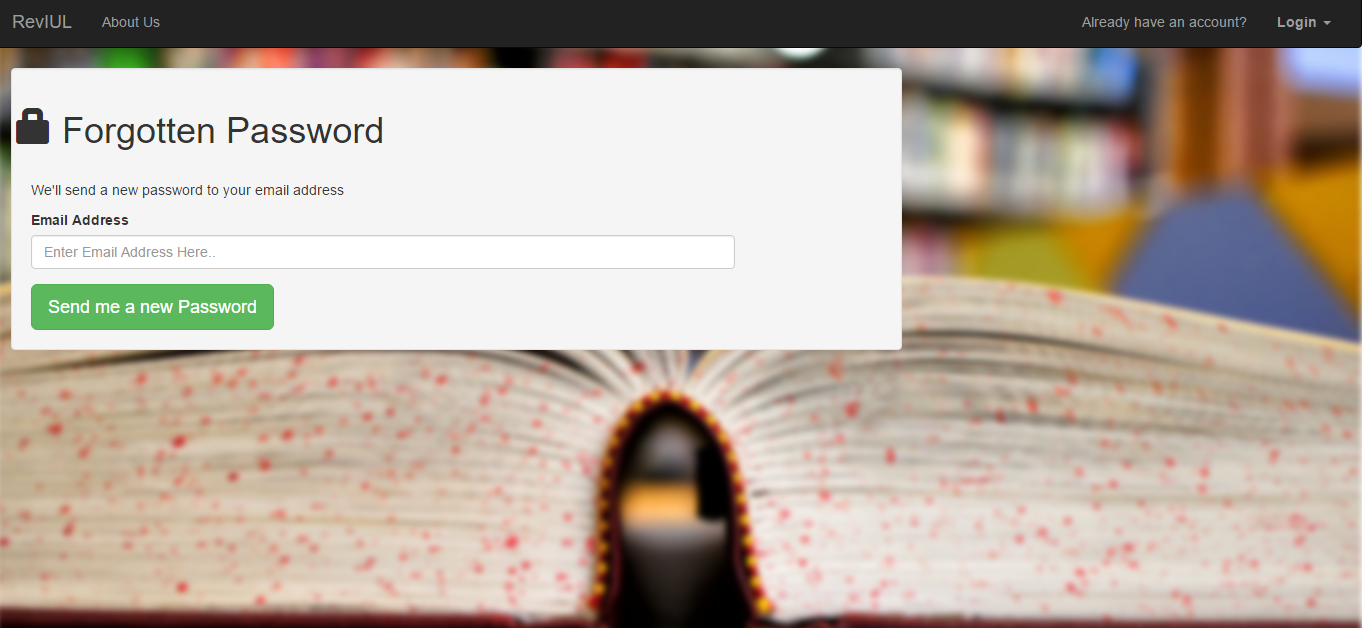


Figure 7: Forgotten Password page for ReviUL.

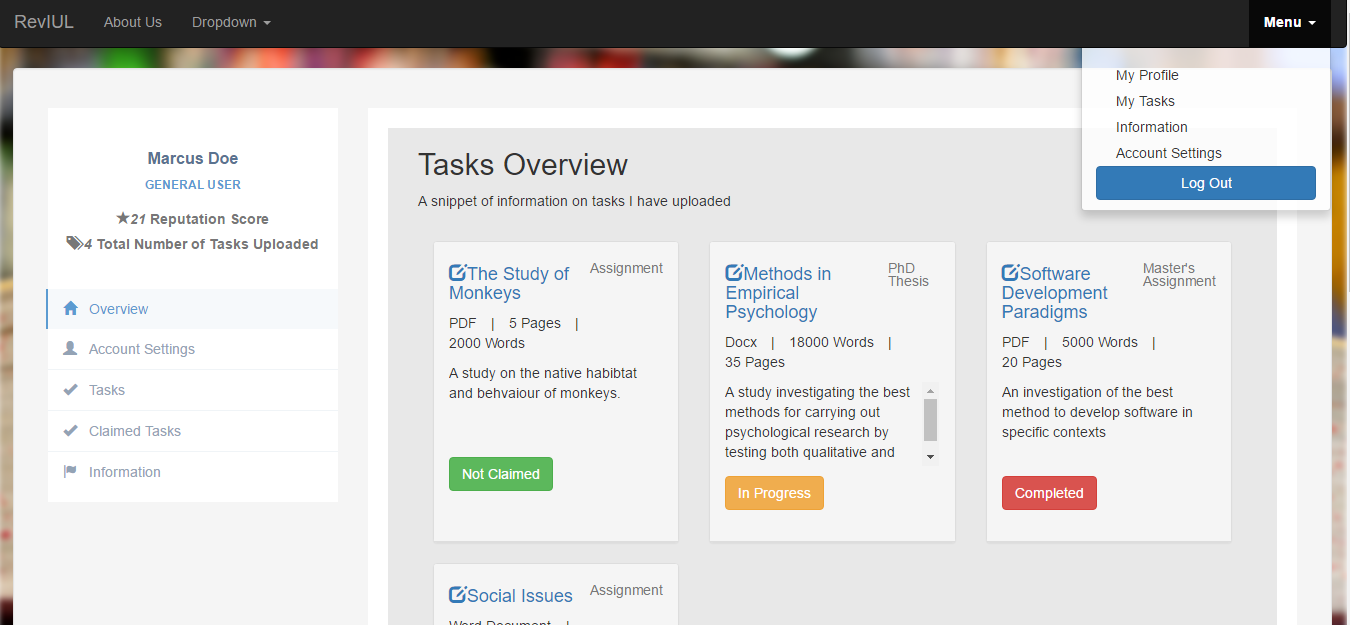


Figure 8: Example Template of a standard user profile on ReviUL

**Appendix 2: Database Tables**

Primary Key =

|  |
| --- |
|  |

Foreign Key =

|  |
| --- |
|  |

**Table Name:** User

**Primary ID:** user\_id

**Description:** The user tables stores user\_id, first and last name, email, password, user-type, discipline and reputation for each user.

**User**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| user\_id | Number | 123456 |
| f\_name | Text | Murphy |
| l\_name | Text | Mary |
| email | Text | mary@ul.ie |
| pass | Text | Staff |
| user-type | Text | Moderator |
| discipline | Text | History |
| reputation | Number | 20 |

* **user\_id** is a unique identifier for each user. Each user will have one unique ID. The id is numeric and can be 11 digits long.
* **f\_name** is the user's first name. The f\_name is alphabetic and can be 100 characters in length.
* **l\_name** is the users surname. The l\_name is alphabetic and can be 100 characters in length.
* **email** is the users email. The email is numeric and can be 128 characters in length. It is also must be unique.
* **pass** is the users unique password. The pass is alphabetic and can be up to 128 characters in length.
* **user-type** is the user’s access type – this is entered as a varchar and will be limited to standard, moderator, banned. User-type is alphabetic and can be up to 40 characters in length.
* **discipline** is the users department in university. The discipline is alphabetic and can be up to 128 characters in length.
* **reputation** is the users reputation depending on their performance on the website. Reputation is numeric and can be 11 digits long.

**Table Name:** Task

**Primary ID:** task\_id

**Description:** The user tables stores task\_id, creator\_id, documentID, task\_type, description, task\_status, claim\_deadline, completion\_deadline.

**Task**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| task\_id | Number | 954783 |
| creator\_id | Number | 548932 |
| document\_id | Number | 478132 |
| task\_type | Text | PhD thesis |
| description | Text | Study of Monkeys |
| task\_status | Text | Claimed |
| flagged | Number | 0 |
| claim\_deadline | Date | 20 – 02 - 2018 |
| completion\_deadline | Date | 20 – 09 - 2017 |

* **task\_id** makes reference to the individual task uploaded by a user. It would be unique to that task. Task\_id is numeric and can be 20 digits long.
* **creator\_id** is the reference to the creator of the task. It is a foreign key. Creator\_id is numeric and can be 11 digits long and can be 11 digits long.
* **document\_id** is the identifier for the document type which will makes reference to the table Documents (see Documents). DocumentID is numeric and can be 11 digits long.
* **task\_type** will describe the type of document the user is uploading. Task\_type is alphabetic and can be up to 128 characters in length.
* **description** will hold the description of the task. Description is alphabetic and can be up to 200 characters in length.
* **task\_status** holds information if the task has been claimed or not. Task\_status is alphabetic and can be up to 40 characters in length.
* **Flagged** will state is a task has been flagged or not by another user. It will be numeric and can be 1 digit in length.
* **Claim\_deadline** is the deadline for the task to be claimed. Claim\_deadline is numeric.
* **Completion\_deadline** is the deadline that the task must be reviewed by once claimed. Completion\_deadline is numeric

**Table Name:** Claimed\_Task

**Primary ID:** claimer\_id, task\_id

**Description:** The task tables stores task\_id and creator\_id

**Assigned\_Task**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| claimer\_id | Number | 911283 |
| task\_id | Number | 121416 |

* **claimer\_id** is the identifier of who has claimed a task. Claimer\_id is numeric and can be 20 digit in length.
* **task\_id** makes reference to the individual task uploaded by a user and it is a foreign key. It would be unique to that task. task\_id is numeric and can be 11 digit in length.

These two are both primary and foreign keys – each come from a different table and the combination of both identify a unique row in the table.

Table Name: Document

Primary ID: document\_id

**Description:** The task tables stores document\_id, no\_pages, no\_words, format and storage\_address.

**Document**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| document\_id | Number | 369852 |
| no\_pages | Number | 4 |
| no\_words | Number | 952 |
| format | Text | pdf |
| storage\_address | Text | C:\Files\CurrentFiles\samplethesis.pdf |

* **document\_id** is the unique identifier for a document. Document\_id is numeric and can be 11 digits in length.
* **no\_pages** is the number of pages in a document. Document\_id is numeric and can be 11 digits in length.
* **no\_words** is the number of words in a document. Document\_id is numeric and can be 11 digits in length.
* **format** is the document type. Format is alphabetic and can be up to 5 characters long.
* **storage\_address** is where the document is kept. Storage\_address is alphabetic and can be up to 200 characters long.

Table Name: Task\_Tag

Primary ID: task\_id and tag\_id

**Description:** The task tables stores task\_id and tag\_id.

**Task\_Tag**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| task\_id | Number | 112233 |
| tag\_id | Number | 758961 |

* task\_id is the unique identifier for a task. It is also present in the Assigned Task and Task table (see Task and Assigned Task). Task\_id is numeric and can be 20 digits in length
* tag\_id is the unique identifier for the tags a user inputs. Tag\_id is numeric and can be 11 digits in length

Table Name: Tag

Primary ID: tag\_id

**Description:** The task tables stores tag\_id and tag\_name

**Tag**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| tag\_id | Number | 182235 |
| tag\_name | Text | Women in History |

* **tag\_id** is the unique identifier for a tag. It is also present in Task\_Tag table (see Task\_Tag). Tag\_id is numeric and can be 11 digits in length.
* **tag\_name** is the name attached to a tag. Tag\_name is alphabetic and can be up to 128 characters long and it must also be unique.

Table Name: User\_Tag

Primary ID: user\_id and tag\_id

**Description:** The task tables stores user\_id and tag\_id

**User\_Tag**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| User\_id | Number | 188835 |
| Tag\_id | Number | 101010 |

* user\_id is the unique identier for users. It is also present in the User Table (see User). User\_id is numeric and can be 11 digits in length.
* tag\_id is the unique identifer for a tag. It is also present in the Tag, Task\_Tag tables (see Tag and Task\_Tag). Tag\_id is numeric and can be 11 digits in length.

**Show relationships between tables**

|  |
| --- |
|  |
| A **User** may or may not have any created tasks. A **Task** can only have one creator (User) and does not necessary have to be assigned to a claimer (Claimed\_Task). A **Claimed\_Task** can only be claimed once (PK task\_id). A **Task** also has one **Document** (this layout allows for multiple documents to be attached to a task in future versions). A **User** can have between one and four **Tags** – this is implemented through use of the **User\_Tag** Table. This maps user\_ids to tag\_ids, and removes possible nulls in the User table and will easily allow for more than four tags if ever required in the future. This is the same for **Task\_Tag** table – a task can have one to four tags as well. |

**Appendix 3: Processes**

|  |  |
| --- | --- |
| **Process Number** | P1 |
| **Process Title** | Registration |
| **Brief Description** | User inputs request for registration. Once completed correctly, the user will be directed to their new profile. The request for registration will be rejected if the information submitted is incorrect |
| **Inputs** | Information required from the registration form inputted by the user. |
| **Detailed Description** | User has the intent of registering with the website. The user completes the web form with the correct information displayed on the web page. This will result in the correct registration to the website. If the information is incorrectly input, an error will be displayed asking the user to re-submit the form with the correct information. |
| **Output** | Generates a new profile for the user and directs the user to it. |

|  |  |
| --- | --- |
| **Process Number** | P2 |
| **Process Title** | Login |
| **Brief Description** | Validates information user has inputted into the web form and allows or denies access to their profile. |
| **Inputs** | User email and password. |
| **Detailed Description** | User enters their email and password into the two text fields provided. This information will be their unique email and password they provided at registration. The process will validate if the information provided is correct in these two inputs. If correct they will be directed to their profile. If incorrect an error will be displayed and they will be requested to re-submit the form. |
| **Output** | Directs the user to their profile. |

|  |  |
| --- | --- |
| **Process Number** | P3 |
| **Process Title** | Forgotten Password |
| **Brief Description** | Allows the user to request a new computer generated password if they cannot relocate their own in order to access their profile. |
| **Inputs** | User email. |
| **Detailed Description** | User enters their email address and requests for a new password to be sent to them They select send new password and the system will generate a random password and email it to the user. The user will then be requested to input the new password into the log in screen. If correct they will be directed to their profile. If the information is incorrect they will be requested to re-submit the form. |
| **Output** | Directs the user to their profile. |

|  |  |
| --- | --- |
| **Process Number** | P4 |
| **Process Title** | Profile Page |
| **Brief Description** | User information will be displayed on this page. User information will include task details and the option for other users to contact them. |
| **Inputs** | User Login |
| **Detailed Description** | Once the user has successfully registered/logged in, they will be brought to their profile page. This page will contain snippets of information such as reputation scores, user type and overviews. This will be an overview of tasks they have uploaded. A tag overview of tags the user has used will also be displayed on this page. |
| **Output** | Task and tag overview. |

|  |  |
| --- | --- |
| **Process Number** | P5 |
| **Process Title** | Task Upload |
| **Brief Description** | Users will enter details into the provided text boxes in relation to their task. If the form is entered with the correct information and validated the task will upload. If incorrect information is provided, an error will be displayed and the user will be asked to resubmit the form again. |
| **Inputs** | Task information such as file type, a brief description of their task, tags to describe the task, task type, page count, word count and a sample of the document. |
| **Detailed Description** | Users will provide information about the task they wish to upload in the provided text fields. Such information will include file type, a brief description of their task, tags to describe the task, task type, page count, word count and a sample of the document. This information will be validated to ensure it is in the correct format. If incorrect information is provided, an error will be displayed and the user will be asked to resubmit the form again |
| **Output** | The users task will upload for other users to have the option to claim. |

|  |  |
| --- | --- |
| **Process Number** | P6 |
| **Process Title** | Tasks |
| **Brief Description** | Users will select a task after searching for it and information about the task will be displayed. |
| **Inputs** | Task information after searching for the task. |
| **Detailed Description** | After a user finds a task that they may have an interest in proof reading and selected said task, information on the task will be uploaded after querying the database for the information. The user can then check the information displayed on the task. Other users can then message them in relation to tasks they have uploaded in order to attempt to claim the task for proof reading. The other users can input the correct information into the text boxes provided to message the users. Once they click message, the message will be sent to the user. If fields are left empty or unfilled, the users will be alerted to this and prompt them to fill all text boxes before submitting a message. |
| **Output** | Task information and the option to message the user to claim the task. |

|  |  |
| --- | --- |
| **Process Number** | P7 |
| **Process Title** | Flagged Tasks |
| **Brief Description** | Moderators (users with the correct amount of reputation points) will be informed of a flagged task. They will then review the task and based on their decision the task may be removed and user banned or the task will be deemed as appropriate and no action will be taken. |
| **Inputs** | Another user will flag a task as inappropriate and moderators will be informed. |
| **Detailed Description** | A user may flag a task if they deem it to be inappropriate. Moderators will then be informed that a task has been flagged by another user. They will then review the task and decide if it is appropriate or inappropriate. If they deem it to be inappropriate, the information will then be sent to the database in order to delete the task form the task table and input the user as banned in the user type. Otherwise, if deemed appropriate, no action will be taken and the moderator will "de-flag" the task. |
| **Output** | User will be banned and task deleted or no action will be taken. |