**<Malaxy> 🌌**

**Requirements and Specification Document**

**<2022-12-14>, version 1.2**

**Project Abstract**

In this project, we want to design a website that would help students do their course selection, which would save the students a lot of research time, providing them with an integrated platform with plenty of information about other students’ comments, course content, grade distribution, etc.

**Document Revision History**

Rev. 1.0 <2022-10-12>: initial version

Rev. 1.1 <2022-11-17>: update database schema, rework user interface, add course filters to search feature, fix typos

Rev. 1.2 <2022-12-14>: update user interface, update use cases

**Customer**

Undergraduate and graduate college students are the primary stakeholders. At the same time, the customer base can be extended to any group of students who need to choose courses (including high school students, for example). Initially, we will specifically target students at the University of Wisconsin-Madison who want accurate, extensive, and comprehensive information. Everything they need to determine the best possible course depending on their requirements, will be provided in a single space. We did random surveys in Union South and discovered that students are interested in browsing the website to see how they are doing through the course evaluation and how they can potentially move forward.

Other customers include teachers, teaching assistants, and course designers. During the same survey in union south, we discovered instructors are also interested in getting feedback on their classes as it will be restriction-free, in a space where students can provide genuine evaluations on classes and professors without fear of recourse. They can then use the feedback from students evaluating the course, where the teaching team can improve their teaching style, content, curriculum, etc.

[UW-Madison's Registrar's Office](https://registrar.wisc.edu/) will provide a course database for us.

**Competitive Landscape**

While there are many websites that provide information to help with course selection, they are sometimes not up-to-date or not representative, and they are often on different websites, requiring students to go to many pages on each site to get comprehensive information about a class.

References to similar systems are madgrades and RateMyProfessor. However, unlike Madgrades and RateMyProfessor, Mad Course Evaluator gives a space for users to not only leave comments but also communicate on follow-up questions as well. Besides, RateMyProfessor aims at rating professors, which give limited help on whether the course is designed well or other course based information. Additional reference is to Amazon. The comment area in Amazon gives lots of hints to us as it allows users to rate a product, leave questions, and wait for them to be answered.

**Competitors Analysis**

* Madgrades (<https://madgrades.com/>)
  + Similarities:
    - search for a course
    - overview of the grade distribution over different semesters
  + Identified strengths:
    - allow interactions between users
    - karma system ensures the quality of discussion board
    - class comparison
    - shopping cart
* Rate My Professors (<https://www.ratemyprofessors.com/school?sid=18418>)
  + Similarities:
    - information about professors
    - rate professors
  + Identified strengths:
    - allow interactions between users
    - karma system ensures the quality of discussion board
    - class comparison
    - shopping cart
* UW-Madison Grade Distribution Search (<http://wisc.alext.se/>)
  + Similarities:
    - search for a course
    - overview of the grade distribution over different semesters, instructors
  + Identified strengths:
    - allow interactions between user
    - karma system ensures the quality of discussion board
    - class comparison
    - shopping cart

**Our Strengths**

* discussion board, users can share their opinions or ask questions about one course/instructor, we provide a community for students to communicate

**What can we learn from their implementation?**

* Madgrades does a good job on visualizing the grade distribution and users are able to export the statistics. Besides, Madgrades allows users to explore courses instead of only searching for a specific course.
* RateMyProfessor allows users to report ratings. This feature is pretty important since some ratings could be extremely toxic or biased.
* UW-Madison Grade Distribution Search allows users to search for a department and would display all courses provided by that department

**What product features can create competitive differentiators?**

* Madgrades, RateMyProfessor, and UW-Madison Grade Distribution Search do not exhibit too many details about the course/professor. However, it is highly likely students want to know more about the course structure or how a professor organizes the course. Therefore, our discussion board features can provide students with that information; students can easily post questions/express concerns about courses. One thing we need to think about is how to encourage users to discuss.
* Furthermore, our class comparison page allows users to compare courses so they can figure out the advantages/disadvantages of each course then make the final decision.
* Last but not least, our shopping cart feature allows users to add their favorite courses to a shopping cart, then they can access these courses from the shopping cart at a later point. Therefore, they do not need to go back to the main page and search for courses again.

**User Requirements**

1. **Visitor**
   1. **Front Page:** Visitor can see a number of most popular courses, a “Search Button”, a “Malaxy” button, and a ”Log in” button.
      1. When visitor click on a showing course, a window should show the most liked review of that course.
   2. Visitor can access the courselist page by clicking on the Courselist button located in the header of the website(Logged in user has access to the same user requirements of b)
      1. Visitor can search a course by typing into the “Search for a Course” input box
         1. When the user types in characters into the input box the courses displayed will change correspondingly
         2. When the user removes all text or clicks on the clear button, the courses displayed will change correspondingly
      2. Visitor can filter the courselist by selecting from the filters located next to the search input box
         1. When the user selects a certain Major from the dropdown/input select options the course list updates the displayed courses correspondingly
         2. When the user selects a certain Breadth from the dropdown/input select options the course list updates the displayed courses correspondingly
         3. When the user selects a certain Level from the dropdown/input select options the course list updates the displayed courses correspondingly
      3. Visitor can change the number of courses displayed per page
         1. The dropdown located at the top right section of the courselist provides a list of available number of courses to be displayed.
         2. Selecting a number will display that amount of courses to the course list
   3. Visitor can search a course by clicking on the “Search Button”
      1. When the Search Button is clicked, a ***search window*** will appear.
      2. Visitor can type in the course name, and enter the input by press “Enter”
         1. When typing, there should be a course list appearing under the ***search window***, showing the most relevant courses based on the visitor’s input.
      3. After entering the desired course, a ***“Course Info” window*** should appear showing the course’s name and logo, GPA Graph, and the most liked review.
         1. Visitor can close the ***“Course Info” window*** by clicking the “Close” button on the top right of the window.
   4. Visitor can click on the Header’s “Malaxy” button, which will refresh the page.
   5. Visitor can click on the “Log in” button to sign in
      1. When clicking on the “Log in” button, a “***Log in” window*** should appear allowing the visitor to type in their email address (Username) and password, and Log in by clicking on the “Log in” button.
      2. There should be a “Forgot password?” button that can navigate to a **“Forgot password” page.**
         1. The **“Forgot password” page** allow the visitor to reset password by entering the Email Address and the Verification Code.
      3. There should be a “Sign up” button allow new user to sign up.
         1. When clicking on the “Sign up” button, the ***“Sign up” window*** should appear and allow visitor to enter Email Address and Verification code to sign up.
2. **Logged in User**
   1. After sign up, the first-time user will be navigated to a **“Select Course” page**.
      1. In the “Select Course” page, there should be a search bar on the top of the page, a course list under the search bar.
      2. User can add a course by type in the course name and click on the “add” button on the end of the search bar.
         1. When typing, there should be a course list appearing under the search bar, showing the most relevant courses based on the visitor’s input.
      3. When a course is added, the added course will appear on the course list as a course item.
      4. User can delete a selected course by clicking the “delete” button on the end of the course item.
      5. The “Complete” button at the bottom of the course list allow user to complete selecting the courses.
   2. After course selection, user will be navigated to the **Main page** (Old users will also be navigated to this page after logged in).
      1. Visitor can see the selected courses, a “Search Button”, a “Malaxy” button, and a ”User Profile” button.
         1. User can click on one of the selected courses and a ***“Course Info” window*** should appear
            1. In the ***“Course Info” window***, user can see the course’s name and logo, GPA Graph, a review list ranked by number of likes, and a text bar.

In the review list, there are several reviews listed.

Each review shows the reviewer’s profile image, the texts, and a “Like” button with number of likes showing under.

User can click the “Like” button to like the review and unlike it by clicking again.

There is a number list at the bottom of the review list indicating the page of the review list. User can click “next” to go to next page to see more reviews.

The text bar allows user to type in review and submit the review by clicking the “Submit” button.

After submitting the review, user can see his/her review on the top of the review list.

There is a “Delete” button on the user’s review allowing user to delete.

User can close the window and return to the main page by clicking the “Close” button on the ***“Course Info” window***.

* + - 1. Visitor can search a course by clicking on the “Search Button”
         1. The searching features are specified in 1.b.i. & 1.b.ii.
         2. After entering the desired course, a ***“Course Info” window*** should appear, allowing the user to write a review and see other course’s information.
         3. When user submit a review to a new course (not included in user’s course selection list), that course will automatically added to the user’s course list, and showing it on the **Main Page.**
      2. User can return and reload the Main Page by clicking the “Malaxy” button.
      3. User can modify profile information by clicking the “Profile” button.
         1. After clicking the “Profile” button, user will be navigated to the **Profile page**.

In the **Profile page**, user can see a profile image, a   
“Change password” button, a “Courses” button, and a “Log out” button.

The “Change password” button will navigate to a **Change Password page**, user can change password by first enter Email address, verification code, and new password.

The “Courses” button will navigate user to the **“Select Course” page,** specified in 2.a.

The “Log out” button will log out user account and navigate user to the **Front Page,** specified in 1.a

**Use Cases**

For Visitors and Logged in Users:

| Name | Log In to Malaxy |
| --- | --- |
| Actor | User with account |
| Triggers | Click on the login button |
| Events | User enters the wisc email and password for their account.  If invalid there will be a notification popup alerting the user. |
| Exit Condition | If the backend is able to verify that the user account is valid |
| Post Conditions | The user will be routed to the course search page |
| Acceptance Test | When logged in with incorrect credentials, the user is notified through a popup.  When logged in with credentials, user is routed to the search page. |

| Name | Sign up to Malaxy |
| --- | --- |
| Actor | Visitor |
| Triggers | Click on the signup button on the header of the page  Click on the signup button on the login page |
| Events | User enters a wisc email and the password they want to set.  \* Major selection may be added in the future |
| Exit Condition | The Email the user input is a wisc email and isn’t already a member  The password passes the preset specifications shown to the visitor |
| Post Conditions | When the wisc email is verified and account is created user is routed to the select page |
| Acceptance Test | The email address ends with @wisc.edu  Email address is not already a user  Password is at least 8 characters including a capitol and a special character |

| Name | Log out from Malaxy |
| --- | --- |
| Actor | Logged in User |
| Triggers | Log out button is selected from the profile dropdown in the header |
| Events | User opens the dropdown by clicking on the profile then clicks on the logout button |
| Exit Condition | The login token is removed |
| Post Conditions | User is routed to the search page but without the logged in info |
| Acceptance Test | Token is deleted.  User is no longer able to access the discussion posts |

| Name | Search Course |
| --- | --- |
| Actor | Visitors or Logged in Users |
| Triggers | Click on the “Search” button |
| Events | A search box exists in the courselist page that the user can click on to type in the course name they want to access |
| Exit Condition | User is allowed to view the course list page regardless of user credentials. The search value can be reset through erasing the input or clicking on the x to clear the input |
| Post Conditions | Courselist is updated to display only the courses containing the searched phrase |
| Acceptance Test | The courses being displayed in the course list contains only the classes that have the searched text contained in the course name |

| Name | Filter Course By Breadth |
| --- | --- |
| Actor | Visitors or Logged in Users |
| Triggers | Click on the Breadth Select filter |
| Events | A filter for selecting a breadth filter exists in the courselist page that the user can click on to filter the course list by the breadth they want to access |
| Exit Condition | User is allowed to view the course list page regardless of user credentials. All can be selected to remove the filtering by breadth |
| Post Conditions | Course list is updated to display only the courses corresponding to the breadth selected |
| Acceptance Test | The courses being displayed in the course list contains only the classes that are of the selected breadth |

| Name | Filter Course By Major |
| --- | --- |
| Actor | Visitors or Logged in Users |
| Triggers | Click on the Major Select filter |
| Events | A filter for selecting a major filter exists in the courselist page that the user can click on to filter the course list by the major they want to access |
| Exit Condition | User is allowed to view the course list page regardless of user credentials. All can be selected to remove the filtering by major |
| Post Conditions | Course list is updated to display only the courses corresponding to the major selected |
| Acceptance Test | The courses being displayed in the course list contains only the classes that are of the selected major |

| Name | Filter Course By Level |
| --- | --- |
| Actor | Visitors or Logged in Users |
| Triggers | Click on the LevelSelect filter |
| Events | A filter for selecting a level filter exists in the courselist page that the user can click on to filter the course list by the level they want to access |
| Exit Condition | User is allowed to view the course list page regardless of user credentials. All can be selected to remove the filtering by level |
| Post Conditions | Course list is updated to display only the courses corresponding to the level selected |
| Acceptance Test | The courses being displayed in the course list contains only the classes that are of the selected level |

| Name | Change how many courses are displayed per page |
| --- | --- |
| Actor | Visitors or Logged in Users |
| Triggers | Click on the Course per page dropdown |
| Events | A dropdown for the allowed number of courses displayed per page is clicked. Then the user chooses how many courses they want displayed per page |
| Exit Condition | Exit condition doesnt apply for this user story |
| Post Conditions | The number of courses that is displayed in a singe page increases or decreases to the selected number. |
| Acceptance Test | The default setting is 20 courses per page. Once a larger or smaller number is selected, the courses displayed in the courselist changes correspondingly. |

| Name | Add Discussion post |
| --- | --- |
| Actor | Logged in User |
| Triggers | User clicks on the reply button / tab and enter |
| Events | The user replies to a discussion post or writes a new post then posts. |
| Exit Condition | The user has added text that passes our requirements and triggers the reply button  Only users logged in are shown the discussion board so authentication not required. |
| Post Conditions | The post is added to the discussion board |
| Acceptance Test | The text that the user has written is longer than 10 characters and is not a repetition of the same characters. |

| Name | Add Course |
| --- | --- |
| Actor | Logged in User |
| Triggers | Signing up will route user to the select course page.  Manage my courses button will be available in the profile page which will also route the user to the select course page.  User then selects their course from the list of courses utilizing the search bar. There will be a button to add the course. |
| Events | User searches their course and clicks on the add course button |
| Exit Condition | Use clicks on the add course button |
| Post Conditions | Course is added to the user’s course list |
| Acceptance Test | The User is able to view their added courses through their profile page and can be accessed in future logins. |

| Name | Remove Course |
| --- | --- |
| Actor | Logged in User |
| Triggers | Manage my courses button will be available in the profile page which will route the user to the Manage my courses page. |
| Events | User views their added courses and clicks on the remove course button |
| Exit Condition | Use clicks on the remove course button |
| Post Conditions | Course is removed from the user’s course list |
| Acceptance Test | The list of courses retrieved for the User is the same except for the removed course. |

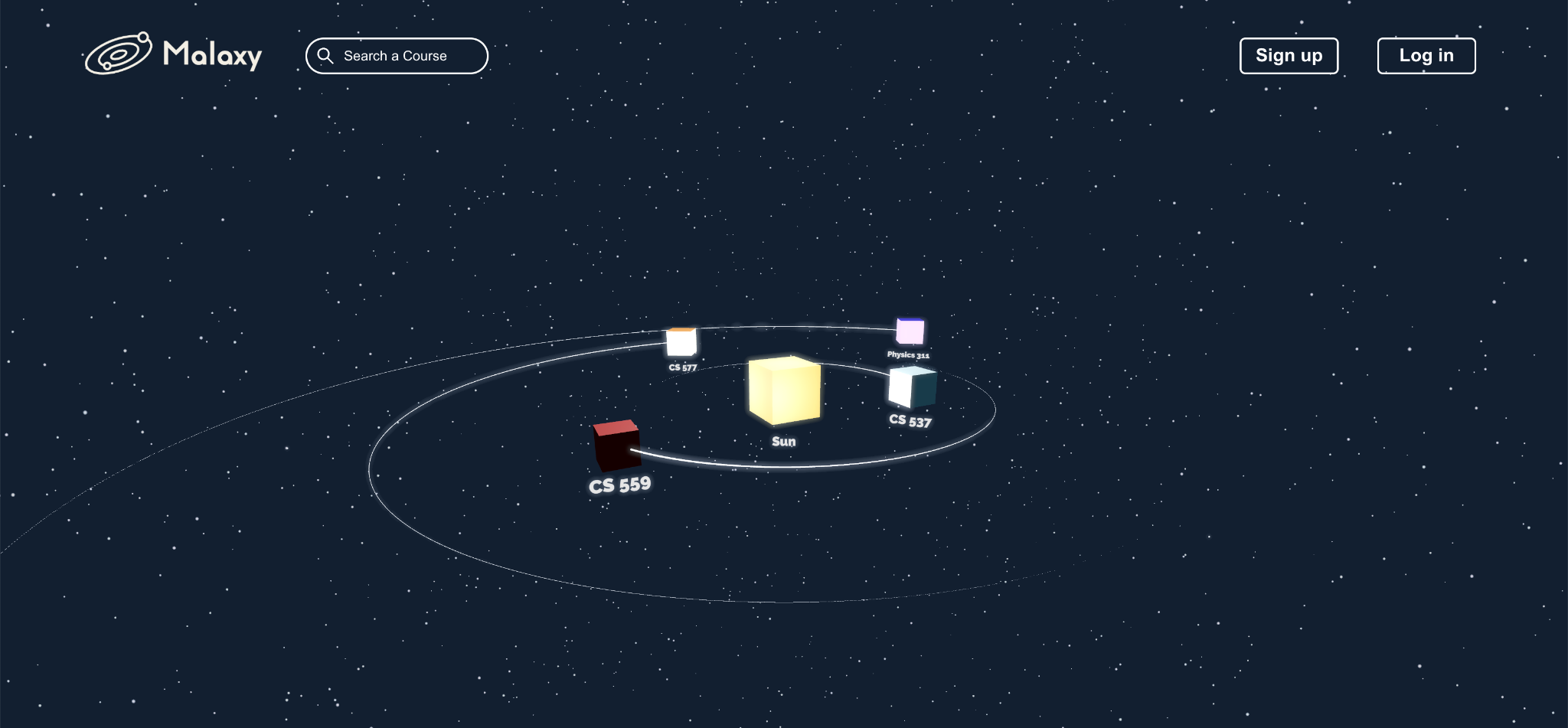
| Name | Delete account |
| --- | --- |
| Actor | Logged in User |
| Triggers | The delete account button will be available in the profile page which will route the user to the search courses page in the logges out state |
| Events | User clicks on the delete account button in the profile page |
| Exit Condition | User clicks on the delete verification modal that will popup once the delete account button is clicked |
| Post Conditions | User information is removed from the server and will not be able to log in using those credentials. |
| Acceptance Test | The user types in “delete account” in the input box of the deletion modal.  User info is deleted from the server.  The comments that the user has created will not be removed as it is anonymous. |

**User Interface Requirements**

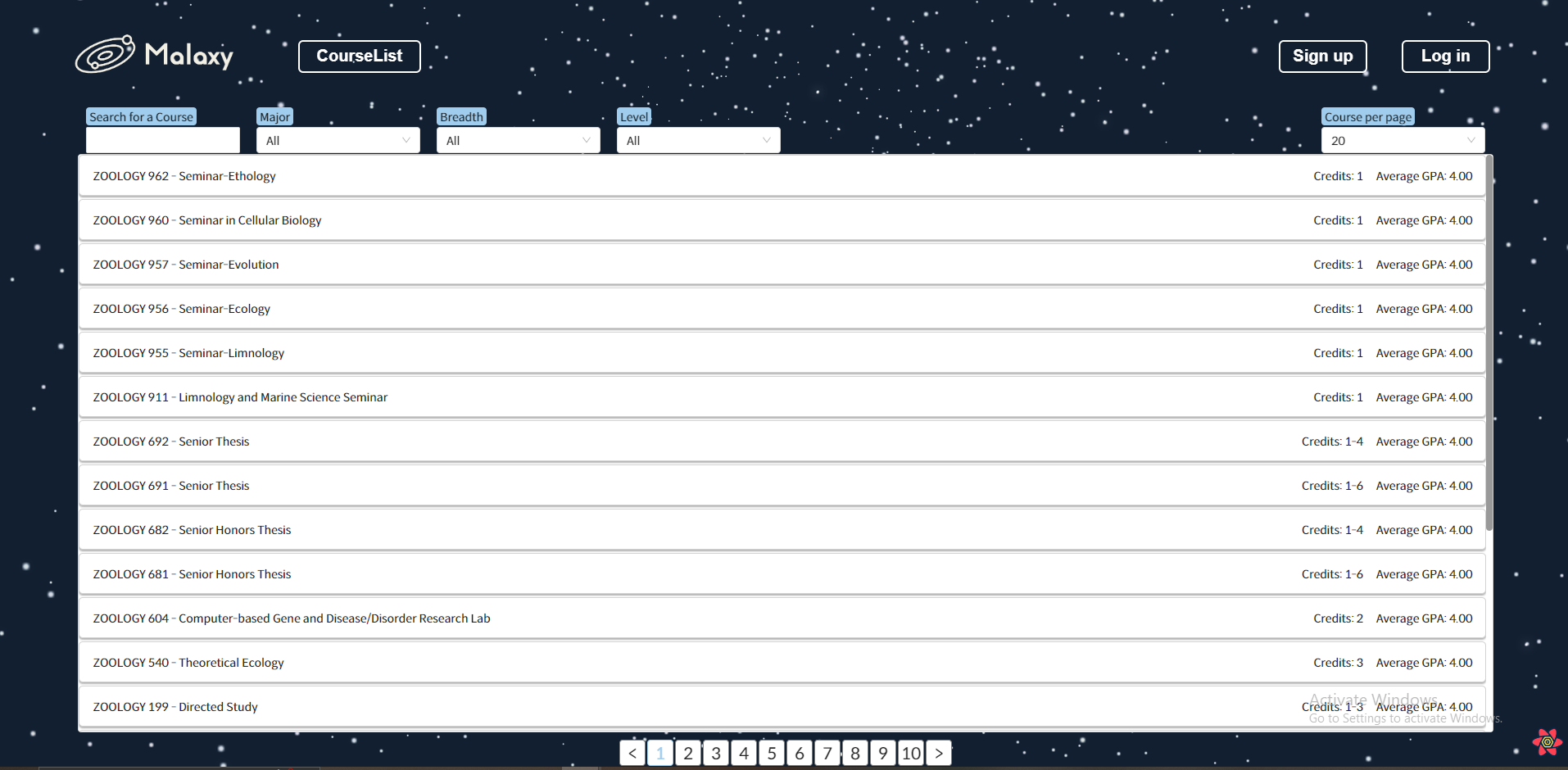
**Page Structure**

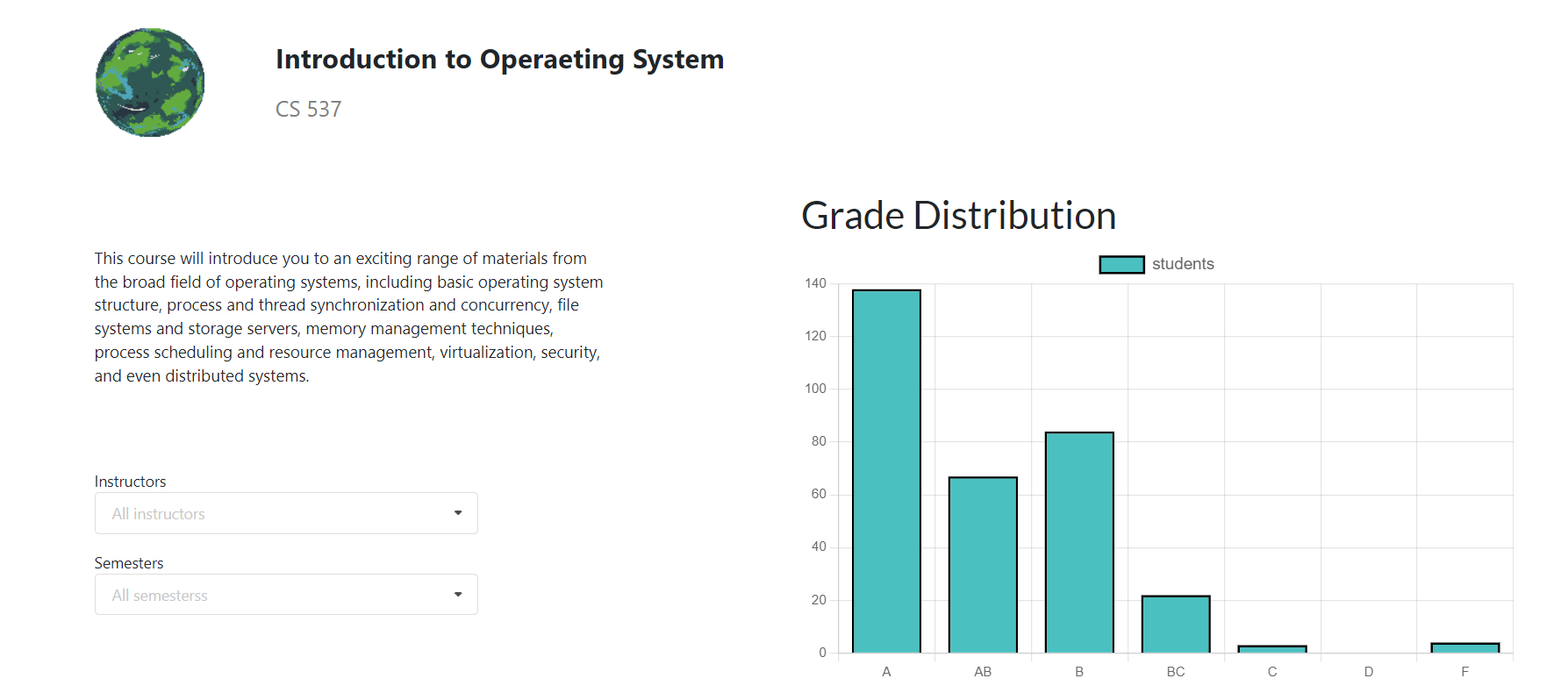
1. **Front Page (Visitor)**
   1. Search window
   2. Course Info window
   3. Log in window
      1. **Reset Password Page**
   4. Sign up window
2. **Select Course Page**
3. **Main Page (Logged in User)**
   1. Search window
   2. **Course Info Page**
   3. **Profile Info Page**
      1. **Change Password Page**

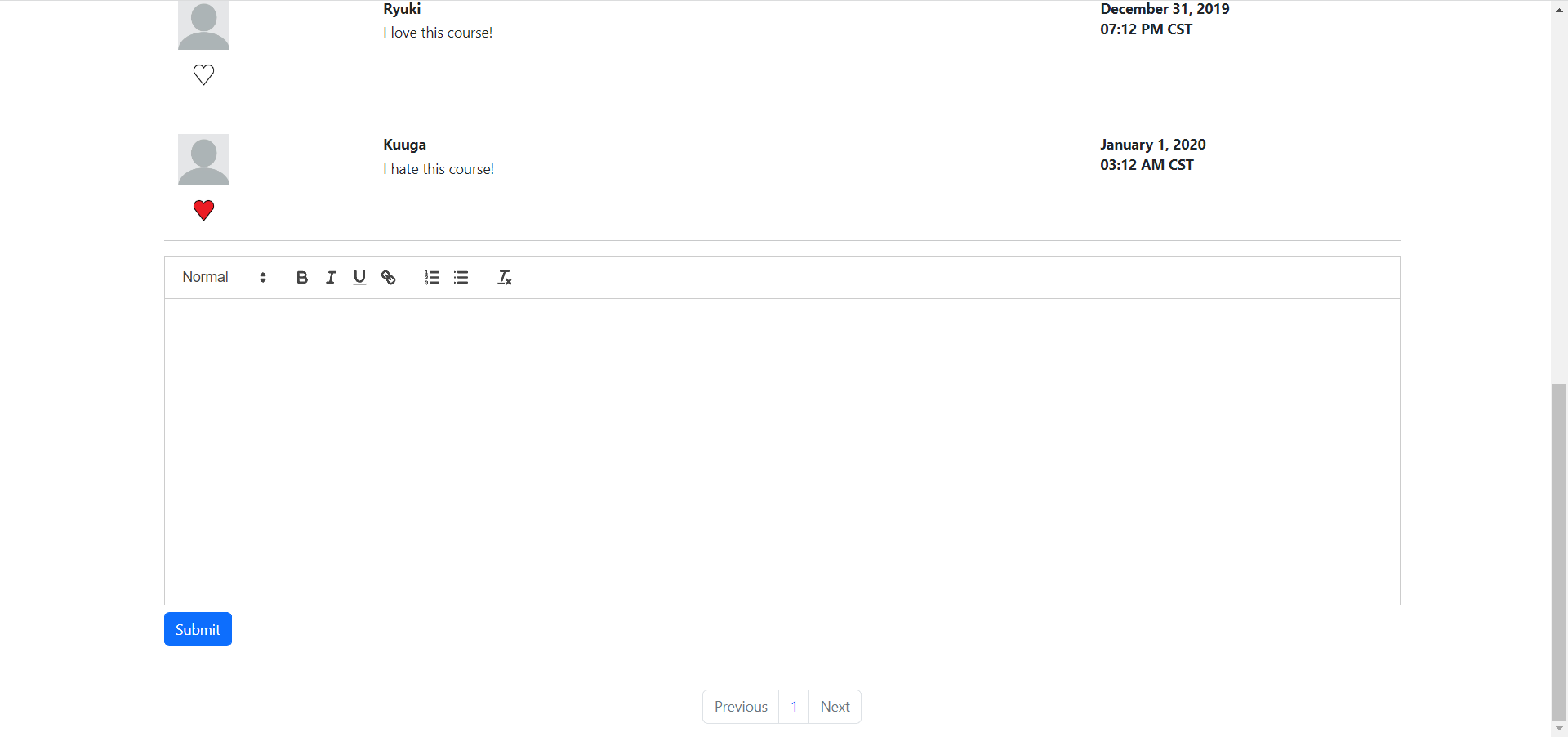
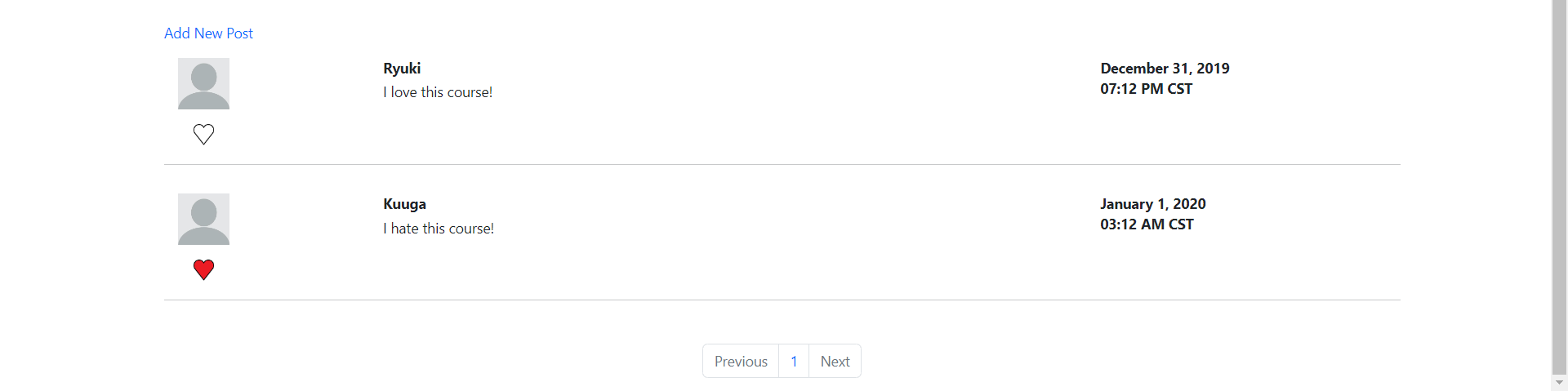
**Visitor Page:**

****

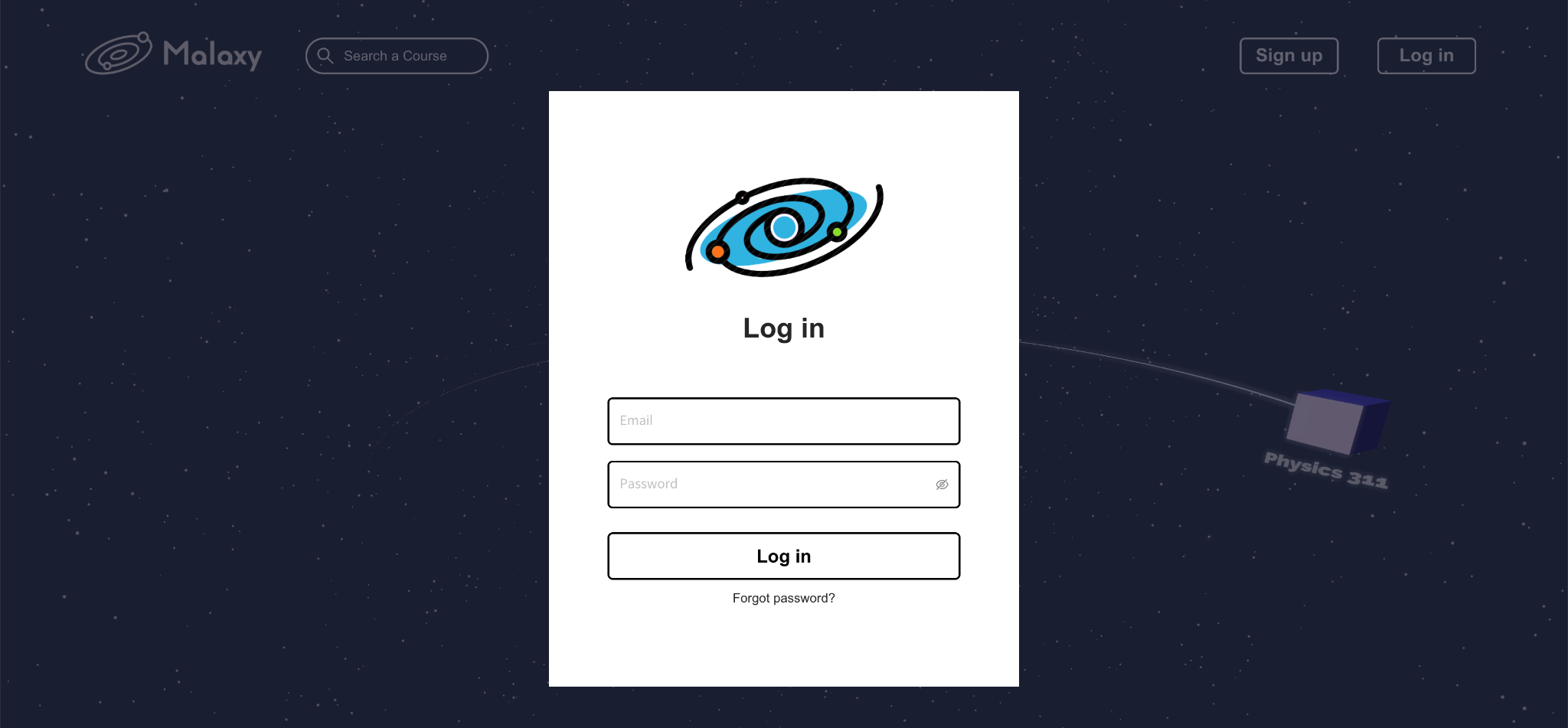
**Course List Page**

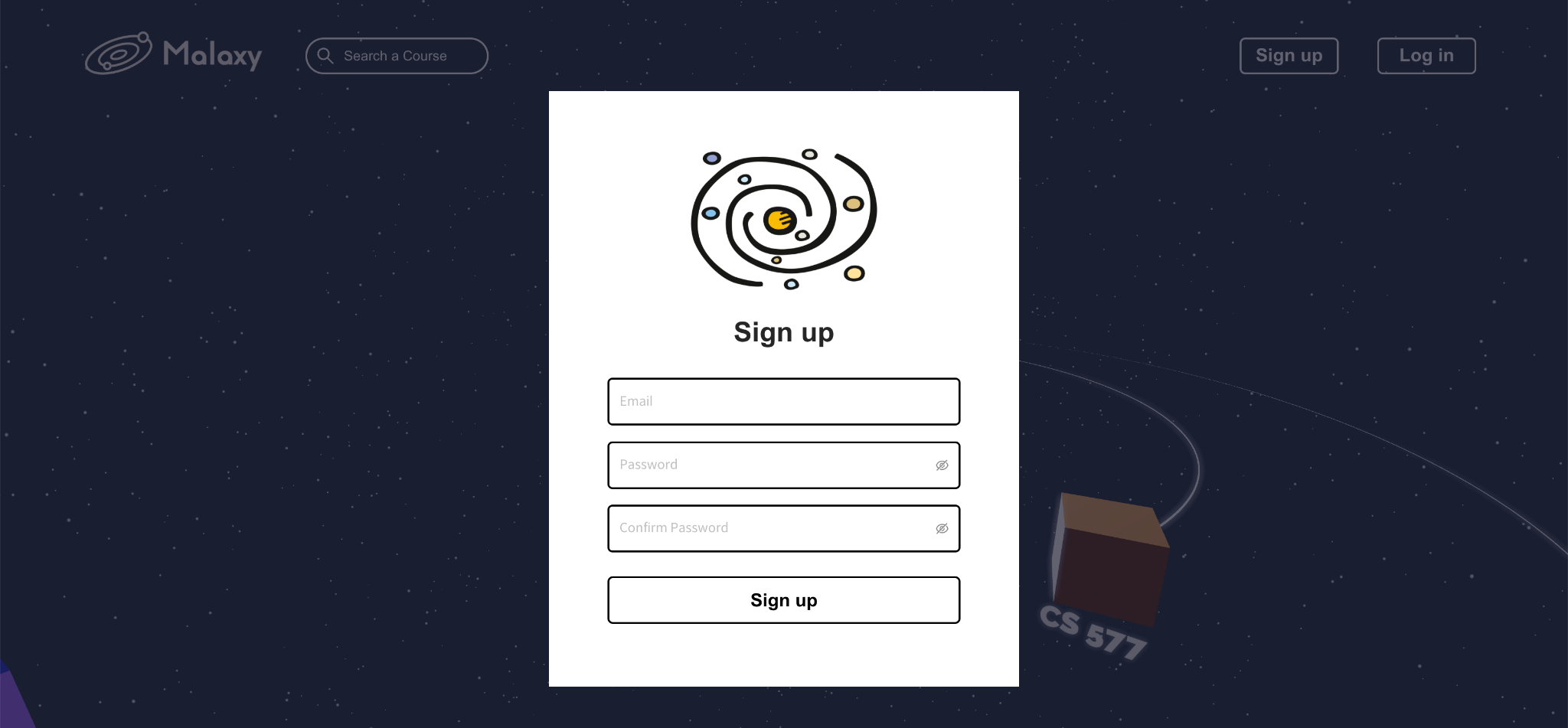
****

**Course Info Page**

****

**Log in and Sign up Page**

****

****

**Security Requirements**

We aim to handle these security concerns:

1. Injection

When untrusted data is sent to the parser as part of a command or query, injection defects such as SQL injection, NoSQL injection, OS injection, and LDAP injection can occur. Malicious data from an attacker can trick the parser into executing non-predated commands or accessing data without proper authorization.

1. Broken Authentication

Often, by misusing an application's authentication and session management capabilities, an attacker can decipher a password, key, or session token, or exploit other development flaws to temporarily or permanently impersonate another user's identity.

1. Sensitive Data Exposure

Many Web applications and apis fail to properly protect sensitive data, such as financial data, medical data, and PII data. An attacker can commit credit card fraud, identity theft, or other criminal acts by stealing or modifying unencrypted data. Unencrypted sensitive data is vulnerable to damage, so we need to encrypt sensitive data, including data in transit, data stored, and browser interaction data

1. [Broken Access Control](https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A5-Broken_Access_Control)

Appropriate access controls are not in place for authenticated users. An attacker can exploit these flaws to gain access to unauthorized features or data, such as: accessing another user's account, viewing sensitive files, modifying another user's data, changing access permissions, etc.

**System Requirements**

**Tools/Frameworks**

* Django
  + Python / Django Rest Framework
* React
  + Javascript / CSS / HTML (JSX)
* MySQL, SQLite

**Performance Requirements**

* None; MySQL provides the basic needs for free

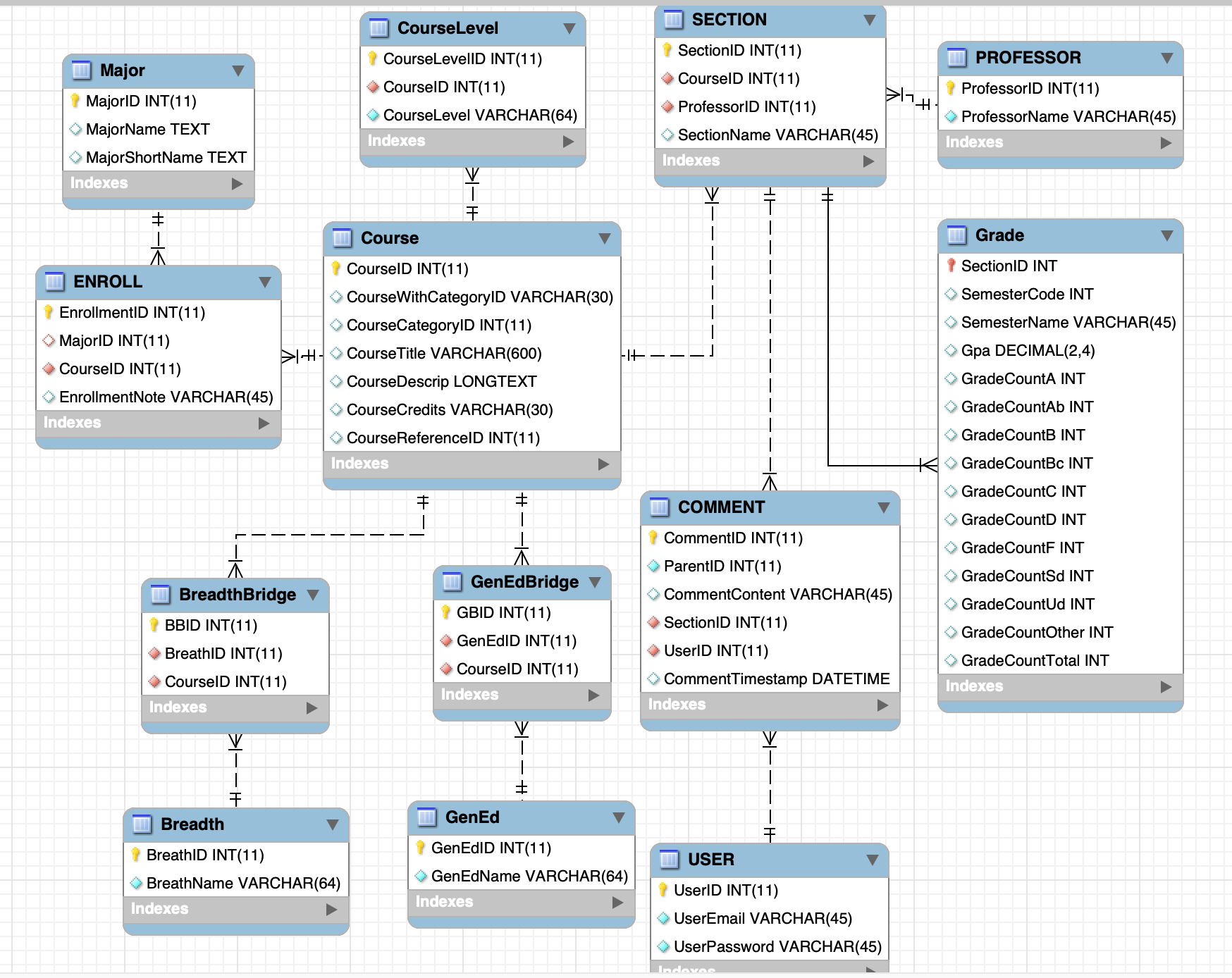
**Data Capacity and Memory Requirements**

* Ischool provide free server for us; MySQL provides the basic needs for free

**User Requirements**

* A modern web browser (e.g. Chrome, Edge, Firefox)
* Internet Connection

**Database Schemas**

****