TYPEit

Welcome to the TYPEit Web App! This project is packaged as a Flask Application using SQLite as a database engine. This is an interactive quiz aimed at high school students aged between 13-17 with a variety of different subjects and a range of difficulty to offer a broad spectrum of challenge to all users.

This app allows for the login of both administrator and student type accounts and also the registration for new students as the class grows. Admins may control who can log into the quiz and also what subjects are available for the students to attempt. The students may then choose which of the available subjects they would like to try, which each have a corresponding question set.

During the quiz, users will be presented with a question and a clue as to what letter the answer starts with just to get them going. It is then up to the user to input the correct word which will award points based on how many you get correct. As the user inputs the word, the line will turn green if the user is on the right path, and turn red if they begin to input wrong characters. Beware, there is a timer that is ever diminishing, if you fail to answer all questions before the timer runs out, your quiz run will end! If you get stuck on a word, do not worry, the skip button will allow continuation to the next word.

At the student dashboard, there will be a results table that will display the best score attained for each subject. Admins will also be able to see the highest score attained for each subject by the students.

**User Story**

1. The user will be directed to a login page and asked for a username and password. This combination will be validated through the database and the login button will bring the user to the appropriate dashboard.
2. If the user has no existing account, one may be created using the registration page which allows for a custom username and password to be entered. By default, all created user accounts will be given student access only and additional admin access will have to be added manually from the backend.
3. The student dashboard opens up into the topic selection page, with sidebar options allowing the student to view his high scores in each subject as well as the highest score attained by classmates.
4. The student may then choose a topic from the selection available on the choose topics page which will then bring them to the question page.
5. The question page consists of a score/streak counter on the top left, a timer on the top middle of the screen and the question in the centre with an input field pre-filled with a hint to start the student off.
6. As the user types in their attempt, the input field will gradually fill green proportional to how complete the word is. If the letter entered is incorrect, the bar will switch to red, notifying the student that the letter entered is wrong.
7. The user only needs to successfully type the full correct answer to submit the answer and the next question will automatically load with the previous answer appended with a tick on the left hand side.
8. If the user chooses to skip the word, score will not be awarded and the answer will be appended with a cross on the left-hand list.
9. Upon completion, the student will be brought to a results screen showing the quiz outcome and a button to return to the dashboard to pick another topic.

**Admin Story**

1. Admins may login with the default admin login.
2. The admin dashboard will open up automatically starting at the choose topics screen where admins may choose to enable or disable topics.
3. Along the left bar, the manage students tab will bring up a page where admins are able to enable/disable login for students using a checkbox function.
4. Admins may also view the class scores for each topic and the highest score attained overall.

**Technology Used**

* HTML
* Bootstrap
* CSS
* Javascript
* Jquery

**Server Side:**

* Flask
* SQLite
* SQLAlchemy

**Server Routes:**

**Open Routes**

'/' - Home

'/login' - Login Screen

'/logout' - Logout Function (returns you to login screen)

'/register' - Register Screen

**Protected Routes**

'/student - choose test topic' - Main student dashboard

'/student - my scores' - Student results screen

'/select topics' - Admin main dashboard for enabling/disabling topics

'/manage students' - Admin dashboard for enabling/disabling student logins

'/class scores' - Admin dashboard for viewing the highest score achieved for each topic

**Database**

The database contains 3 tables; results, topic and user.

The results table holds the values for all of the attempts by students and groups these values by the id of the student that attempted the quiz and the id of the quiz topic.

User table holds the specific id of all users, login information, their hashed password, privilege details of each user and a Boolean value that denotes whether or not the student account is enabled by the admin.

The topic table holds the unique id of the different topics for the quiz, its name and its associated question sets to be selected by the student.

The database diagram is as follows;

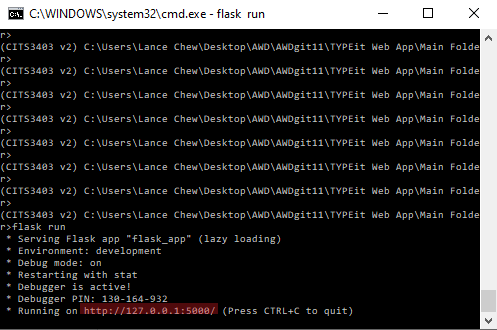


**Modules:**

* **Flask**
  + Flask
  + Flask-login
  + Flask-migrate
  + Flask-sqlalchemy
* **SQL**
  + Alembic
  + SQLite
  + SQLAlchemy
* **Python**
  + Certifi
  + Click
  + Jinja2
  + Markupsafe
  + Pip
  + Python
  + Python-dateutil
  + Python-dotenv
  + Python-editor
  + Setuptools
  + Six
  + Wheel
  + Wincertstore
* **Other** **dependencies**
  + Ca-certificates
  + Itsdangerous
  + Mako
  + Openssl
  + Vc
  + Vs2015\_runtime
  + Werkzeug
  + Zlib

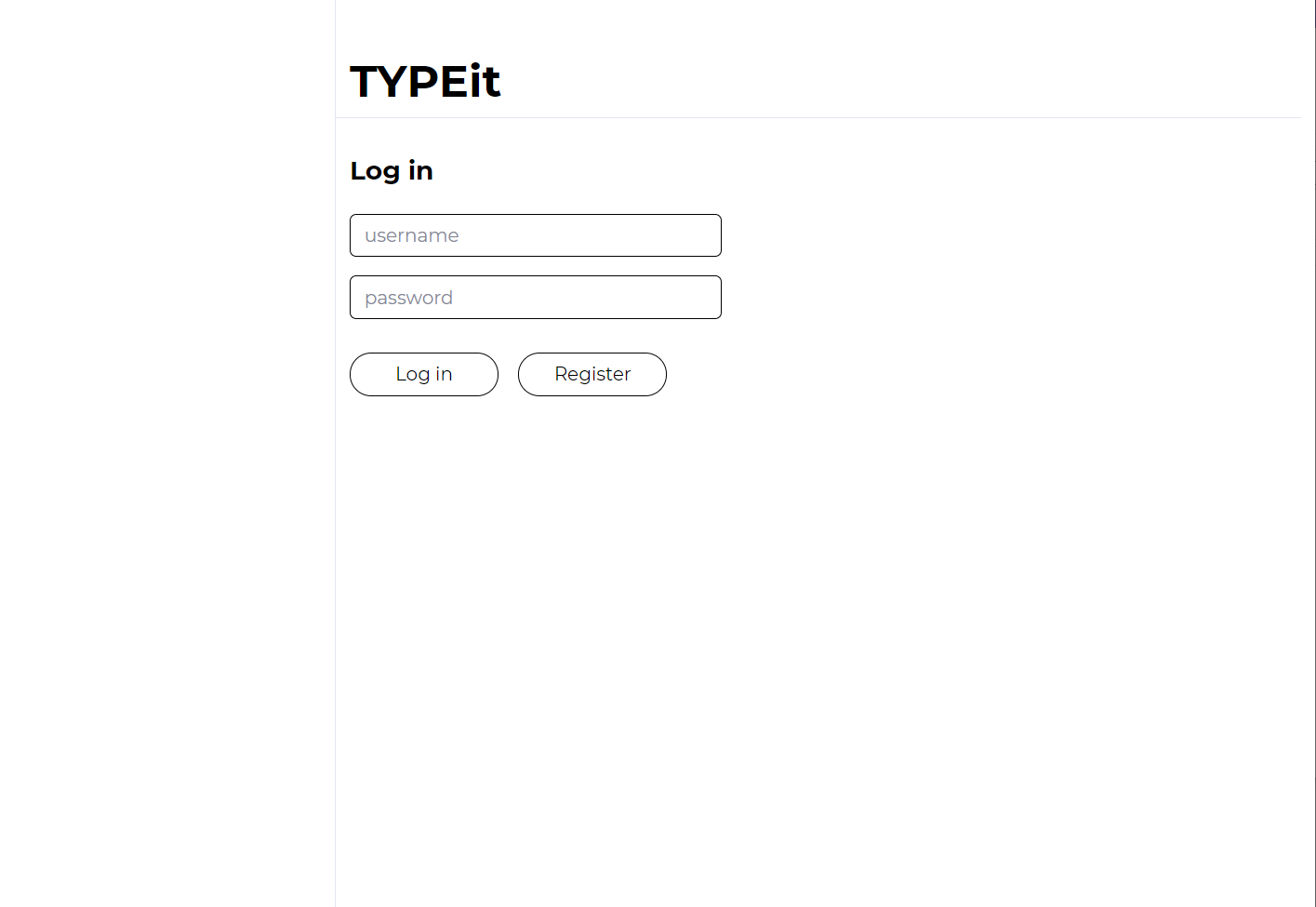
**Build Instructions**

* Setup a new virtual environment
* Copy the repository to the appropriate folder
* Open the console within the virtual environment and navigate to the repository
* Install required dependencies by using requirements.txt
  + Pip install –r requirements.txt
* Enter command
  + Flask run
* Open browser to the local host URL shown in the console.

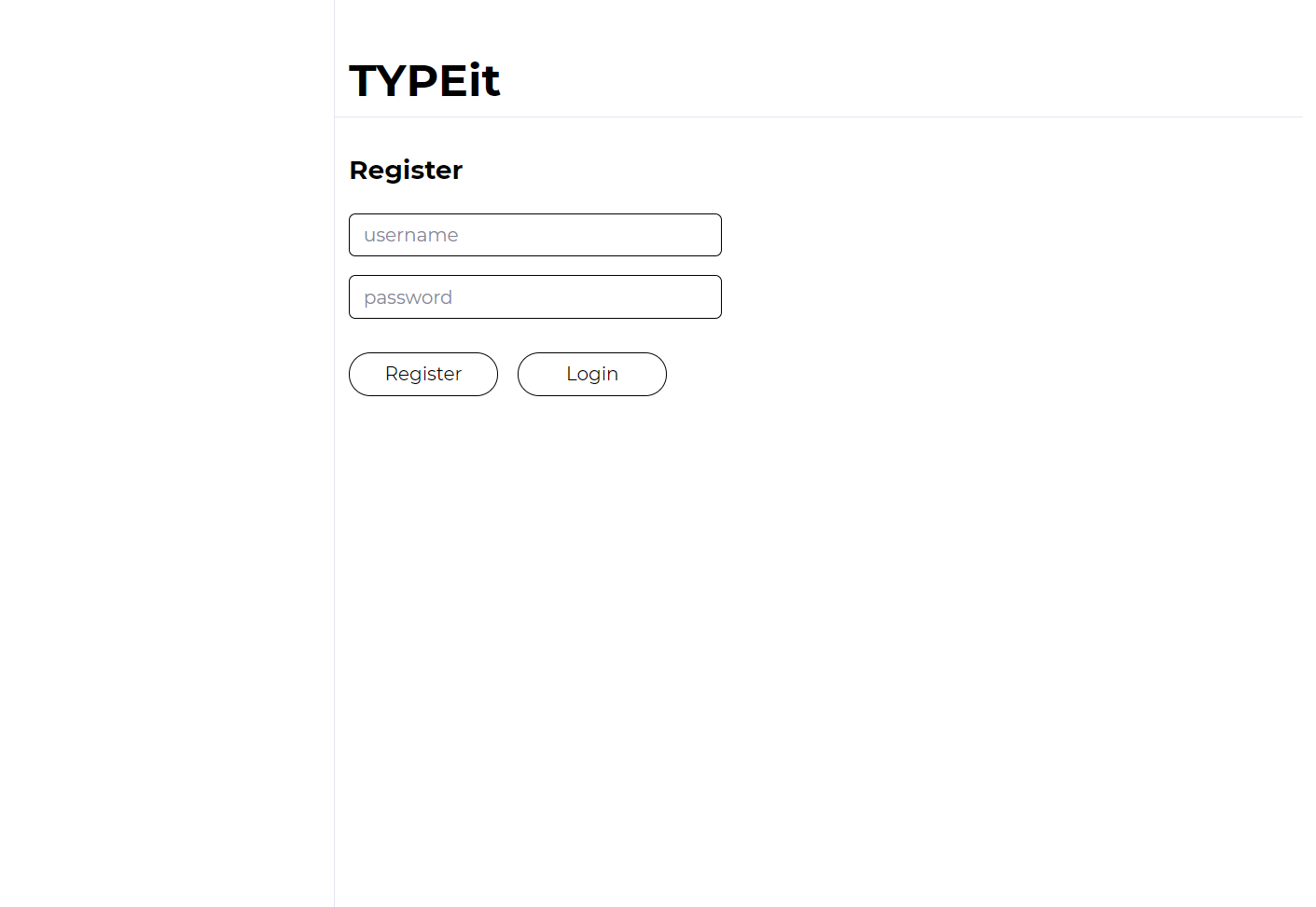


**Screenshots**

Login Screen

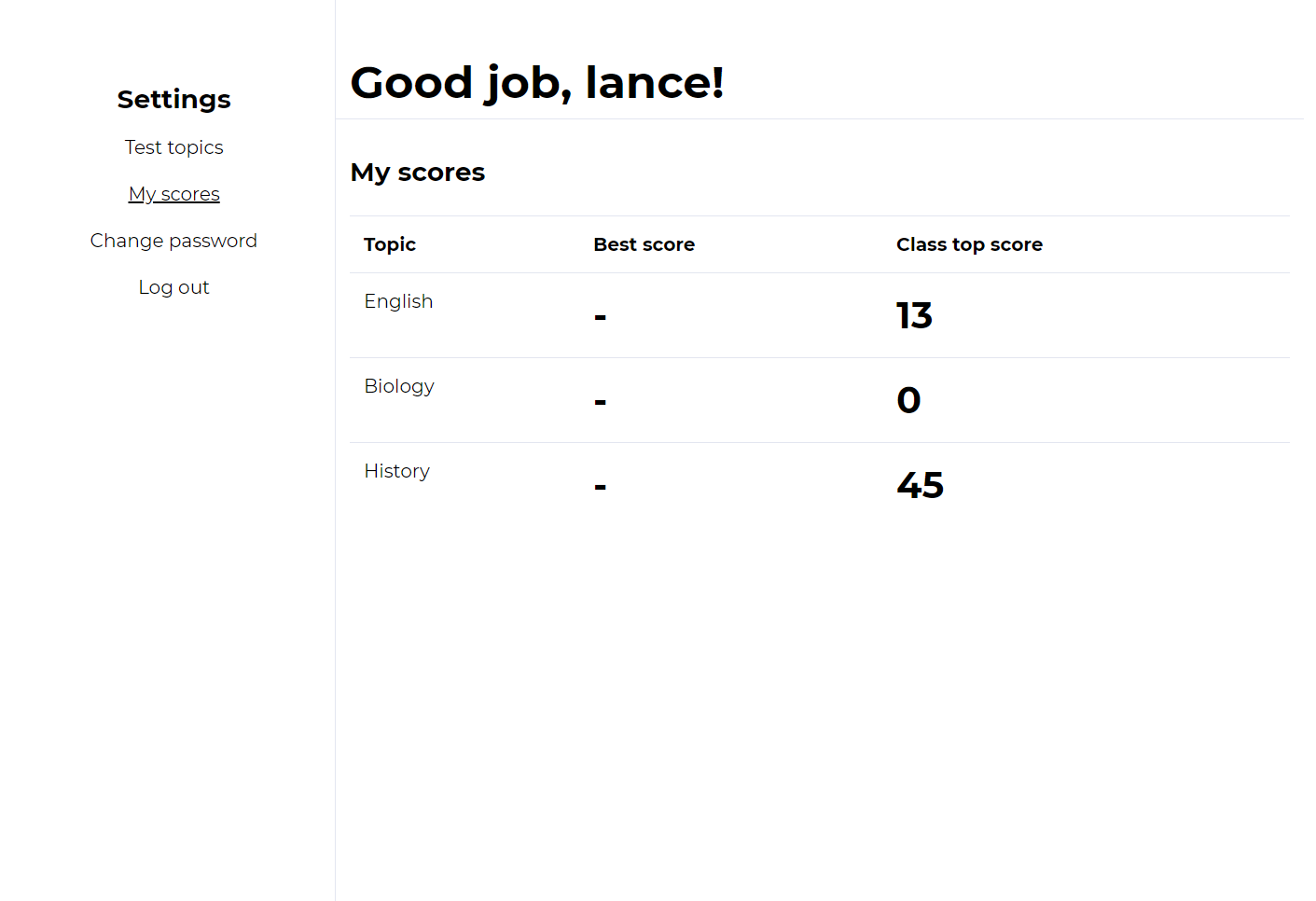
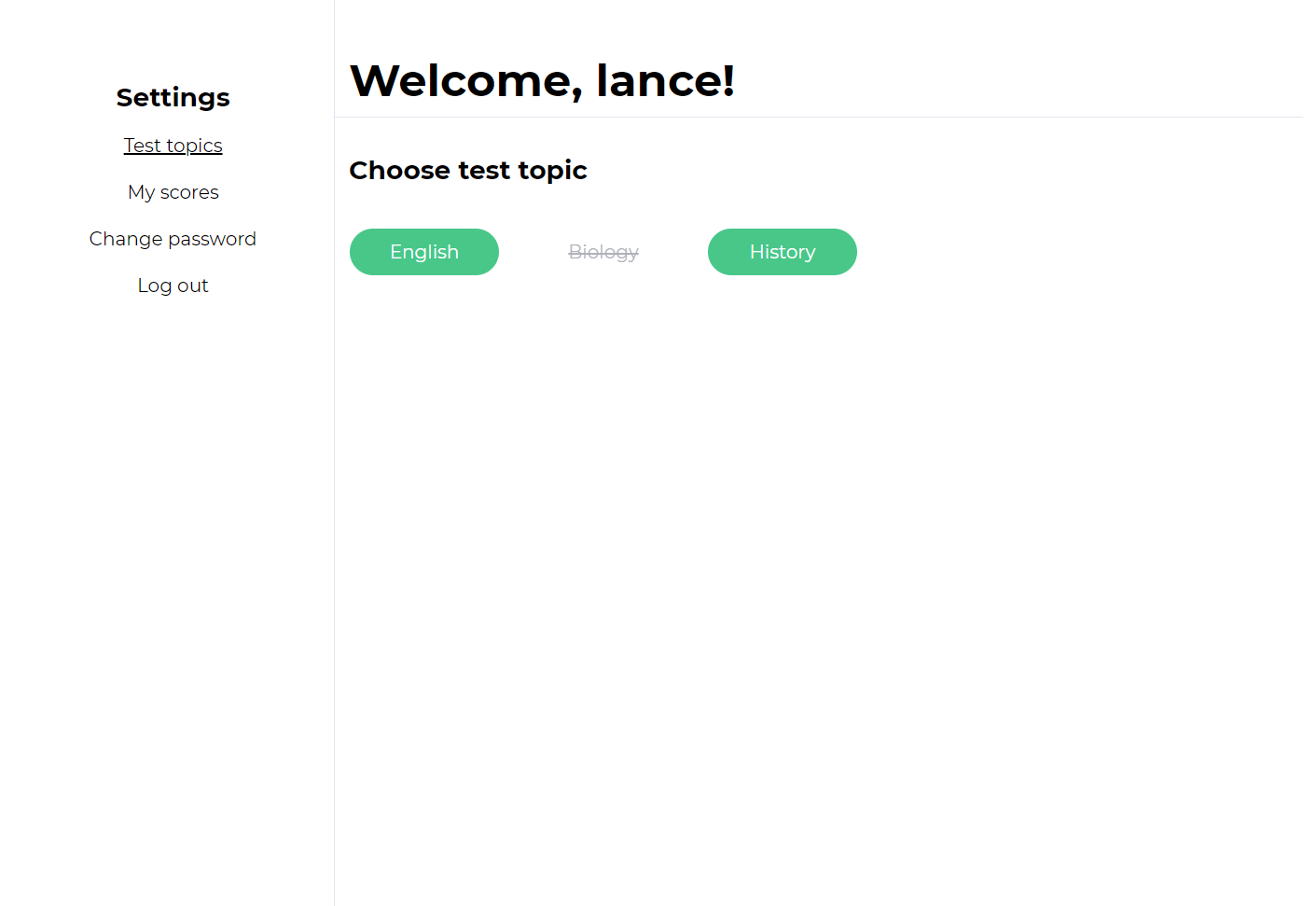


Registration Screen

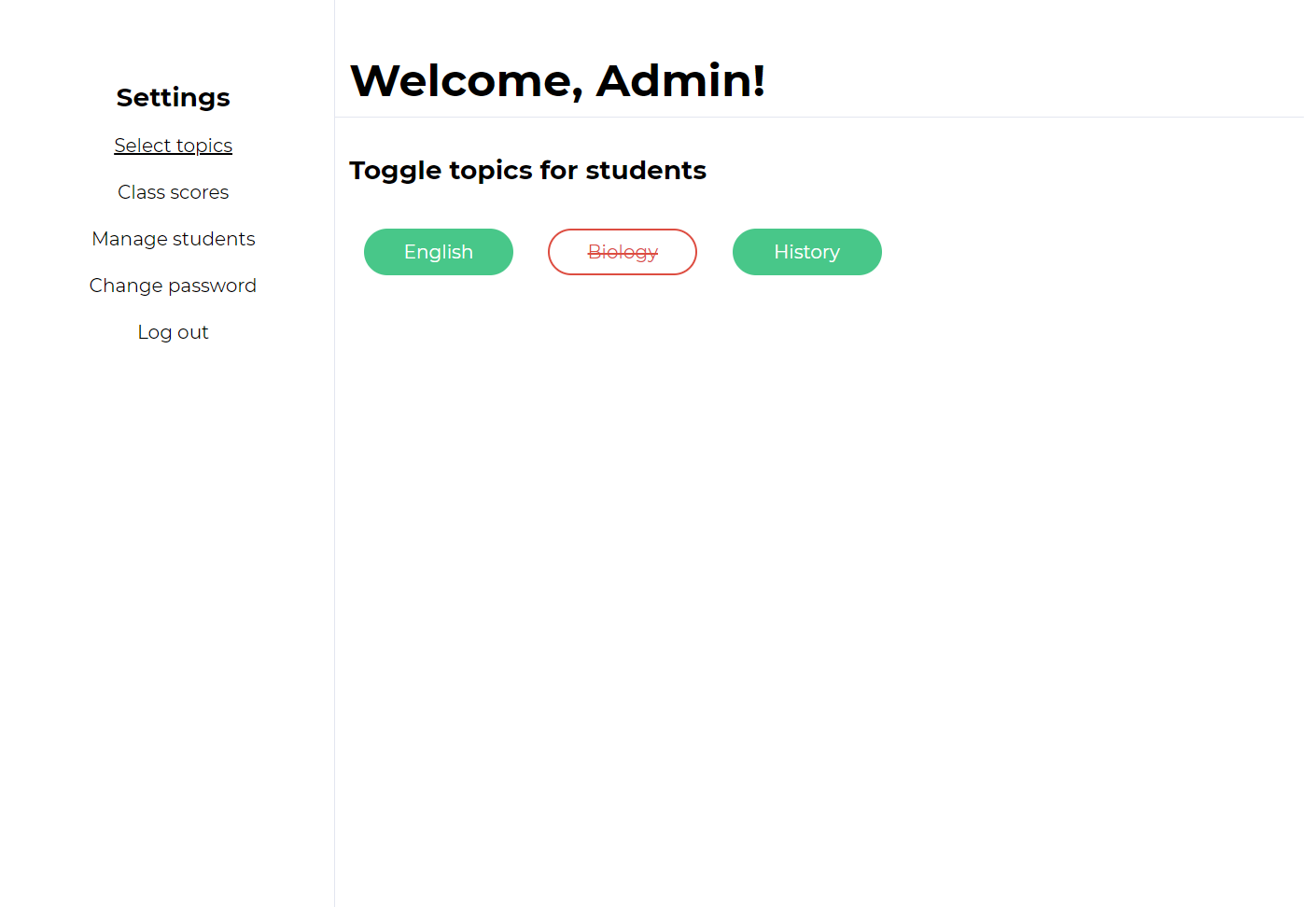
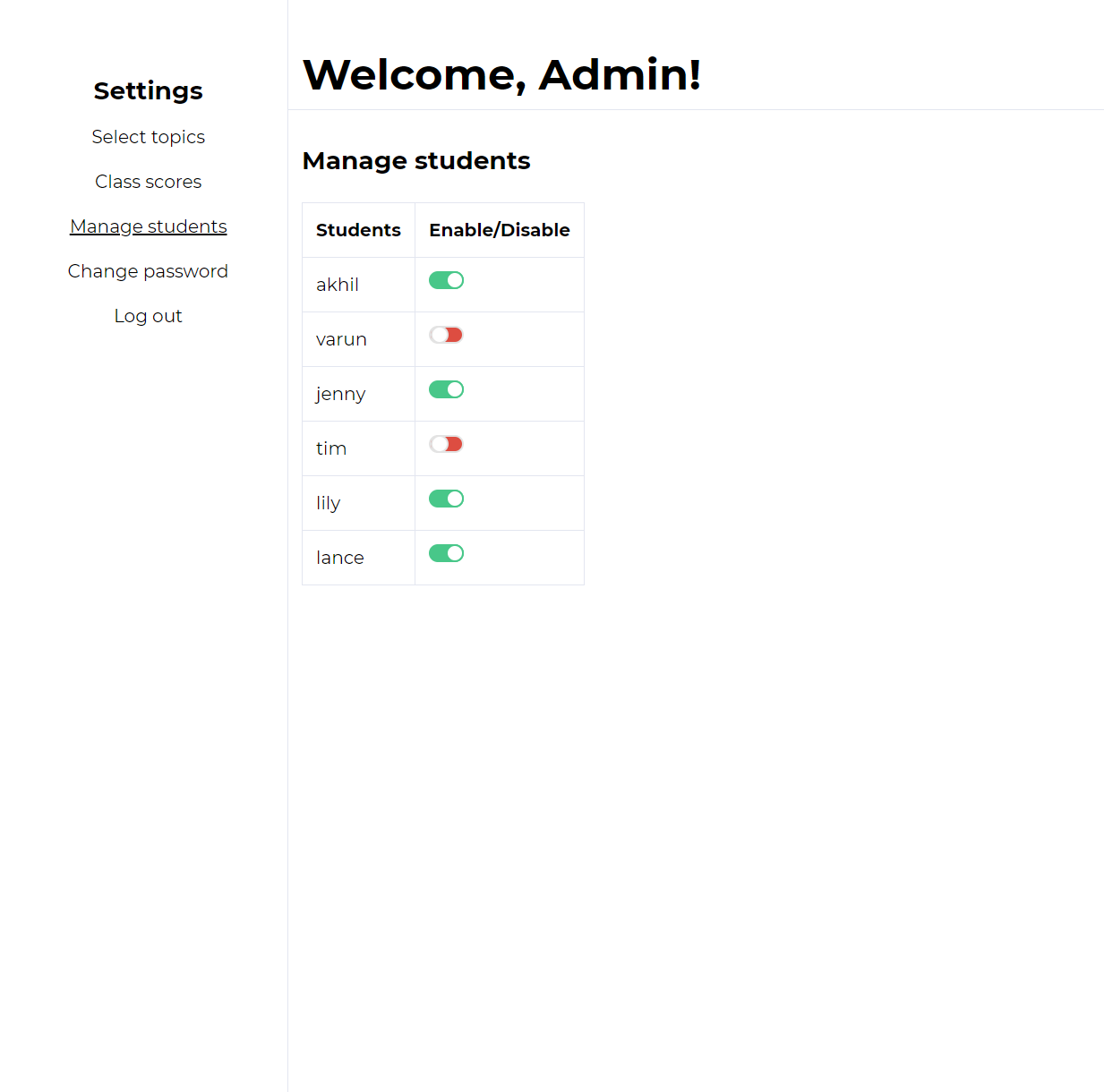


Student Dashboard

Student Scores

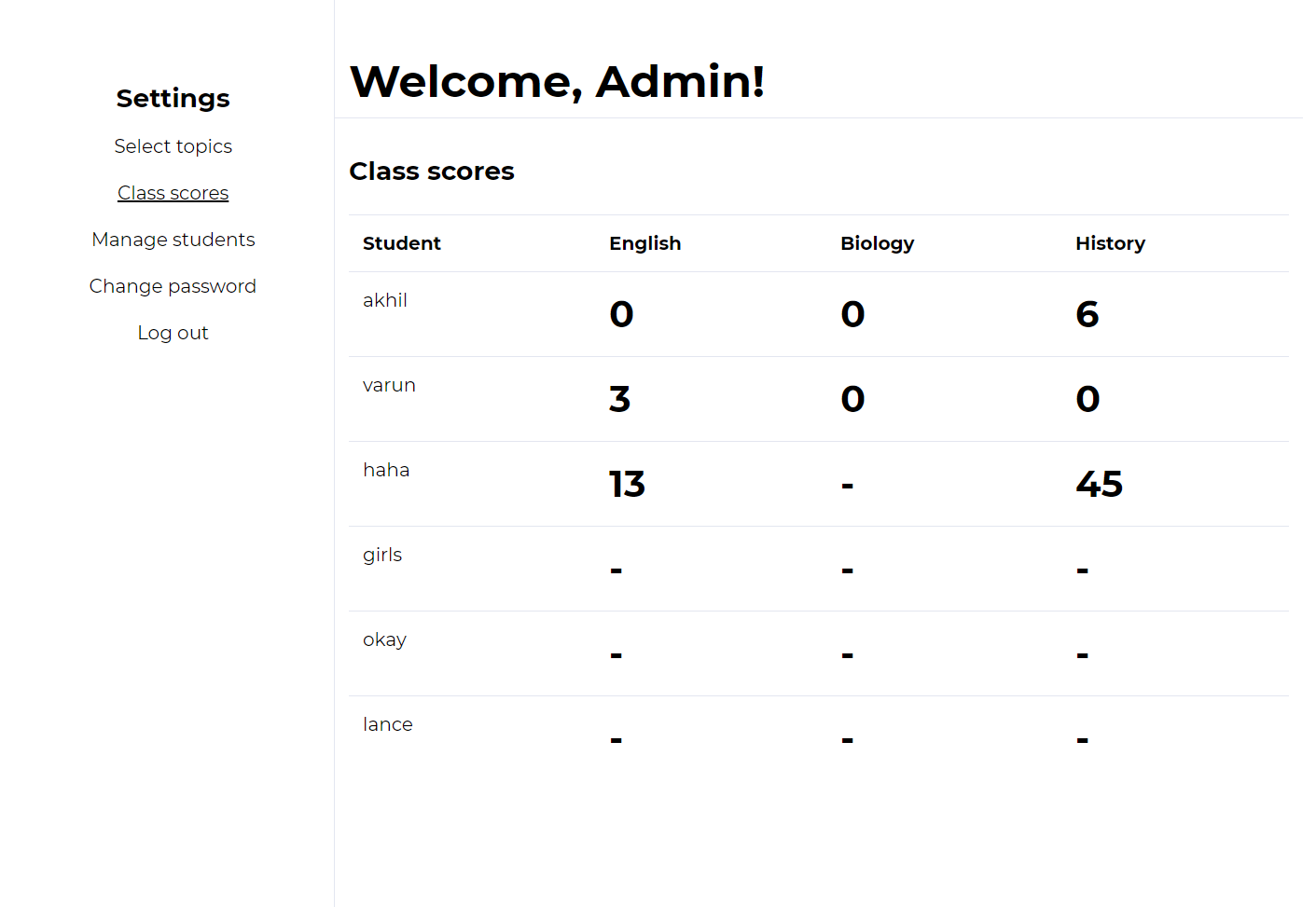


Admin Dashboard

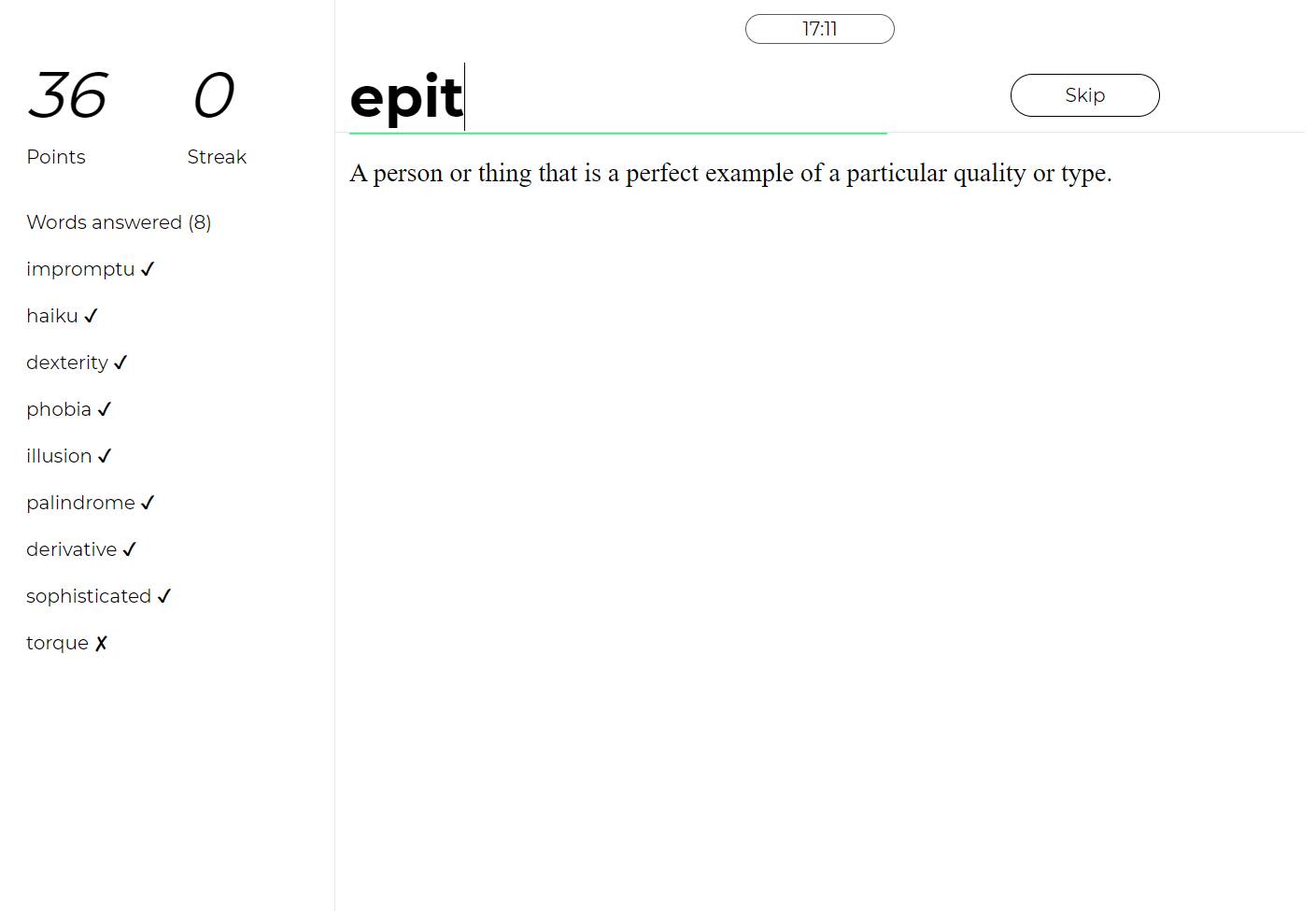


Admin Manage Students

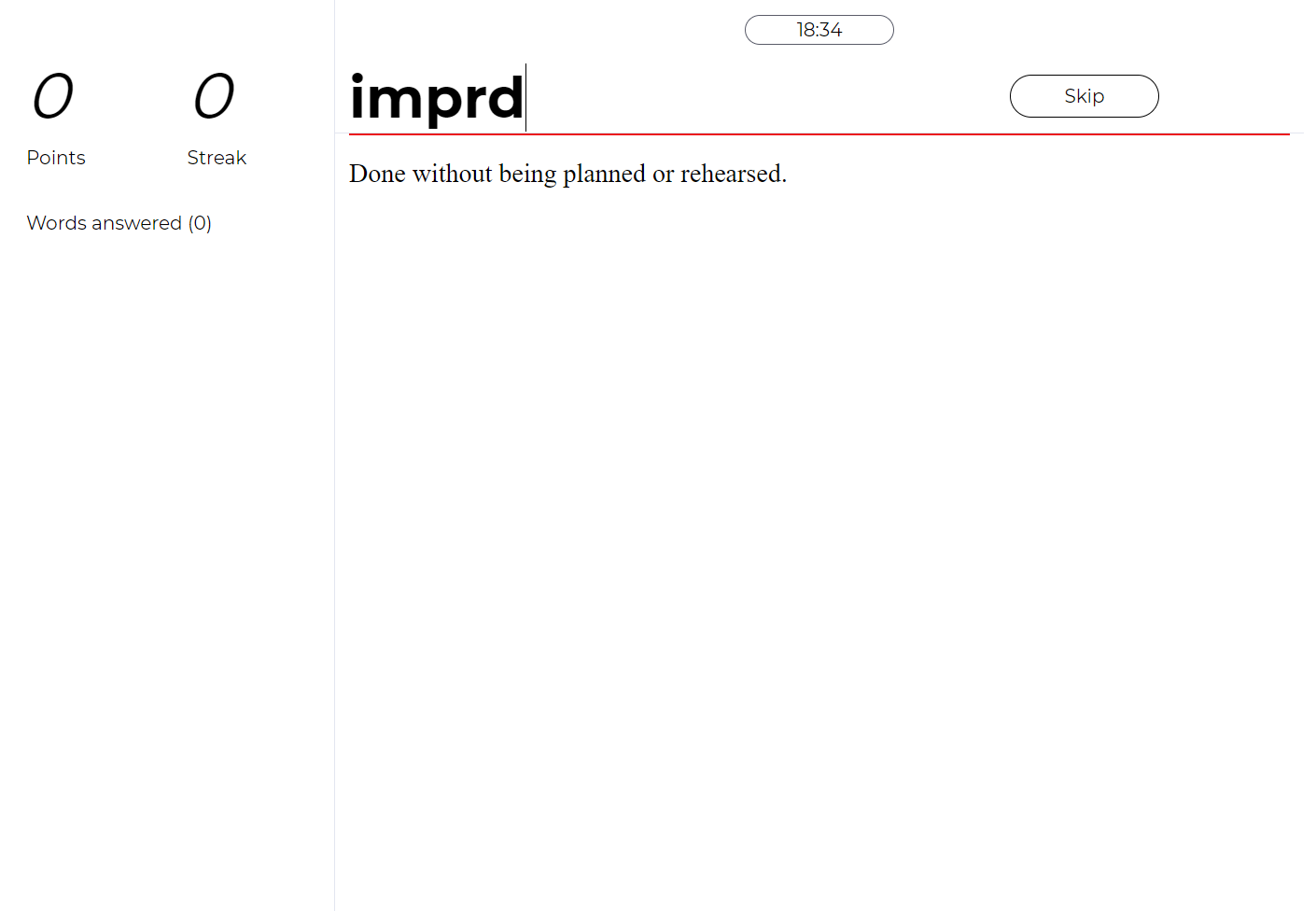
Admin Class Scores



Questions Page



Questions Page with wrong input



Results Page

