
Automated Essay Grader

Kartavya, Aditya, Shreyansh

Nov 27, 2019

BACK END:

1	Project environment setup	3
2	Deployment steps using Docker	5
3	Checkout some code now	7
3.1	MVC architecture connections	7
3.2	Automated essay grader model	8
3.3	Web views	9
3.4	Android application	9
4	Indices and tables	19
	Python Module Index	21
	Index	23

Automated Essay Grader is essentially a usable wrapper for using machine learning models based on Essay grading. Our project runs on Starlette server hosted on Heroku cloud with a noSQL database hosted on Firebase. Using MVC architecture we have created a web interface powered by Bootstrap 4.0. Our project also has a fully featured Android interface.

To check a live web demo [Check out](#)

PROJECT ENVIRONMENT SETUP

To start the project you first need to install all dependencies:

```
>>> pip install requirements.txt
```

After we have all the requirements set up, we will now create an environment

```
>>> conda env create -f Softlab.yml
```

After the environment is set up you can initialize the environment somehow

```
>>> conda activate Softlab
```

Yes that was it!! Now simply start the system by typing

```
>>> python3 main.py
```


DEPLOYMENT STEPS USING DOCKER

- 1) You can simply build the the image file by running following in the directory where Dockerfile is present

```
>>> docker build --tag essay-grader .
```

```
>>> docker run --name essay-grader -p 8000:8000 essay-grader
```


CHECKOUT SOME CODE NOW

3.1 MVC architecture connections

This is a short description of the code belonging to the plumbing between frontend and backend

The code in this file is mainly plumbing between the frontend and ML backend. It consists of routes which map to the browser url. We use the MVC architecture where we code the controller and models. View manager in our code is Jinja2vec

async `main.contrrbPage(request)`

This function gives people the option to contribute the essay set and make the model better.

```
>>> Essay: Hi, this is the festival of diwali and I like to enjoy it with my_
↪ friends and pet cow
```

```
>>> Score: 30/100
```

async `main.evaluate(request)`

The function evaluate: stores the user input essay and corresponding score We have two calls of function evaluate with the distinction of the routes with totally different route.

- 1) `@app.route('/evaluate', methods=["POST"])` This sends the essay to the server and receives the score after some calculation.
- 2) `@app.route('/contribute', methods=["POST"])` This is to store the essay and score pair provided by the user and appreciate the wholsomeness of it

async `main.evaluateFile(request)`

This function gives us a method to get the text out of file directly. Instead of writing text, it is better to upload text.

We see the function on a route `@app.route('/evaluateFile', methods=["POST"])`

async `main.firebase_login(request)`

This function deals with authentication in the code

Param request

Return type

Templating engine for Python3 rendering

```
>>> cool@cool.com and coolcool
```

async `main.firebase_register(request)`

This function registers us on the firebase platform.

It accepts user input from UI, ie. The email and password

The route for the same would be `@app.route('/registration', methods=["POST"])`

async `main.getEssay(request)`

This function `getEssay` is here to give us access to the cloud hosted url whenever. We can now check about donated essays till now. This controller operates on route `@app.route('/{prompt}')`

async `main.login(request)`

For the controller Login, we have two different routes

- 1) `@app.route('/')`, This is invoked when we have a redirect to and from the default start state or direct home href links
- 2) `@app.route('/register')` We get here when it's the login link via This function gives us the ability of logging in if our username and pass match

async `main.show_index(request)`

The function is a controller. It is invoked when we call `/auth`

3.2 Automated essay grader model

Here we have a very short description of the module prediction part of the parent project Automated essay grader model.

This is an efficient wrapper around a automated essay grading system

`prediction.getAvgFeatureVecs(essays, model, num_features)`

Main function to generate the word vectors for word2vec model.

Parameters

- **essays** – Input is essays
- **model** – This input specifies the model to be used to generate Feature vec
- **num_features** – This is a metric of number of columns in the matrix cols

Return type Returns the essay feature vector

`prediction.makeFeatureVec(words, model, num_features)`

This function generates feature vectors for each dimension ie. plane

Parameters

- **words** – We pass the words that we needed
- **model** – Model to decide on what basis police have been arrested

Return type `featureVec` is the result of the function

`prediction.predict(essay)`

This is the function which calculates the score given an essay

Parameters **essay** – This input is the essay that we got from the frontend.

Return type We are returning the `y_pred` values

`prediction.tokenizeEssay(essay)`

This function tokenizes the text provided as input in the form of essay

We remove `stopwords` from the essay

```
>>> This is america  
>>> america
```

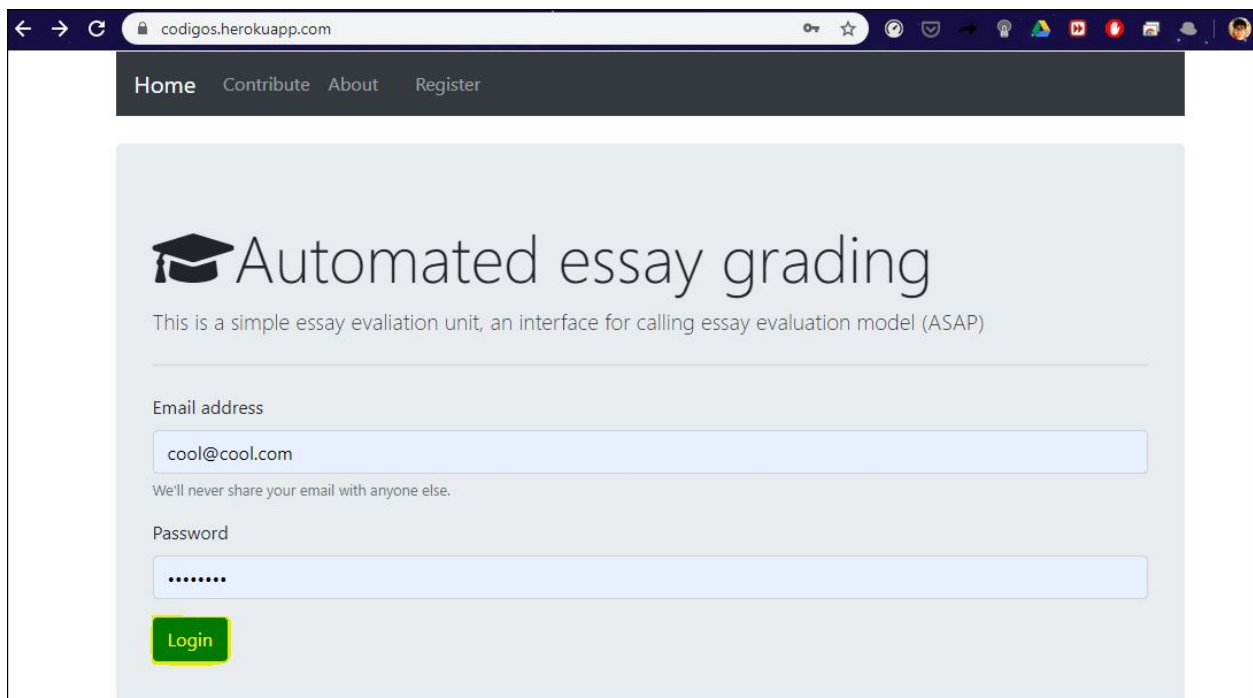
Parameters **essay** – Input in the form of text

Return type Tokenized words

3.3 Web views

This is the documentation for the web front end portal

Login.html

A screenshot of a web browser showing the login page for 'Automated essay grading'. The browser's address bar shows 'codigos.herokuapp.com'. The page has a dark navigation bar with links: Home, Contribute, About, and Register. The main content area has a light gray background. At the top, there is a graduation cap icon followed by the title 'Automated essay grading'. Below the title is a subtitle: 'This is a simple essay evaluation unit, an interface for calling essay evaluation model (ASAP)'. There is a horizontal line separator. Below this, there are two input fields. The first is labeled 'Email address' and contains the text 'cool@cool.com'. Below the email field is a small text: 'We'll never share your email with anyone else.' The second input field is labeled 'Password' and contains a series of dots. Below the password field is a green 'Login' button.

Register.html

Index.html

Score.html

Contribute.html

Thanks.html

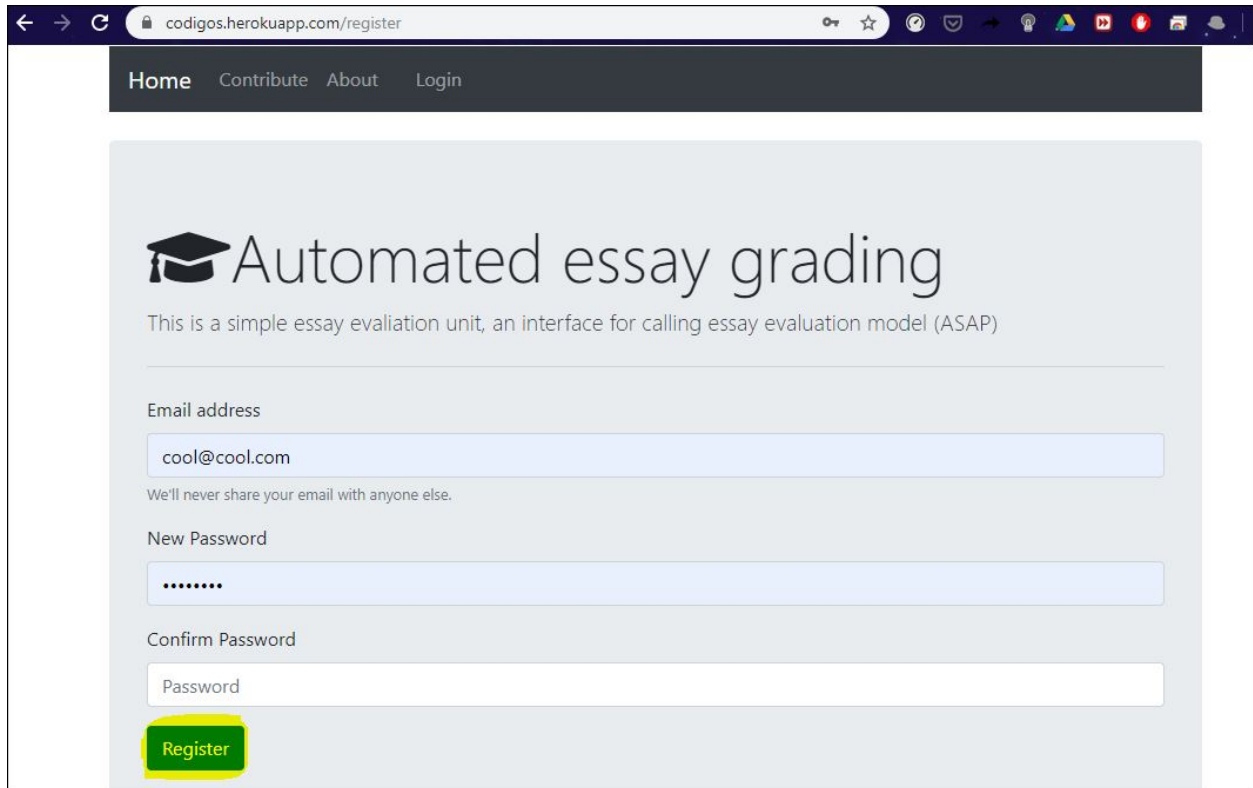
3.4 Android application

This is the documentation for android application

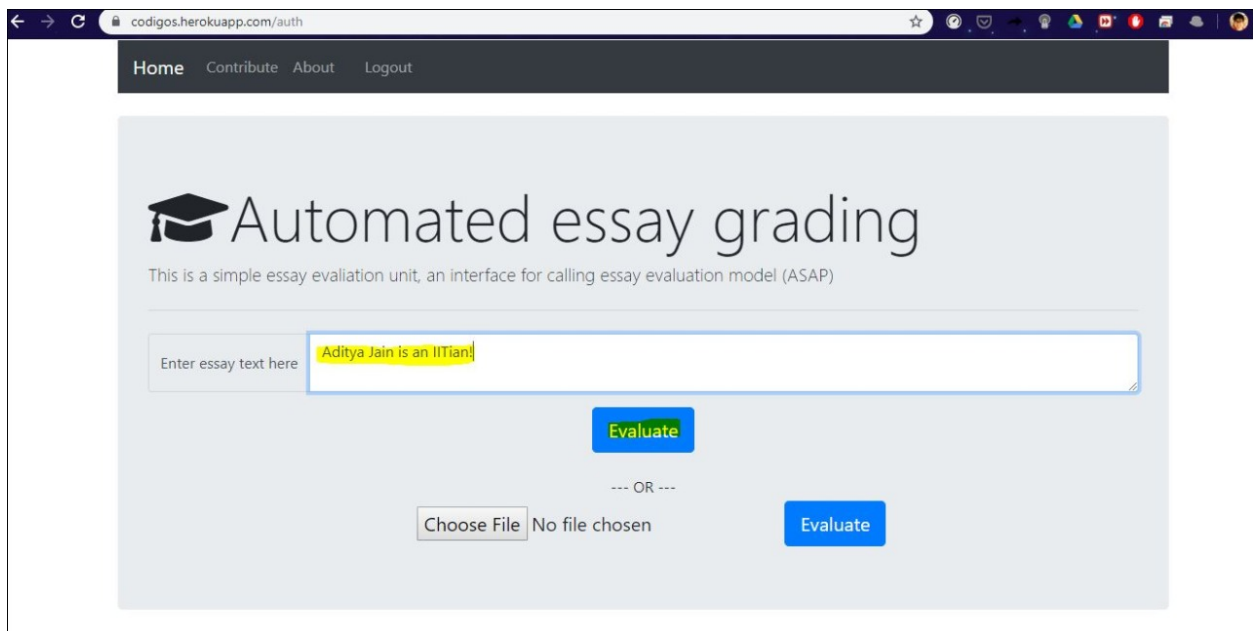
Login

Register

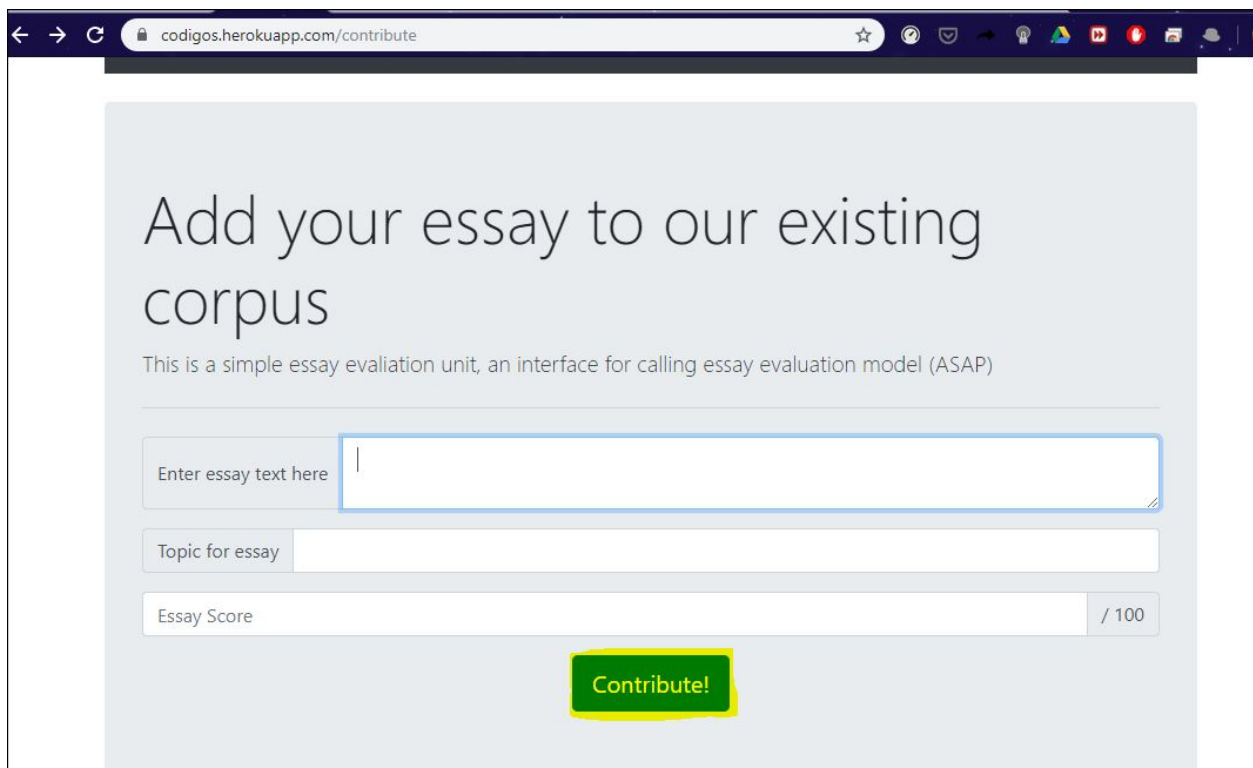
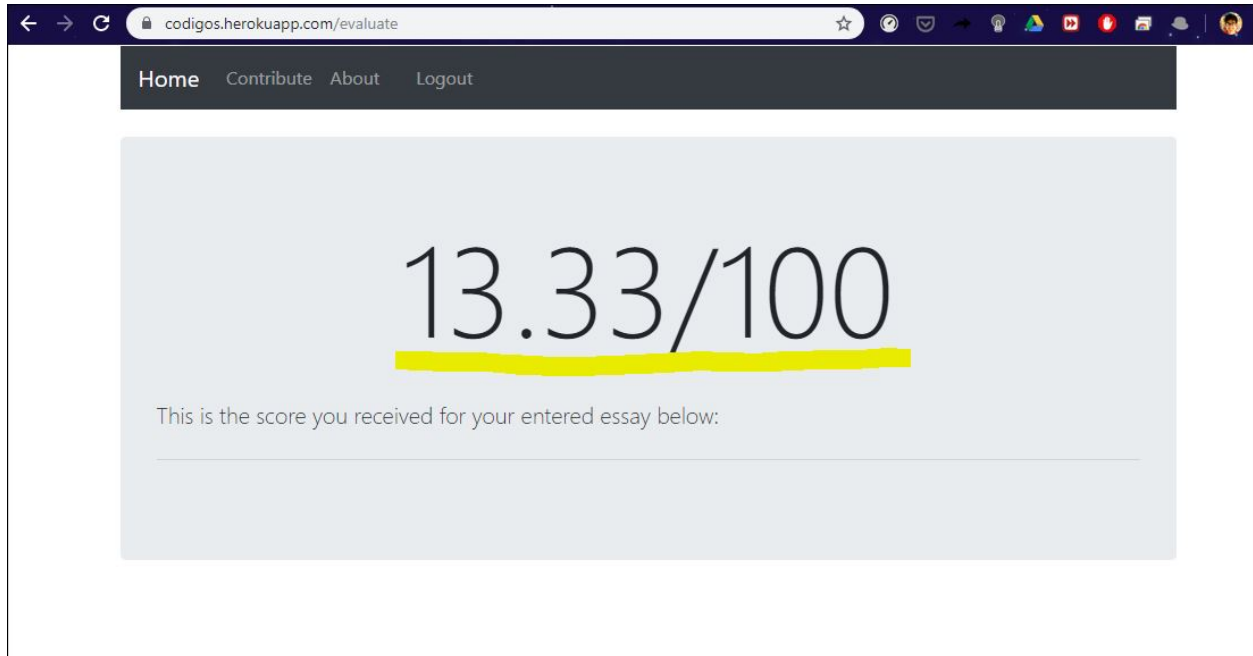
Evaluate

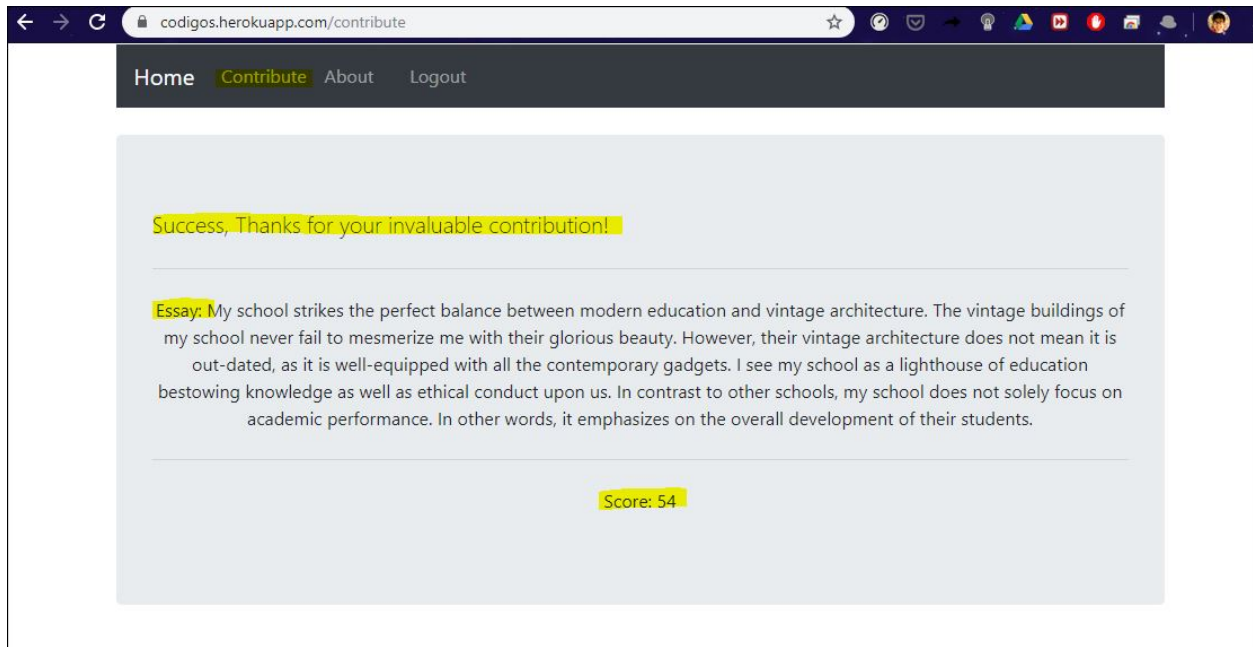


The screenshot shows the registration page of the Automated Essay Grader application. The browser address bar displays `codigos.herokuapp.com/register`. The navigation bar includes links for Home, Contribute, About, and Login. The main heading is "Automated essay grading" with a graduation cap icon. Below the heading is a subtext: "This is a simple essay evaluation unit, an interface for calling essay evaluation model (ASAP)". The registration form consists of three input fields: "Email address" (containing "cool@cool.com"), "New Password" (masked with dots), and "Confirm Password" (containing "Password"). A green "Register" button is located at the bottom of the form. A small text line below the email field states: "We'll never share your email with anyone else."



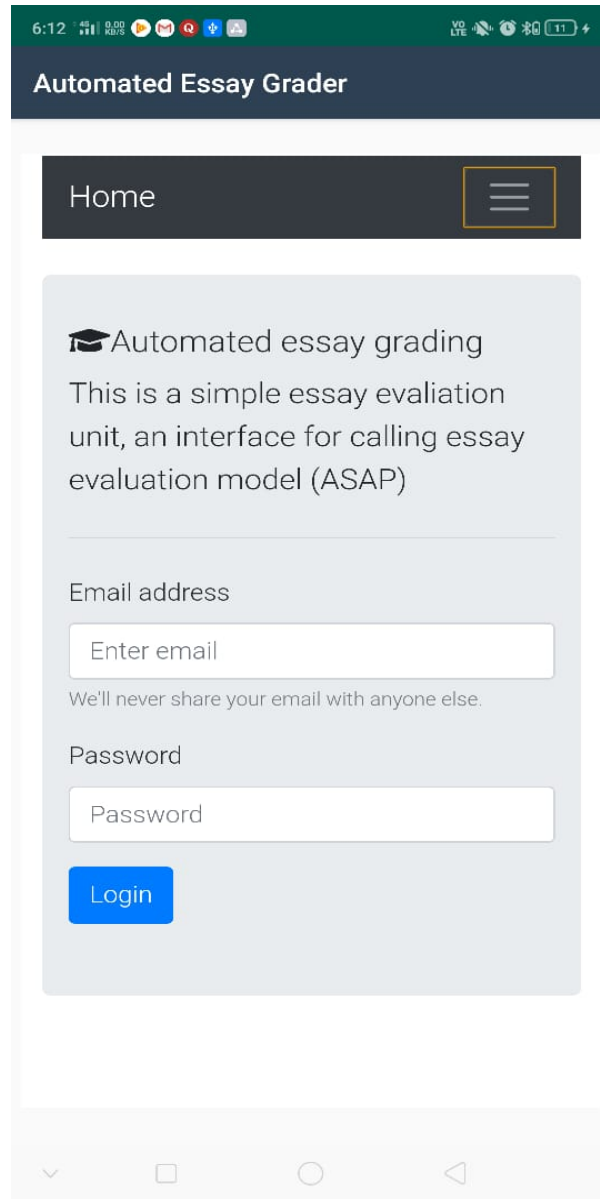
The screenshot shows the authentication page of the Automated Essay Grader application. The browser address bar displays `codigos.herokuapp.com/auth`. The navigation bar includes links for Home, Contribute, About, and Logout. The main heading is "Automated essay grading" with a graduation cap icon. Below the heading is a subtext: "This is a simple essay evaluation unit, an interface for calling essay evaluation model (ASAP)". The authentication form features a large text input field labeled "Enter essay text here" containing the text "Aditya Jain is an IITian!". Below this field is a blue "Evaluate" button. Further down, there is a section with a "Choose File" button, the text "No file chosen", and another blue "Evaluate" button, separated by a "--- OR ---" separator.

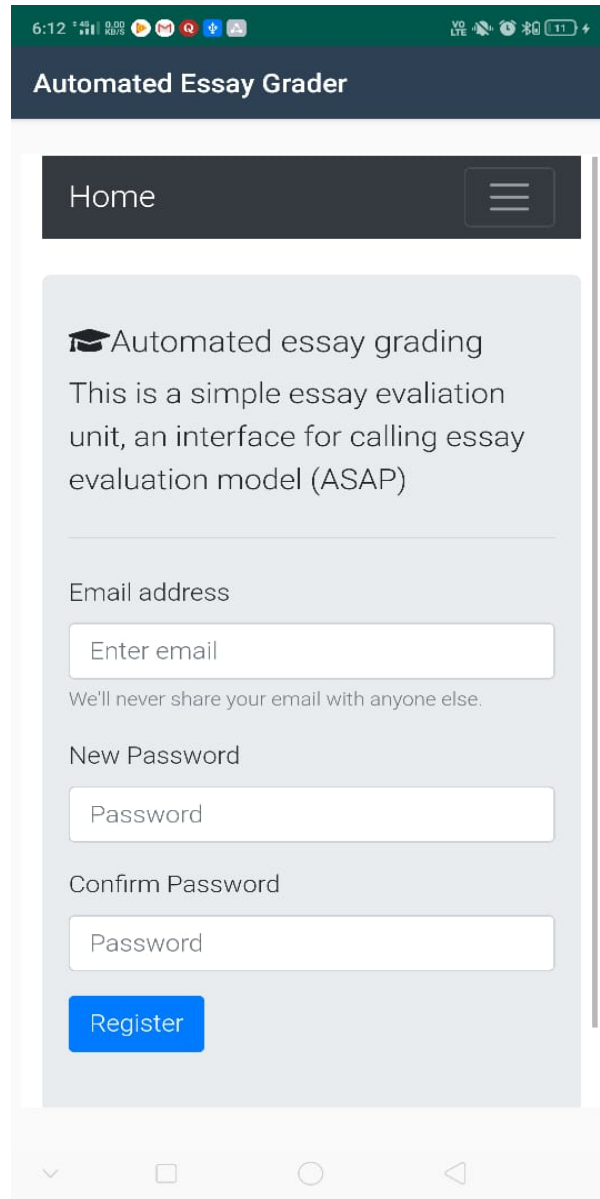




Score

Contribute





The screenshot shows a mobile application interface for 'Automated Essay Grader'. At the top, there is a status bar with the time 6:12, signal strength, 9.00 KB/s, and various icons. Below the status bar is a dark blue header with the text 'Automated Essay Grader'. The main content area has a dark blue bar with 'Home' and a hamburger menu icon. Below this, there is a light blue box containing a graduation cap icon, the text 'Automated essay grading', and a description: 'This is a simple essay evaluation unit, an interface for calling essay evaluation model (ASAP)'. Below the description are three input fields: 'Email address' with a placeholder 'Enter email', 'New Password' with a placeholder 'Password', and 'Confirm Password' with a placeholder 'Password'. A blue 'Register' button is at the bottom of the form. The bottom of the screen shows a white navigation bar with four icons: a downward arrow, a square, a circle, and a left-pointing triangle.

6:12 9.00 KB/s

Automated Essay Grader

Home

Automated essay grading

This is a simple essay evaluation unit, an interface for calling essay evaluation model (ASAP)

Email address

Enter email

We'll never share your email with anyone else.

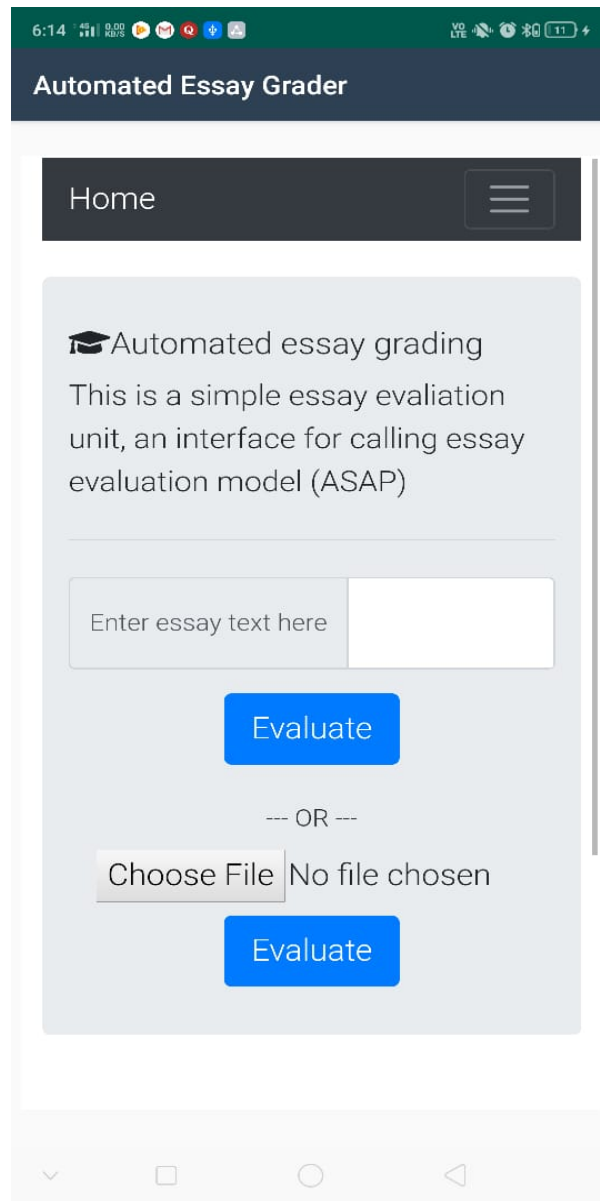
New Password

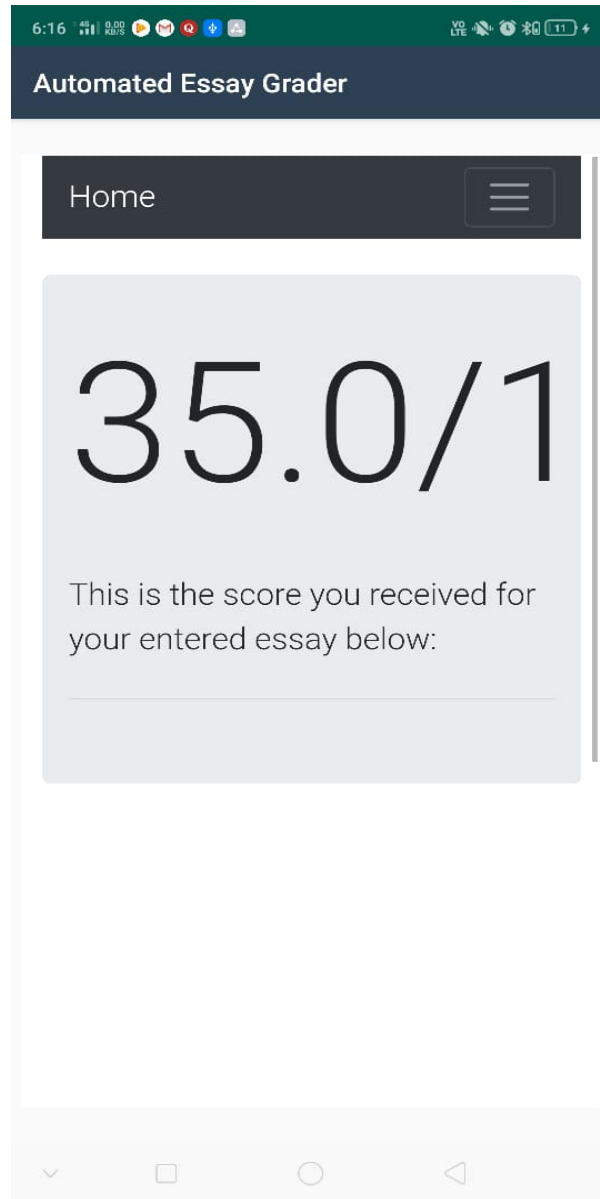
Password

Confirm Password

Password

Register





The screenshot shows the 'Automated Essay Grader' app on an Android phone. The status bar at the top displays the time 6:12, 4G LTE signal, 4.00 KB/s speed, and various notification icons. The app's title bar is dark blue with the text 'Automated Essay Grader'. Below this is a dark grey header with the word 'Home' and a hamburger menu icon. The main content area has a light grey background and contains the following text: 'Add your essay to our existing corpus' and 'This is a simple essay evaluation unit, an interface for calling essay evaluation model (ASAP)'. Below the text are three input fields: 'Enter essay text here' (a large text area), 'Topic for essay' (a single-line text field), and 'Essay Score' (a single-line text field) followed by '/ 100'. A blue 'Contribute!' button is positioned below the input fields. At the bottom of the screen is the standard Android navigation bar with back, home, and recent apps icons.

6:12 4G LTE 4.00 KB/s

Automated Essay Grader

Home

Add your essay to our existing corpus

This is a simple essay evaluation unit, an interface for calling essay evaluation model (ASAP)

Enter essay text here

Topic for essay

Essay Score / 100

Contribute!

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

m

main, [7](#)

p

prediction, [8](#)

INDEX

C

`contrbPage()` (*in module main*), 7

E

`evaluate()` (*in module main*), 7

`evaluateFile()` (*in module main*), 7

F

`firebase_login()` (*in module main*), 7

`firebase_register()` (*in module main*), 7

G

`getAvgFeatureVecs()` (*in module prediction*), 8

`getEssay()` (*in module main*), 8

L

`login()` (*in module main*), 8

M

`main (module)`, 7

`makeFeatureVec()` (*in module prediction*), 8

P

`predict()` (*in module prediction*), 8

`prediction (module)`, 8

S

`show_index()` (*in module main*), 8

T

`tokenizeEssay()` (*in module prediction*), 8