

Automated Transfer Credit Evaluator (ATCE) - Milestone 5 Evaluation

Tyler Dionne, Kendall Kelly

Task	Completion %	Tyler	Kendall	To Do
User Model Update	100%	0%	100%	N/A
Authentication System Setup	100%	50%	50%	N/A
Login Page Design & Development	100%	100%	0%	N/A
Registration Page Implementation	100%	0%	100%	N/A
Authentication CSS Styling	100%	50%	50%	N/A
Route Protection & Authorization	100%	100%	0%	N/A
Session Management	100%	0%	100%	N/A
User Interface Integration	100%	100%	0%	N/A
Testing	100%	50%	50%	N/A

Task 1: User Model Update

- Expanded on the already existing User db model to use the UserMixin argument which comes from the Flask-Login library and add necessary attributes and methods for the authentication system.
- This meant adjusting the User class to properly handle the login sessions, user ID retrieval and authentication state checking.
- This addition of UserMixin makes it so the model can handle is_authenticated, is_active, and is_anonymous properties required by Flask-Login.
- We will see these methods used later.

```
class User(db.Model, UserMixin):  
    id = db.Column(db.Integer, primary_key=True)  
    username = db.Column(db.String(20), unique=True, nullable=False)  
    email = db.Column(db.String(120), unique=True, nullable=False)  
    password = db.Column(db.String(60), nullable=False)  
  
    def __repr__(self):  
        return f"User('{self.username}', '{self.email}')
```

Task 2: Authentication System Setup

- Integrated the Flask extensions Flask-Login and Flask-Bcrypt into app.py to create a authentication system.
- Involved configuring a secure random secret key and setting up a login manager.

```
app.config['SECRET_KEY'] = os.urandom(24) # generate random secret
key for sessions
bcrypt = Bcrypt(app)
login_manager = LoginManager(app)
login_manager.login_view = 'login' # name of the route to redirect users to
login
login_manager.login_message_category = 'info'

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))
```

Task 3: Login Page Design & Development

- Designed a dark themed login page to match the aesthetic of our application and allow the user to fill out a form to login.
- The page has form validation, error messaging, and a remember me feature.

```
{% with messages = get_flashed_messages(with_categories=true) %}
{% if messages %}
    {% for category, message in messages %}
        <div class="alert alert-{{ category }}">{{ message }}</div>
    {% endfor %}
{% endif %}
{% endwith %}
```

...

```
<form method="POST" action="{{ url_for('login') }}">
    <div class="form-group">
        <label for="username">Username</label>
        <input type="text" id="username" name="username" required>
    </div>
    <div class="form-group">
        <label for="password">Password</label>
        <input type="password" id="password" name="password" required>
    </div>
    <div class="form-check">
        <input type="checkbox" id="remember" name="remember">
        <label for="remember">Remember Me</label>
    </div>
    <button type="submit" class="auth-button">Log In</button>
</form>
```

Task 3: Login Page Design & Development

- Needed proper login route handling with POST request processing.

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    if request.method == 'POST':
        username = request.form.get('username')
        password = request.form.get('password')
        remember = True if request.form.get('remember') else False

        user = User.query.filter_by(username=username).first()

        # check if user exists and password is correct
        if user and bcrypt.check_password_hash(user.password, password):
            login_user(user, remember=remember)
            next_page = request.args.get('next')
            return redirect(next_page) if next_page else
            redirect(url_for('index'))
        else:
            flash('Login unsuccessful. Please check username and password.',
                  'danger')
```

Task 4: Registration Page Implementation

- Made a complete user registration system that validates inputs and securely stores user data.
- The system has duplicate checking logic and password hashing for security:

```
hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')
```

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    if request.method == 'POST':
        username = request.form.get('username')
        email = request.form.get('email')
        password = request.form.get('password')
        confirm_password = request.form.get('confirm_password')

        # check if username or email already exist
        user_exists = User.query.filter_by(username=username).first()
        email_exists = User.query.filter_by(email=email).first()

        if user_exists:
            flash('Username already taken. Please choose a different one.', 'danger')
        elif email_exists:
            flash('Email already registered. Please use a different one.', 'danger')
        elif password != confirm_password:
            flash('Passwords do not match.', 'danger')
        else:
            # hash the password and create new user
            hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')
            new_user = User(username=username, email=email, password=hashed_password)
            db.session.add(new_user)
            db.session.commit()
            flash('Your account has been created! You can now log in.', 'success')
            return redirect(url_for('login'))
```

Task 6: Route Protection & Authorization

- Used the `@login_required` decorator to protect sensitive routes from unauthorized access. This way when you try to click on the ATCE tool without being logged in you cannot use it.
- This works with with the login manager to properly redirect unauthorized users.
- After the user logs in the system automatically redirects it back to the originally requested protected page using Flask's `next` parameter.

```
@app.route('/atce')  
@login_required  
def atce():  
    return render_template('atce.html')
```

...

```
login_manager = LoginManager(app)  
login_manager.login_view = 'login'  
login_manager.login_message_category = 'info'
```

...

```
next_page = request.args.get('next')  
return redirect(next_page) if next_page else redirect(url_for('index'))
```


Task 7: Session Management

- Made complete session handling with secure login, logout, and remember me functionality.
- For logout used proper session clearing.

```
login_user(user, remember=remember)
```

```
...
```

```
@app.route('/logout')
```

```
def logout():
```

```
    logout_user()
```

```
    return redirect(url_for('index'))
```

Task 8: User Interface Integration

- Updated main navigation and index page to dynamically display options based on authentication status.
- This way say a user successfully logs into the site they will no longer see the Login and Register button on the navigation bar they will see a Logout button and vice versa based on the value of `user.is_authenticated`.
- Changed the "Get Started" button button to also change based on user's authentication state.

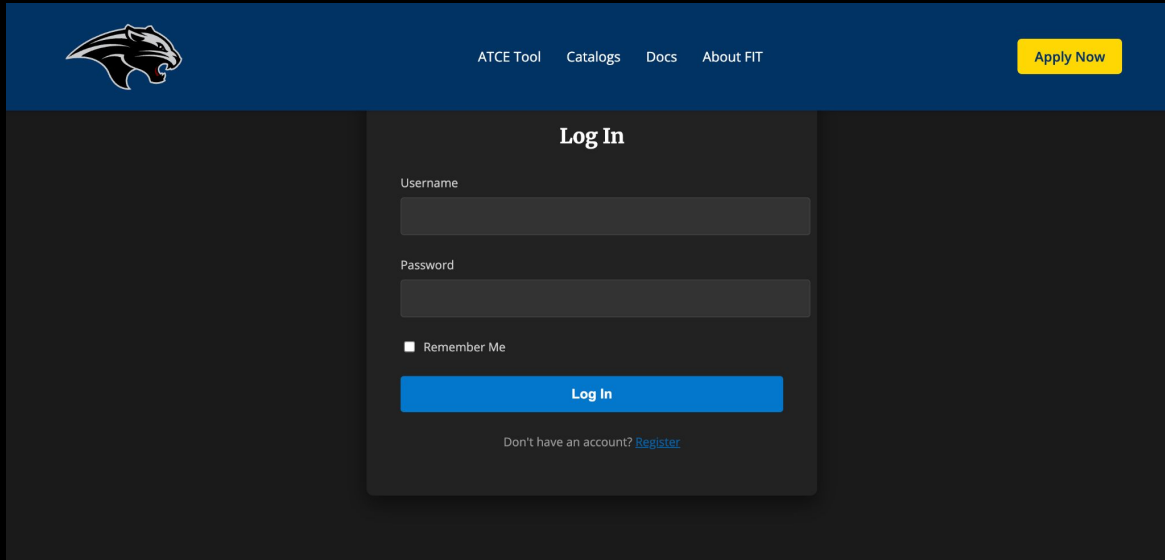
```
{% if current_user.is_authenticated %}
<li><a href="{{ url_for('logout') }}" class="nav-link">Logout</a></li>
{% else %}
<li><a href="{{ url_for('login') }}" class="nav-link">Login</a></li>
<li><a href="{{ url_for('register') }}"
class="nav-link">Register</a></li>
{% endif %}
```


...

```
<div class="button-group">
  {% if current_user.is_authenticated %}
    <a href="{{ url_for('atce') }}" class="cta-button">Use ATCE
Tool</a>
  {% else %}
    <a href="{{ url_for('login') }}" class="cta-button">Get Started</a>
  {% endif %}
  <a href="#" class="secondary-button">Learn More</a>
</div>
```

Demo

- From the homepage go to the Login button on the menu at the top.





ATCE Tool Catalogs Docs About FIT

Apply Now

Log In

Username

Password


☐ Remember Me

Log In

Don't have an account? [Register](#)

Demo

- Do not have account so go to register and enter info.



ATCE Tool Catalogs Docs About FIT

[Apply Now](#)

Username

TYLER

Email

tdionne2021@my.fit.edu

Password

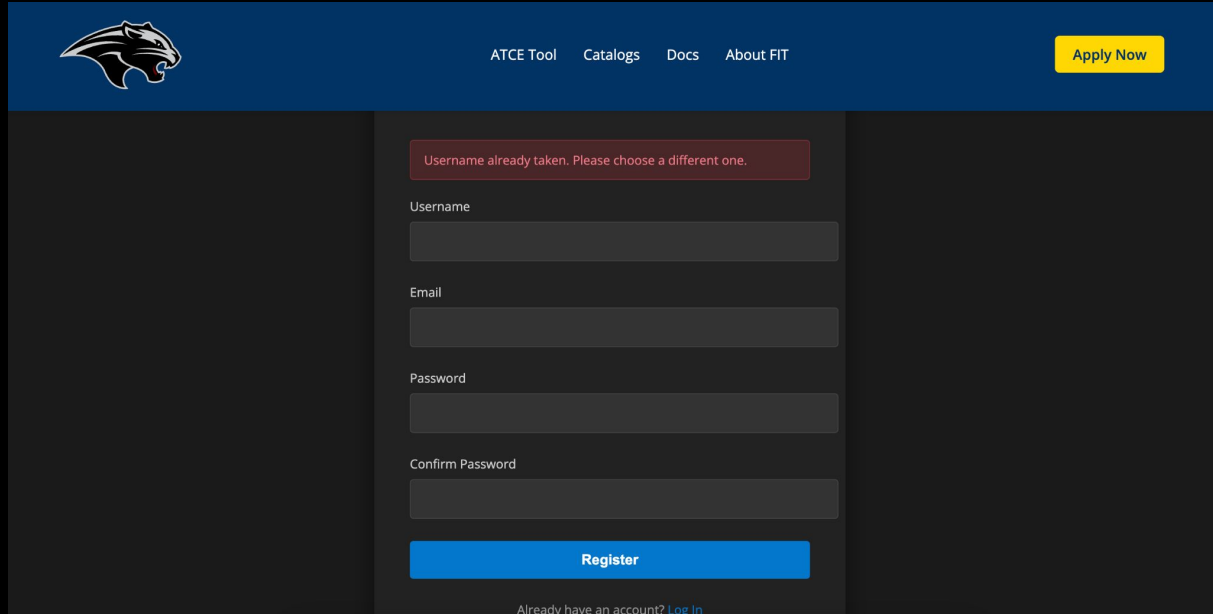
Confirm Password

[Register](#)

Already have an account? [Log In](#)

Demo

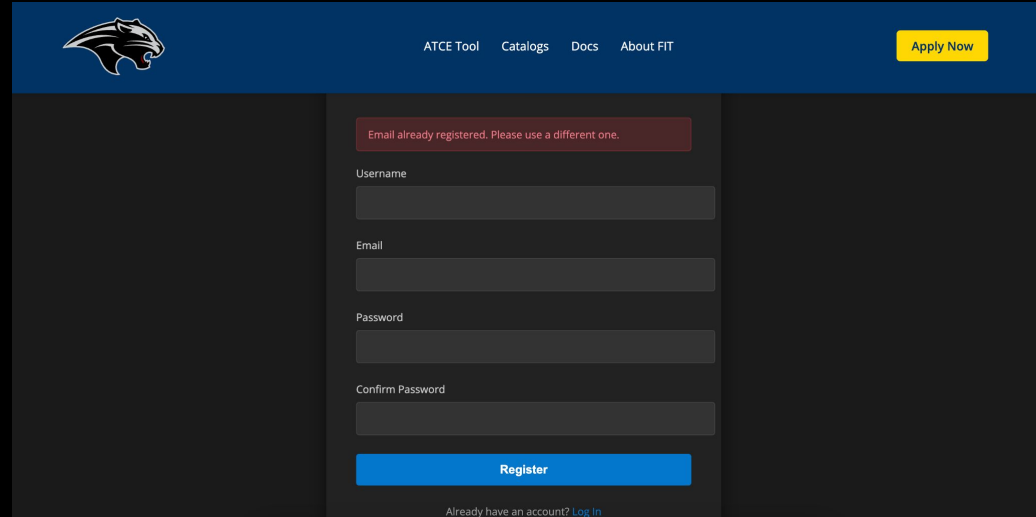
- Test warning messages by entering the same username as an already registered user, the same email as an already registered user, and two different passwords.



The screenshot shows a web application interface for user registration. At the top, there is a dark blue header bar. On the left side of the header is a logo of a stylized animal head (possibly a panther or cougar). To the right of the logo are four links: "ATCE Tool", "Catalogs", "Docs", and "About FIT". Further to the right is a yellow button labeled "Apply Now". Below the header, the main content area has a dark gray background. In the center, there is a registration form. At the top of the form is a red warning message box that says "Username already taken. Please choose a different one." Below this message are four input fields: "Username", "Email", "Password", and "Confirm Password". Each field is a light gray rectangle. At the bottom of the form is a blue button labeled "Register". Below the "Register" button, there is a link that says "Already have an account? [Log In](#)".

Demo

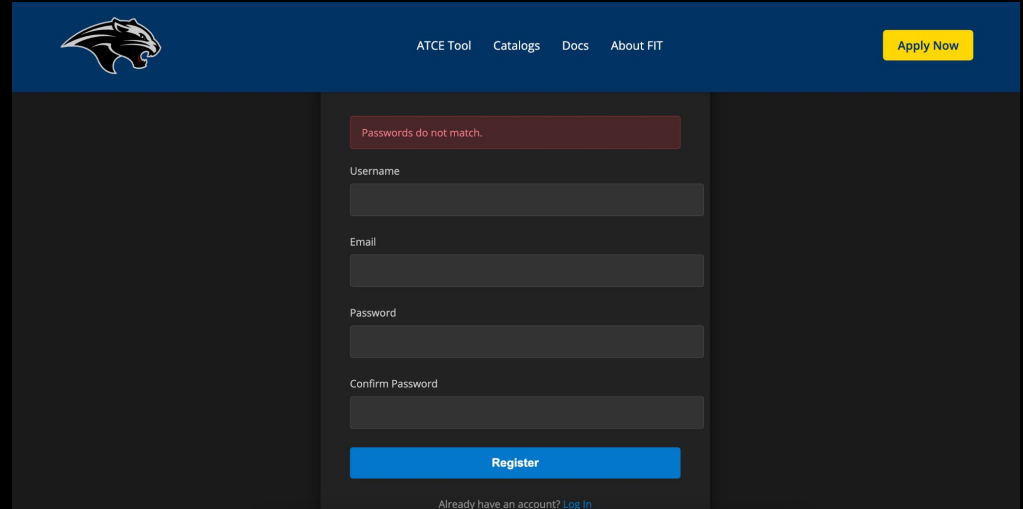
- Test warning messages by entering the same username as an already registered user, the same email as an already registered user, and two different passwords.



The screenshot shows a web registration form. At the top, there is a blue header bar containing a logo on the left, navigation links 'ATCE Tool', 'Catalogs', 'Docs', and 'About FIT' in the center, and a yellow 'Apply Now' button on the right. The main content area has a dark gray background. In the center, there is a white registration form. At the top of this form is a red warning box with the text 'Email already registered. Please use a different one.' Below this are four input fields labeled 'Username', 'Email', 'Password', and 'Confirm Password'. Each field has a light gray border and a small eye icon on the right. At the bottom of the form is a blue 'Register' button. Below the button, there is a link that says 'Already have an account? [Log In](#)'.

Demo

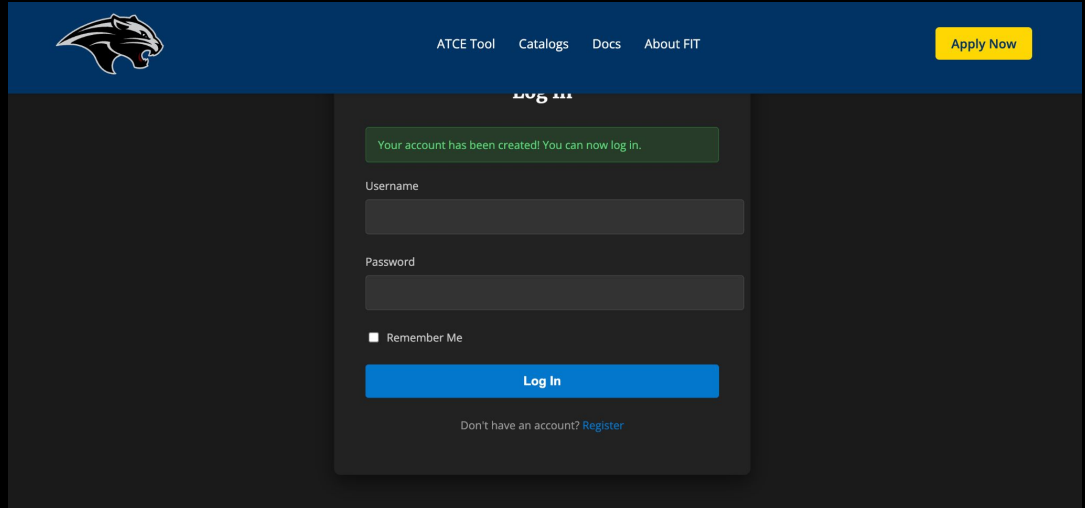
- Test warning messages by entering the same username as an already registered user, the same email as an already registered user, and two different passwords.



The screenshot shows a web registration form. At the top, there is a blue header bar containing a logo on the left, navigation links "ATCE Tool", "Catalogs", "Docs", and "About FIT" in the center, and a yellow "Apply Now" button on the right. The main content area has a dark gray background. In the center, there is a white registration form. At the top of this form is a red error message box that says "Passwords do not match.". Below this are four input fields labeled "Username", "Email", "Password", and "Confirm Password". At the bottom of the form is a blue "Register" button. Below the button, there is a link that says "Already have an account? Log In".

Demo

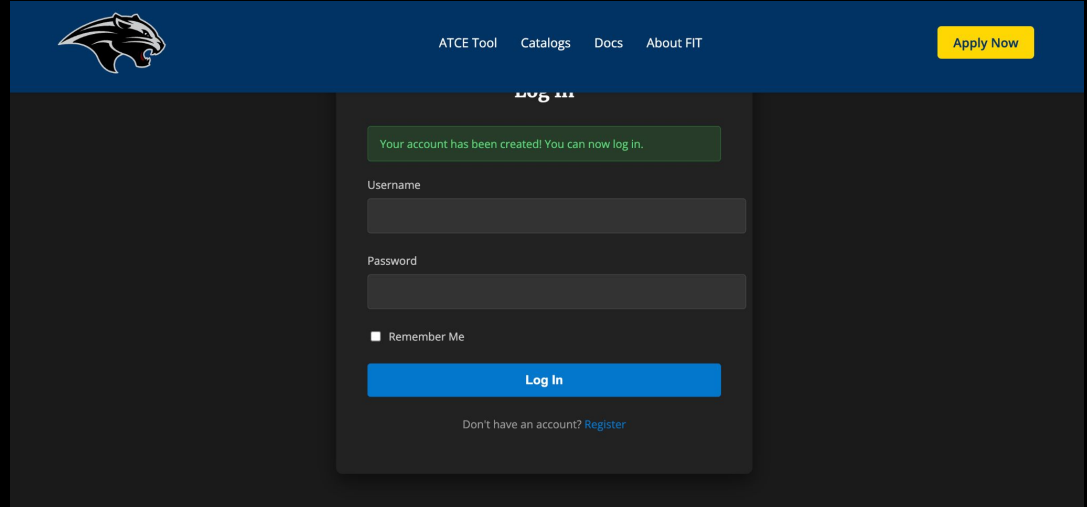
- Test success message by successfully registering



The screenshot displays a web application interface. At the top, a blue header bar contains a logo on the left, navigation links (ATCE Tool, Catalogs, Docs, About FIT) in the center, and a yellow 'Apply Now' button on the right. Below the header, a dark gray background features a central white box. At the top of this box is a green message: 'Your account has been created! You can now log in.' Below the message are two input fields labeled 'Username' and 'Password'. Under the 'Password' field is a checkbox labeled 'Remember Me'. A blue 'Log In' button is positioned below the checkbox. At the bottom of the white box, a link reads 'Don't have an account? Register'.

Demo

- Test login + session management functionality.
- (Note: When logged in won't see login or register buttons on the menu or the Get Started button)



The screenshot shows a web application interface. At the top, there is a dark blue header bar. On the left side of the header is a logo of a stylized animal head. On the right side of the header are navigation links: "ATCE Tool", "Catalogs", "Docs", and "About FIT". Further to the right is a yellow button labeled "Apply Now". Below the header, the main content area has a dark gray background. In the center, there is a white login form. At the top of the form is a green message box that says "Your account has been created! You can now log in." Below this are input fields for "Username" and "Password". There is a checkbox labeled "Remember Me". At the bottom of the form is a blue button labeled "Log In". Below the button is a link that says "Don't have an account? Register".

00

End