VETS 2 TECH

AUTHORS: BETH GALLATIN, CONNIE RODRIGUEZ, JOHN MCDURMON

ORGANIZATION: INGE MARCUS SCHOOL OF ENGINEERING | SAINT MARTIN'S UNIVERSITY

STAKEHOLDERS

Saint Martin's UNIVERSITY

WASHINGTON VETS2TECH

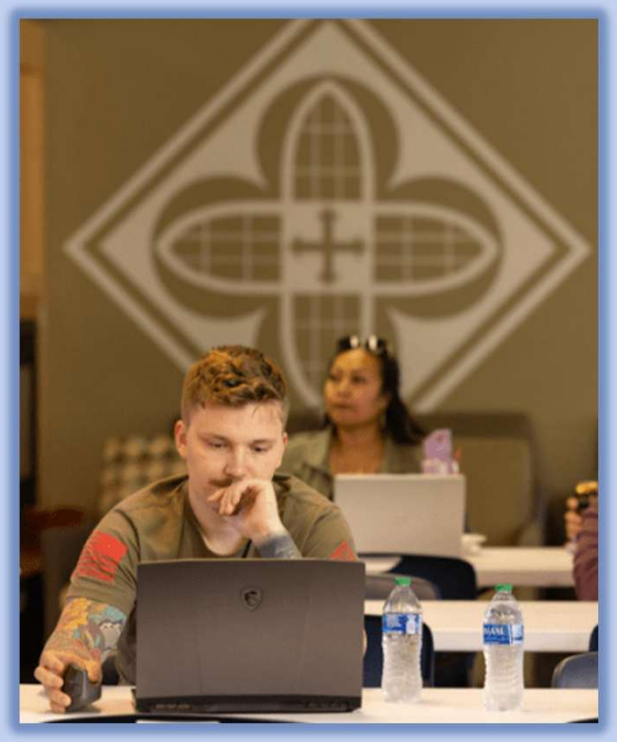
CLIENT DETAILS:

- Accelerated 16-week program
- Professional development education and training in Computer Science/IT
- Provides three learning pathways:
  - Server Cloud Application
  - Cloud Application Development
  - Cybersecurity Administration

OBJECTIVE:

- Create a web platform that guides students to the best-fit program
- Provide real-time AI-powered support using a 'study buddy'
- Leverage Data Analytics for actionable insights to stakeholders
- Ensure Scalability, Usability, and Security

PURPOSE



PROBLEM:

- Veteran's face challenges in transitioning to tech careers
- Accelerated programs require tailored support and resources.

GOALS:

- Provide guided learning paths and interactive study tools
- Enhance engagement through machine learning and AI-powered support
- Deliver predictive insights to improve outcomes

MEET THE TEAM

BETH GALLATIN

Project Manager

Stakeholder
Communication, Backend

CONNIE RODRIGUEZ

Development

Backend, Frontend

JOHN MCDURMON

Design

UI/UX design

PROJECT CONSTRAINTS

TECHNICAL CONSTRAINTS

- AI Integration Complexity
- Scalability vs. Budget Constraints
- Data Privacy and Security Compliance
- Adaptive Testing Model Development

QUALITATIVE CONSTRAINTS

- User-centric Design
- Cross-platform Compatibility
- Stakeholder Feedback Integration
- Ethical Considerations

ENGINEERING AND COMPUTING PRINCIPLES

- **Modular and Scalable Architecture:**
  - Problem Solved: AI Integration Complexity and Scalability vs. Budget Constraints
- **Interface Segregation Principle:**
  - Problem Solved: Overloaded Interfaces and Complexity in Code Maintenance
- **Security by Design:**
  - Problem Solved: Data Privacy, Unauthorized Access, and Compliance
- **Agile Development and Iterative Feedback Integration:**
  - Problem Solved: Stakeholder Feedback Integration and User-Centric Design

DESIGN EXPLORATION AND ALTERNATIVES

FRAMEWORK

- **Django**
- ASP.NET
- React

LANGUAGE

- **Python**
- C#
- JavaScript

QUIZ SYSTEM DESIGN

- **Custom Development**
- Learning Management System
- Open-source Quiz Libraries

DATA ANALYTICS

- **R**
- Python
- Tableau

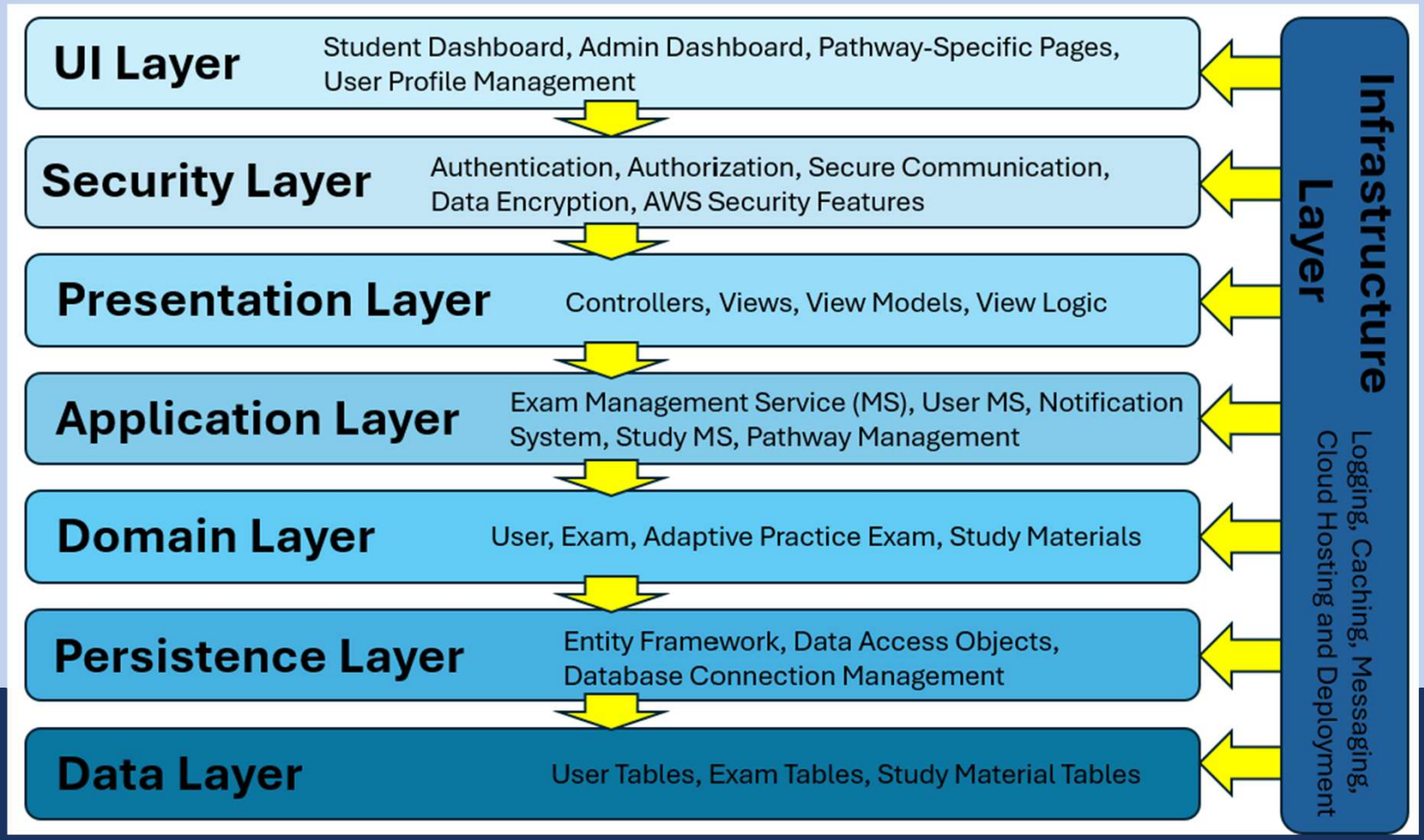
AI POWERED

- **ML-powered AI**
- Basic Rule-based AI

ADAPTIVE LEARNING

- **AI Driven Adaptive Learning**
- Static Quiz Questions

System Architecture



CODES, GUIDELINES, AND REGULATIONS

ASSOCIATION FOR COMPUTING MACHINERY

Fairness, Transparency, Privacy, Integrity

FAMILY EDUCATION RIGHTS & PRIVACY ACT

Data Protection, Consent, Student Rights

INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS

Reliability, Safety, Professionalism, Quality

AMERICAN'S WITH DISABILITIES ACT

Accessiblity, Inclusivity, Usability, Compliance

RISKS

RISKS FOUND	MITIGATION
• User Adoption and Engagement	• Conducted user engagement campaign
• Cross-Platform Compability	• Applying responsive design principles
• Unforseen Budget Overruns	• Implementing real-time budget tracking
• Proprietary Content	• Subject matter expert for intellectual property
• Long-Term Maintenance Challenges	• Thoroughly documenting of all coding processes
• Operator Error in System Managment	• Recursive comprehensive training, procedures, and role based access controls.

RISKS: CHATBOT

RISKS FOUND	MITIGATION
• Content Overlap or Redundancy	• Dynamic Content Creation
• Training Data Quality	• Regularly Updating Training Data
• Integration Complexity	• Using Modular Design for Isolation
• Real-Time Response Challenge	• Implementing Load Balancing
• Security	• Access Control, Tracking and Monitoring

RISKS: MACHINE LEARNING

RISKS FOUND

- Algorithm Bias
- Model Accuracy
- Overfitting
- Security of Machine Learning

MITIGATION

- Using diverse training datasets and regular bias audits
- Applied supervised learning algorithms with extensive validation
- Using cross-validation and regularization
- Employ adversarial training to make model resilient against malicious input

RISKS: DATA ANALYSIS

RISKS FOUND	MITIGATION
• Data Breach/Data Loss • System Security Vulnerablities • Misinterpretation of Data • Algorithmic Errors	• Implemented strong encryption protocols, secure data access control, regular security audits, and penetration testing • Enforcing secure user authentication and maintain transparency and data usage • Employing data validation checks • Executing testing and validation of all data

RISKS: SECURITY

RISKS FOUND	MITIGATION
• Unauthorized Access to Admin Functions • Data Exposure in Transit • Incomplete Session Handling • Injection and XSS Vulnerability • Delayed Incident Response	• Enforce role-based access control • Require HTTPS/TLS 1.3 for all communication • Use session expiration, secure cookies, and automatic logout for inactivity • Using Django's built-in sanitization, validate all inputs • Maintain formal Incident Response Plan (IRP)

ACTION PLAN

2025

Jan

- Transitioned from ASP.NET with C# to Django with Python
- Rebuilt the platform from the ground up to support AI integration and adaptive learning
- Established new database structure and authentication system
- Set up the development environment and reconfigured version control

Feb

- Implemented core platform features, including quizzes, user authentication and study materials
- Began testing and refining the adaptive quiz system
- Addressed cross-platform compatibility with design updates
- Overcame propriety content challenges
- Presented project poster at the Engineering Banquet

ACTION PLAN

2025

Mar

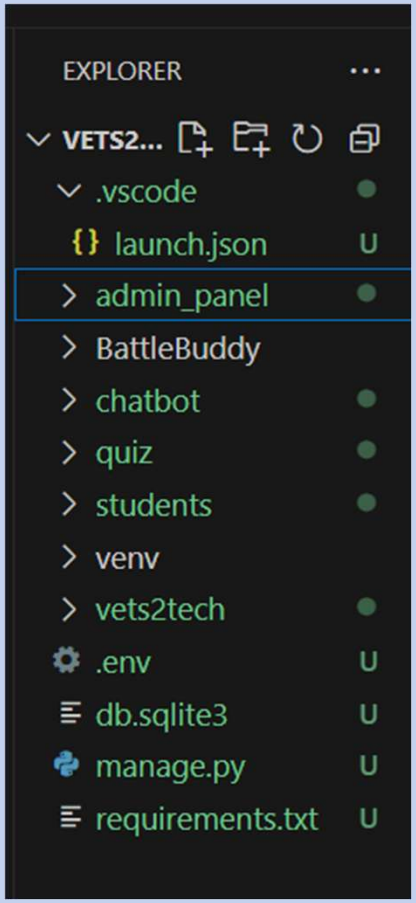
- Conducted full system testing, including functionality, performance, and security
- Fixed bugs and optimization issues to improve user experience
- Finalized role-based access controls and admin functionalities
- Created comprehensive documentation for long-term maintenance

Apr

- Presentation to the EAB board
- Ensured that the system components completed were working properly for stakeholder review
- Conducted last-minute testing and refining at 85% completion
- Preparing for presentation at scholars day
- Preparing for future development

BACKEND

BACKEND




```python
from django.shortcuts import render
from .models import Question, UserQuizProgress
from django.contrib.auth.decorators import login_required

@login_required
def generate_quiz(request, topic=None, question_count=15):
    user = request.user

    # If the user chose the "all" option, pull questions from all topics
    if topic == "all":
        questions = []
        for t, _ in Question.TOPICS:
            progress, _ = UserQuizProgress.objects.get_or_create(user=user, topic=t)
            topic_questions = Question.objects.filter(
                topic=t,
                difficulty=progress.difficulty
            ).order_by('?')[:10]  # 10 per topic
            questions.extend(topic_questions)
    else:
        # Single-topic quiz
        progress, _ = UserQuizProgress.objects.get_or_create(user=user, topic=topic)
        questions = Question.objects.filter(
            topic=topic,
            difficulty=progress.difficulty
        ).order_by('?')[:question_count]

    return render(request, "quiz.html", {
        "questions": questions,
        "topic": topic
    })
```

BACKEND

```python
from django.db import models
from django.contrib.auth.models import User  # Links quiz progress to users

class Question(models.Model):
    """
    Stores a single quiz question with multiple-choice answers.
    Each question is tagged with a topic and difficulty level to support adaptive learning.
    """
    # Difficulty levels used to dynamically adjust question selection based on user performance
    DIFFICULTY_LEVELS = [
        ('easy', 'Easy'),
        ('medium', 'Medium'),
        ('hard', 'Hard'),
    ]

    # Predefined topics to categorize questions and track user strengths/weaknesses
    TOPICS = [
        ('cybersecurity', 'Cybersecurity'),
        ('networking', 'Networking'),
        ('cloud', 'Cloud'),
        ('computer_basics', 'Computer Basics'),
    ]
```

```python
# Core question and answer fields
question_text = models.TextField()  # Stores the question itself
option_1 = models.CharField(max_length=255)
option_2 = models.CharField(max_length=255)
option_3 = models.CharField(max_length=255)
option_4 = models.CharField(max_length=255)
correct_answer = models.CharField(max_length=255)  # Correct answer for validation
difficulty = models.CharField(max_length=10, choices=DIFFICULTY_LEVELS)
topic = models.CharField(max_length=20, choices=TOPICS)

#user will see topic and difficulty level when they are taking the quiz
def __str__(self):
    return f"{self.topic} - {self.difficulty}: {self.question_text}"
```

BACKEND

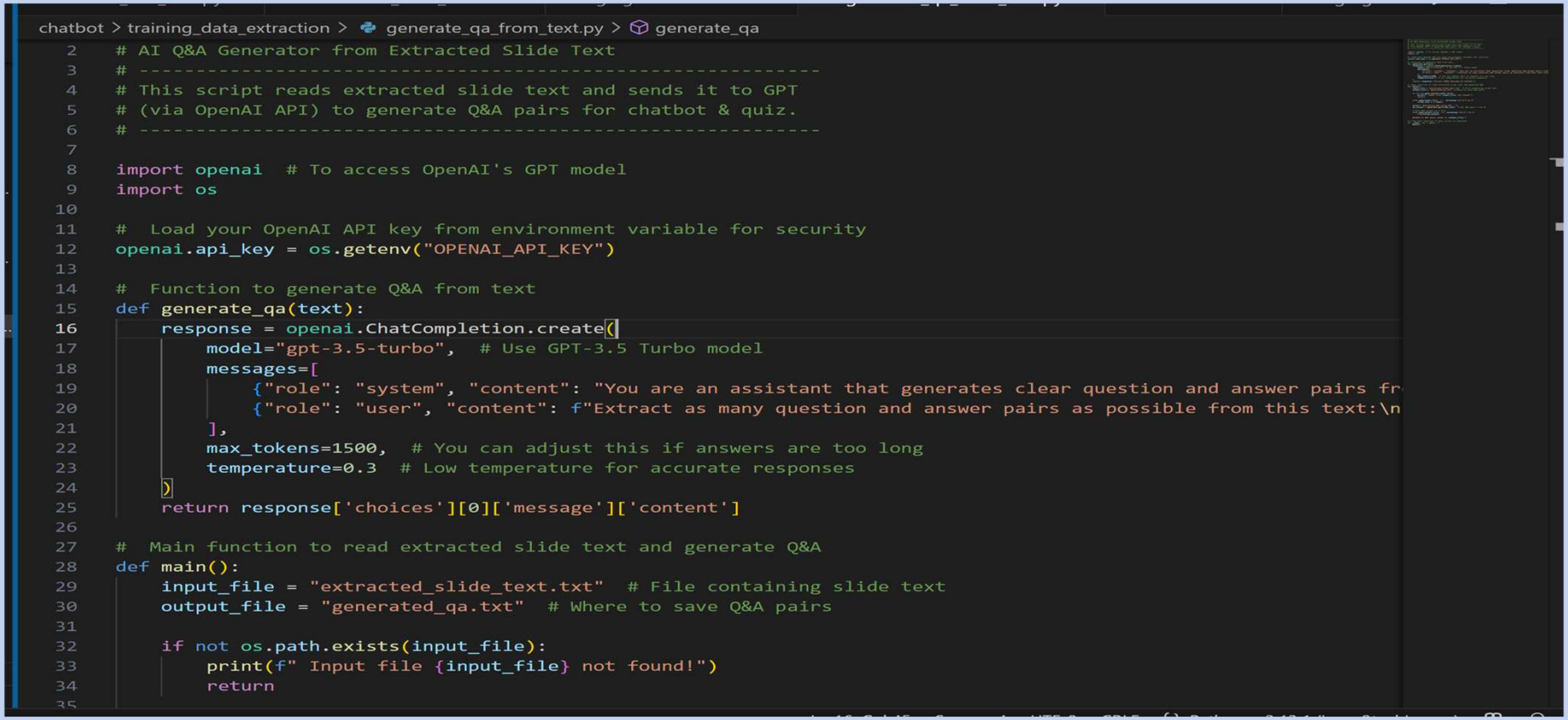
```python
class UserQuizProgress(models.Model):
    """
    Tracks an individual user's performance in each topic and difficulty level.
    Enables adaptive question delivery and personalized feedback by storing ongoing results.
    """
    # User the progress data belongs to
    user = models.ForeignKey(User, on_delete=models.CASCADE)  # Links progress to a user

    # Topic and difficulty being tracked for targeted improvement
    topic = models.CharField(max_length=20, choices=Question.TOPICS)  # Topic being tracked
    difficulty = models.CharField(max_length=10, choices=Question.DIFFICULTY_LEVELS, default="easy")

    # Performance tracking fields
    correct_count = models.IntegerField(default=0)  # Number of correct answers
    incorrect_count = models.IntegerField(default=0)  # Number of incorrect answers

    #shows user the subject and their current level
    def __str__(self):
        return f"{self.user.username} - {self.topic} ({self.difficulty})"
```

BACKEND

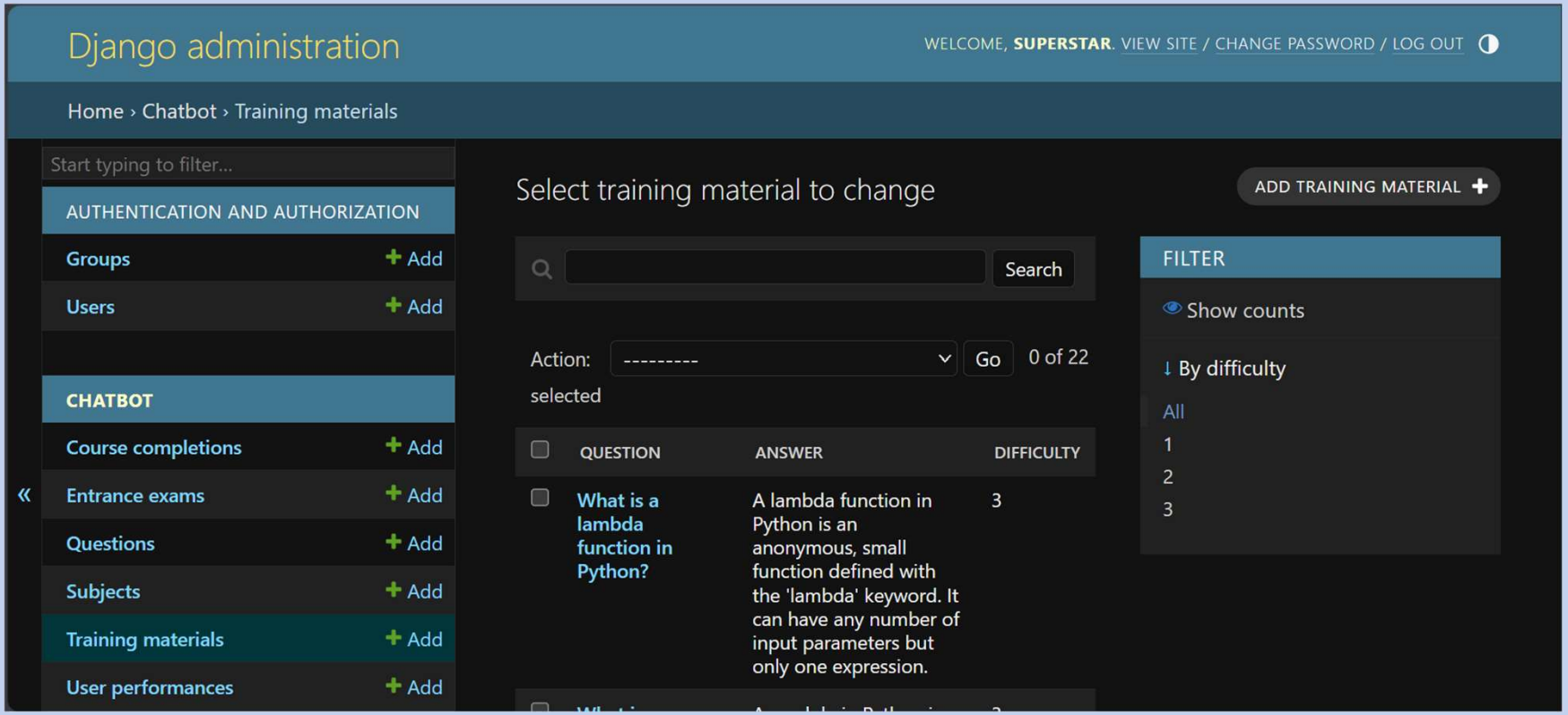
chatbot > training_data_extraction > extract_raw_text.py > ...

```python
# ------------------------------------------------------------
#  Extract Text from PowerPoint Slides
# ------------------------------------------------------------
# This script will search for all .pptx (PowerPoint) files
# inside the 'media/slides' folder and any subfolders (like 'Presentations').
# It will extract all the text from the slides and save it
# into a single text file called 'extracted_slide_text.txt'.
#
# This file will help us review all slide content and
# prepare it for chatbot or quiz training.
# ------------------------------------------------------------

from pptx import Presentation  # Import the library to read PowerPoint files
import os  # Import the library to work with folders and files

# ------------------------------------------------------------
# Function: extract_text_from_all_pptx
# Purpose: Search all folders, extract text from slides, save it to a text file
# ------------------------------------------------------------
def extract_text_from_all_pptx(input_folder, output_file):
    all_text = []  # This will store all text from all slides

    # Walk through all subfolders and files inside input_folder
    for root, dirs, files in os.walk(input_folder):  # os.walk lets us search inside subfolders too
        for filename in files:
            if filename.endswith('.pptx'):  # Only process PowerPoint files
                full_path = os.path.join(root, filename)  # Full path to the .pptx file
                print(f"Processing file: {full_path}")  # Show which file we are working on (DEBUG info)

                # Open the PowerPoint file
                prs = Presentation(full_path)
                slide_texts = []  # Store text from each slide in this list

                # Go through each slide
```

BACKEND


chatbot > training_data_extraction > generate_qa_from_text.py > generate_qa
# AI Q&A Generator from Extracted Slide Text
# --
# This script reads extracted slide text and sends it to GPT
# (via OpenAI API) to generate Q&A pairs for chatbot & quiz.
# --
import openai # To access OpenAI's GPT model
import os
# Load your OpenAI API key from environment variable for security
openai.api_key = os.getenv("OPENAI_API_KEY")
# Function to generate Q&A from text
def generate_qa(text):
response = openai.ChatCompletion.create(
model="gpt-3.5-turbo", # Use GPT-3.5 Turbo model
messages=[
{"role": "system", "content": "You are an assistant that generates clear question and answer pairs fr
{"role": "user", "content": f"Extract as many question and answer pairs as possible from this text:\n
],
max_tokens=1500, # You can adjust this if answers are too long
temperature=0.3 # Low temperature for accurate responses
)
return response['choices'][0]['message']['content']
# Main function to read extracted slide text and generate Q&A
def main():
input_file = "extracted_slide_text.txt" # File containing slide text
output_file = "generated_qa.txt" # Where to save Q&A pairs
if not os.path.exists(input_file):
print(f" Input file {input_file} not found!")
return

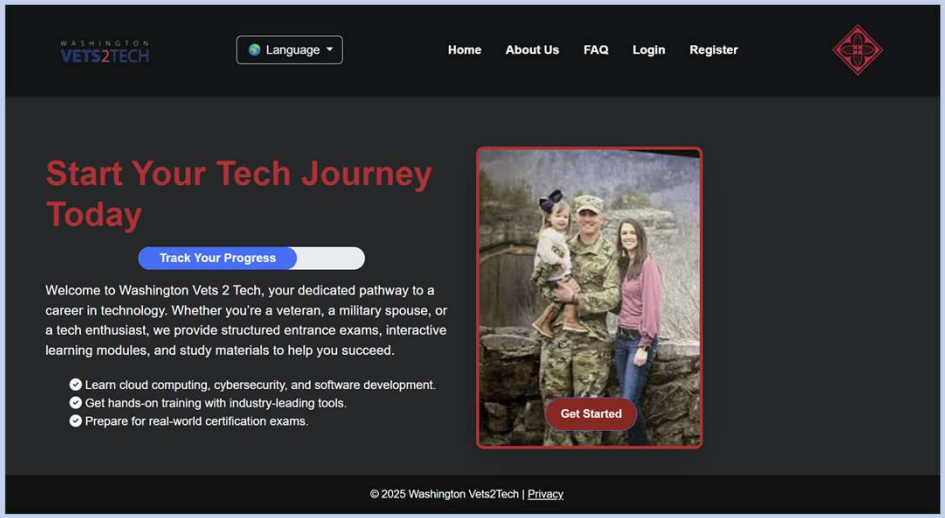
BACKEND


Django administration
WELCOME, SUPERSTAR. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Chatbot › Training materials
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
CHATBOT
Course completions + Add
Entrance exams + Add
Questions + Add
Subjects + Add
Training materials + Add
User performances + Add
Select training material to change
ADD TRAINING MATERIAL +
Search
Action: --------- Go 0 of 22 selected
QUESTION
ANSWER
DIFFICULTY
What is a lambda function in Python?
A lambda function in Python is an anonymous, small function defined with the 'lambda' keyword. It can have any number of input parameters but only one expression.
3
FILTER
Show counts
By difficulty
All
1
2
3

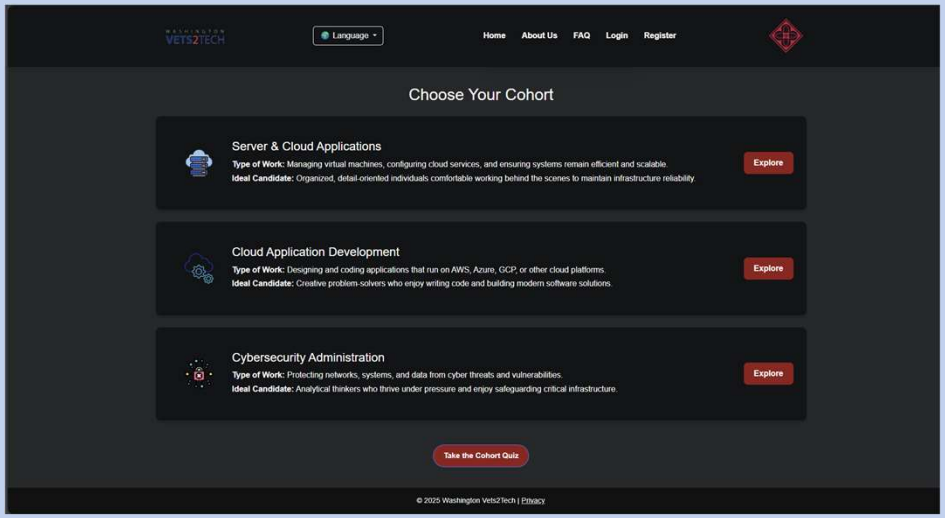
FRONTEND

HOME PAGE

Home Page



Cohort Selection



TRANSLATION BUTTON

Internalization Framework

```
from django.conf.urls.i18n import i18n_patterns
from django.views.i18n import set_language
```

- Internationalization i18 framework is imported
- The languages are imported

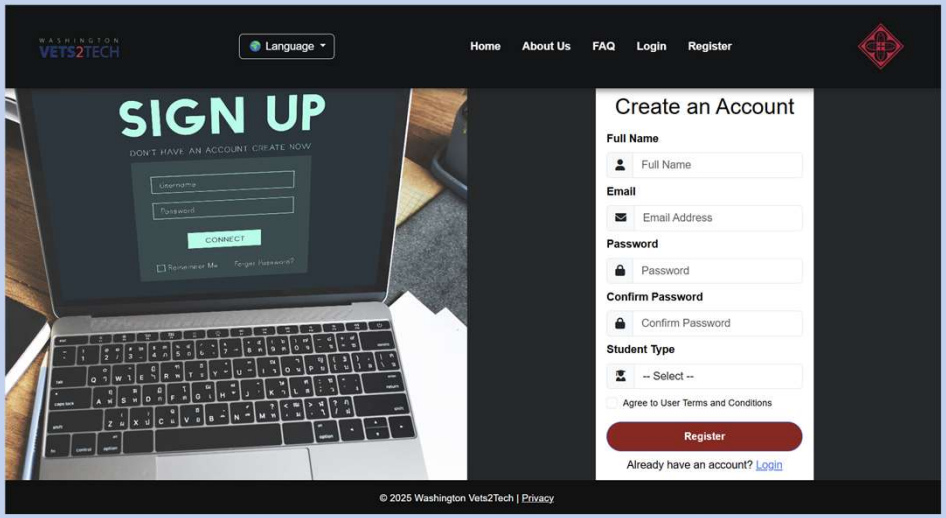
Integration for HTML

```
<html lang="{{ LANGUAGE_CODE }}">
<head>
    {% load static %}
    {% load i18n %}
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}{% trans "Washington Vets2Tech" %}{% endblock %}</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
</head>
```

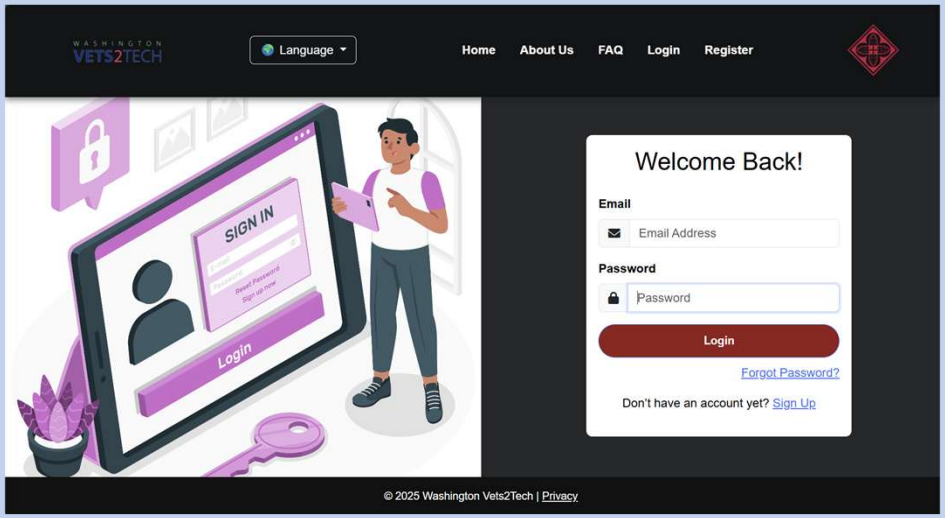
- {% load i18n %} brings framework to the Template html file
- Trans and Block Trans are introduced
- The button's functionality Is embedded into the navigation bar

USER INFORMATION

Sign Up

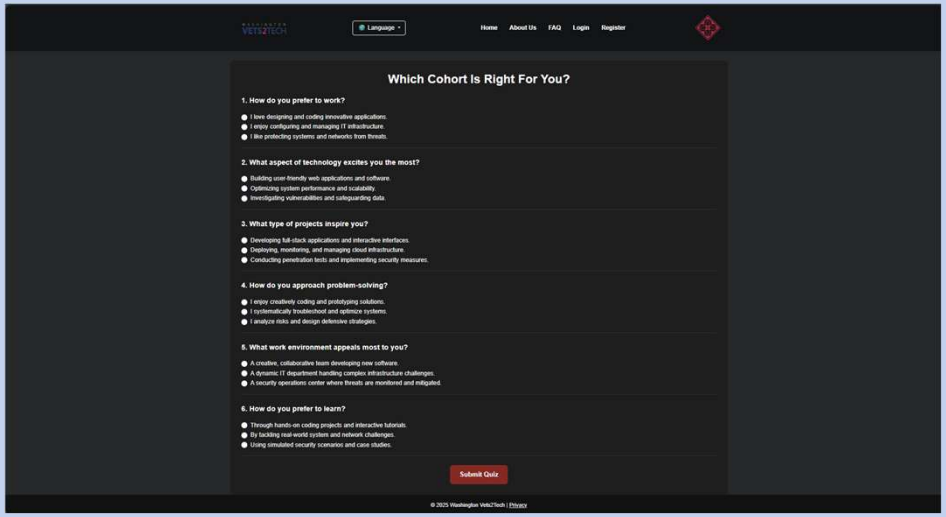


Login

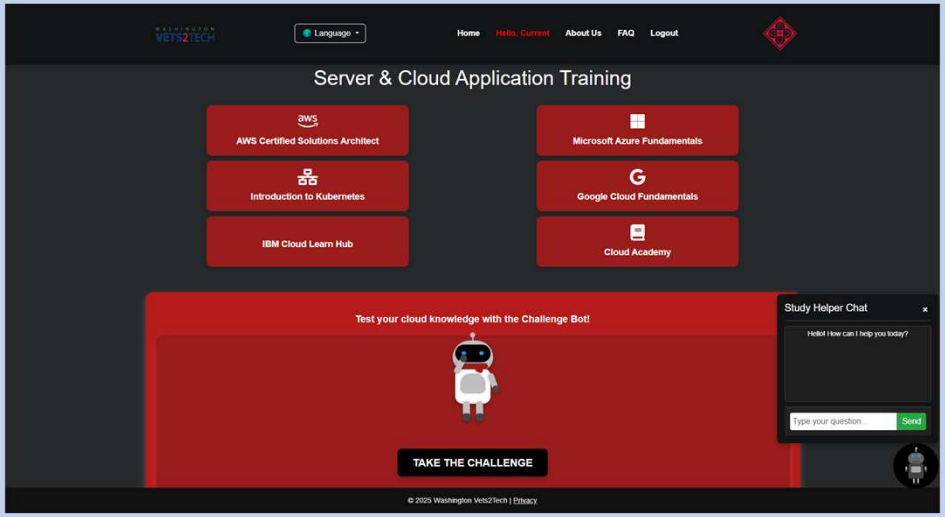


COHORTS

Cohort Quiz



Cohort Materials



ROBOTS

Test Robot Html

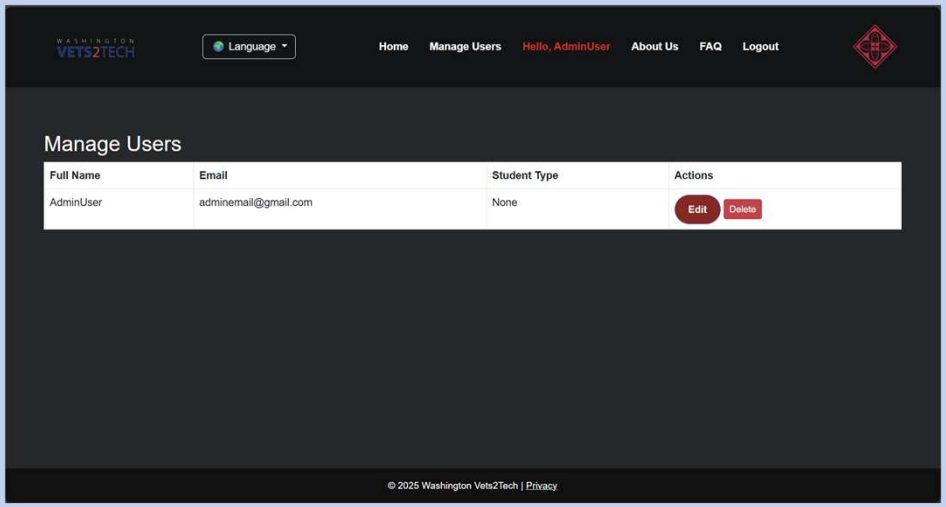
```
<div class="quiz-bot-section">
    <p>Test your cybersecurity skills with the Challenge Bot!</p>

    <!-- Red container with the robot + black button -->
    <div class="challenge-bot-red">
        <div class="my-html-robot">
            <!-- Antenna -->
            <div class="robot-antenna">
                <div class="antenna-ball"></div>
            </div>

            <!-- Helmet + Visor -->
            <div class="robot-helmet">
                <div class="robot-visor">
                    <!-- Eyes -->
                    <div class="robot-eye eye-left"></div>
                    <div class="robot-eye eye-right"></div>
                </div>
            </div>

            <!-- SCARF (neck + tail) -->
            <div class="robot-scarf">
                <div class="scarf-neck"></div>
                <div class="scarf-tail"></div>
            </div>

            <!-- Body with blue chest plate -->
            <div class="robot-body">
                <div class="chest-plate"></div>
            </div>

            <!-- Arms - white arms with blue hands -->
            <div class="robot-arm arm-left">
                <div class="robot-hand hand-left"></div>
            </div>
            <div class="robot-arm arm-right">
                <div class="robot-hand hand-right"></div>
            </div>

            <!-- Legs - white legs with blue feet (both animate) -->
            <div class="robot-leg leg-left leg-wiggle">
                <div class="robot-foot foot-left"></div>
            </div>
            <div class="robot-leg leg-right leg-wiggle">
                <div class="robot-foot foot-right"></div>
            </div>
        </div>

        <!-- Black challenge button -->
        <a href="{% url 'challenge_test' cohortn'cybersecurity' %}" class="btn-quiz-black">Take the Challenge</a>
    </div>
</div>
```

CSS Classes

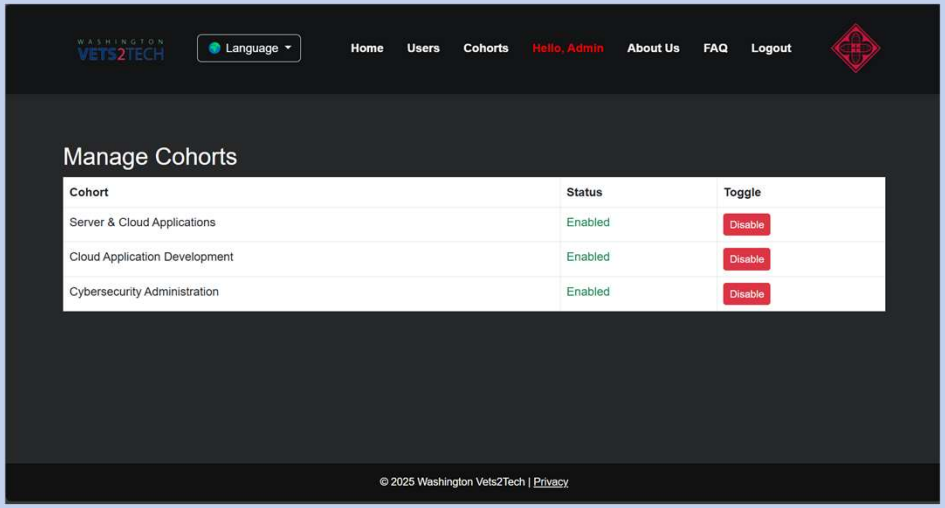
```
.my-html-robot {
    position: relative;
    width: 120px;
    height: 180px;
    animation: jump 1.5s infinite ease-in-out;
}

@keyframes jump {
    0%, 100% { transform: translateY(0); }
    50%      { transform: translateY(-20px); }
}

.robot-antenna {
    position: absolute;
    top: -10px;
    left: 50%;
    transform: translateX(-50%);
    width: 3px;
    height: 15px;
    background: silver;
    border-radius: 2px;
}
.antenna-ball {
    position: absolute;
    top: -6px;
    left: 50%;
    transform: translateX(-50%);
    width: 10px;
    height: 10px;
    background: silver;
    border-radius: 50%;
}

.robot-helmet {
    position: absolute;
    top: 0;
    left: 50%;
    transform: translateX(-50%);
    width: 80px;
    height: 60px;
    background: silver;
    border-radius: 50px 50px 40px 40px;
    box-shadow: 0 4px 8px rgba(0,0,0,0.3);
}
.robot-visor {
    position: absolute;
    top: 8px;
    left: 50%;
    transform: translateX(-50%);
    width: 70px;
    height: 35px;
    background: #000;
    border-radius: 20px;
    display: flex;
    justify-content: space-around;
    align-items: center;
}
```

ADMINISTRATION

User Control Panel



Cohort Control Panel



NEXT STEPS FOR APPLICATION ENHANCEMENT

2025

Apr

- Finalize Core Features:
  - Adaptive Quiz System
  - Ai Chatbot Companion
  - Accessibility Implementation
  - Bilingual Support
  - Data Collection for Analytics
  - Polishing Usability Testing
- Final Client Presentation Preparation

May

- Final Proof of Concept Deliverables:
  - Functional adaptive quiz
  - Working AI chatbot UI
  - High-accessibility frontend
  - Bilingual interface
  - Data export setup analytics
  - Basic user testing and feedback loop
  - Code and documentation clean-up

Future

- Custom Server Roadmap
  - Infrastructure
  - App Deployment
  - Monitoring and Backup
  - Scalability Ready
  - Maintenance

THANK YOU

Engineering Advisory Board

Saint Martin's University
Hal and Inge Marcus School of Engineering

Dr. Dvorak

April 3, 2025