Tech Stack Selection

This survey is designed to understand your technology stack preferences for your capstone project. It's essential to remember that the choices you make here will not only influence the tools, libraries, and frameworks you'll be working with throughout your project but will also be the skills and tools you can add to your CVs. You are encouraged to research alternatives, read about the technologies, and make informed decisions. Choose wisely, as these decisions can shape your professional journey.

Team #: 2

Tech Stack Selection
In this section, you will be presented with a list of popular tools, frameworks, languages, etc. Your task is to:

1. **Select**: Review the available options and choose the option that you're most comfortable with or prefer to utilize for your capstone project. If none of the provided options are relevant to your project, simply write "NA (Not Applicable)" next to the "Your Choice" section.
2. **Rationalize**: Provide a brief explanation for your choice, detailing the reasons why you believe this option suits your project best.
3. **Experience**: Indicate whether your team has prior experience working with the selected programming language. If your team has completed projects, coursework, or had other experiences with it, this is your opportunity to showcase the collective proficiency of the group.

1. Programming Languages:
Your Choice: Python
Why did you choose this? It has many libraries and easy integration with tools and environments we want to use.
Do you have experience with this programming language? Yes, all members have some experience.

2. Web Development Frameworks:
Your Choice: Django
Why did you choose this? It has built in forms, authentication option, out of the box admin dashboard options, and templates for front-end. Prevalent in industry, so good to learn.
Do you have experience with this framework? No

3. Mobile Development:
Preferred Choice: N/A
Rationale for Preference: Focus on web app for MVP
Have you worked with this mobile platform before? No

4. **Database Systems:**

Preferred Choice: NoSQL via AWS DynamoDB
Rationale for Preference: DynamoDB integrates seamlessly with other AWS services. It's NoSQL based which offers greater flexibility and scalability with low latency even at scale. Has fast read/write for dynamic features- such as polls, upvoting, points and leaderboards. We don't anticipate particularly complex queries, so this is an acceptable trade off.
Have you worked with this database before? Yes

5. **Cloud Computing**

Preferred Choice: AWS
Rationale for Preference: Extensive ecosystem with many services that will be helpful: User accounts and authentication (Cognito, OAuth), database (DynamoDB), asset storage (S3), event driven triggers (lambda, event bus), ML support (Comprehend), frontend to backend connection (API Gateway).
Have you worked with this cloud platform before? Yes

6. **Front-end Technology:**

Preferred Choice: Django Templates
Backup Choice: HTML/CSS/JavaScript - we have some experience in this. Planned pivot if templates don't work out.
Rationale for Preference: We decided to use Django as our framework, so using Django templates for frontend development seems logic.
Have you worked with this front-end technology before? No

7. **UI/UX Design**

Preferred Choice: Figma
Rationale for Preference: It's widely used and a good tool for us to learn/demonstrate. It's cloud based and easy to collaborate.
Have you worked with UI/UX design tools before? Yes

8. **Back-end Technology:**

Preferred Choice: Django
Rationale for Preference: Django works for frontend and backend, simplifying our stack.
Have you worked with this back-end technology before? No

9. **Cloud Services**

Preferred Choice: Compute = Elastic Beanstalk (hosting Django),
Lambda and/or Event Bus (event driven actions)
Storage = S3
DB = DynamoDB
ML = Comprehend
Security and Identity = Cognito, IAM, security groups
Rationale for Preference: Elastic Beanstalk automates provisioning and scaling for EC2, Lambda and Event Bus provide event triggers (which helps with dynamic content, still working out

between these choices), S3 is straightforward for storage, DynamoDB is NoSQL based offering better scaling and flexibility, Cognito provides login, IAM separates residents from city admin, security groups provide firewalls for resources
Have you worked with this cloud platform before? Yes

10. Service Protocols
Preferred Choice: RESTful
Rationale for Preference: It's simple, fast, and prevalent. It works with standard HTTP methods (GET, PUT, etc) which aligns well with CRUD operations. Most of our users needs (submitting polls, posting topic threads, etc fall into this arena).
Have you worked with this protocol before? No

11. **User Authentication Methods**
Preferred Choice: OAuth
Rationale for Preference: It works well with AWS services, can be enforced with IAM. No password sharing between services and has built in token expiration
Have you worked with this method before? Yes

12. Data Privacy and Security
Preferred Choice: AWS services

- SSL/TLS - baked in to API Gateway
- Encryption algorithms (e.g., AES) - Baked in to S3 and RDS
- VPNs - Manage through VPC in AWS
- Firewalls - Use security groups for EC2/Elastic Beanstalk
- Monitoring - Cloudwatch

Rationale for Preference: Strong support exists in AWS ecosystem.
Have you worked with this method before? Yes

Additional Tech Resources Needed:
Please list any specific technologies, tools, or resources you anticipate needing for your project:

1. AWS CodePipline / GitHub Actions to automate builds, tests, deployments
2. Pytest (python) or unittest (django)