**Proposal**

***Features of  Back-End Server***

* Use of REST API: This enables the making of client applications more secure and easy. And also Less Troublesome. More forms of client applications can be built easily and work seamlessly with all clients and the servers.
* The API Blocks Complete access to the Database from the client. Provides basic indirect access for storing responses and access to the questions by the client.
* **Containerization** of infrastructure: the back-end is containerized so that we can achieve **high availability, auto-scaling, redundancy, security, etc.** Features supported by almost every cloud provider. Containerization also ensures the security of the infrastructure. This will save a lot of costs and increase the reliability of the infrastructure. Containerized applications also increased the development cycle by using CI/CD pipelines. And automates Testing and deploying and updates/rollbacks, patching processes.
* **Cloud-Ready:** The development build is deployed using server-less Functions on Vercel but the Production build will be deployed on Kubernetes which again is supported by all major cloud Providers.
* Non-Relational and Containerized Database helps to scale the platform and saves costs and hassle.

Proposal *Features of  Back-End Server*

* Use of REST API: This enables the making of client applications more secure and easy. And also Less Troublesome. More forms of client applications can be built easily and work seamlessly with all clients and the servers.
* The API Blocks Complete access to the Database from the client. Provides basic indirect access for storing responses and access to the questions by the client.
* **Containerization**of infrastructure: the back-end is containerized so that we can achieve **high availability, auto-scaling, redundancy, security, etc.** Features supported by almost every cloud provider. Containerization also ensures the security of the infrastructure. This will save a lot of costs and increase the reliability of the infrastructure.
* **Cloud-Ready:**The development build is deployed using server-less Functions on Vercel but the Production build will be deployed on Kubernetes which again is supported by all major cloud Providers.
* Non-Relational and Containerized Database helps to scale the platform and saves costs and hassle.
* **CI/CD**: Integration of CI/CD pipelines Automates the build and Deployment process. Containerized applications increases the development cycle by using CI/CD pipelines. And automates Testing and deploying and updates/rollbacks, patching processes.

Future plans:   
Production Deployment on Kubernetes Engine on AWS/Azure.  
API authentication, more safety and security measures.