**Software Architecture Report**

1. What the purpose of the software project is:
   1. It’s a social media application meant to connect people around the globe.
   2. It offers capabilities similar to other popular social media apps, such as creating posts, uploading pictures, leaving comments, liking content, getting recommendations, chatting, sending follow requests etc.
2. Guides on how to:
   1. clone repository, make sure you have docker installed and run startup.sh script
   2. npm install + npm build
   3. just push to production branch on github
   4. Division into components, services and models for front end and controllers, routers and services for backend
3. Application entry points:
   1. Redis database for session management and Postgres database for all entities in the application
   2. Pictures and text content provided by users
   3. Configuration files for github actions, typescript config files, package.json
4. High level diagrams of the architecture
   1. User/data journeys
   2. Most valuable output
5. Deployment plan
   1. We utilized a DigitalOcean Droplet (a Virtual Machine farm located in Frankfurt) to deploy both our FrontEnd and BackEnd solutions. With the assistance of Nginx and its valuable reverse-proxy functionality, we successfully made our product accessible to all. Additionally, Nginx enables us to scale this project using its load balancing system
   2. We implemented the CI/CD pipelines using Github Actions. The action is triggered when code is being pushed to the production branch. The FrontEnd Angular solution is built and stored as artifacts on Github's servers before deployment via SFTP. The BackEnd code is uploaded to the Virtual Machine (VM) via SFTP. Subsequently, the action logs into the VM and executes scripts to initiate the site
6. Description of the QA process
   1. Integration tests that ensure normal usage flows work correctly (user login, signup, create post etc.)
7. External dependencies included in the project
   1. ChatGPT API
   2. socket.io, express, knex, bycrypt