|  |  |
| --- | --- |
| Note! | |
| Ziel  Beschreibung der Ergebnisse | Ein Tool für das Management von Notizen, die in Gruppen organisiert sind. Jede Notiz und Gruppe sollte durch das Web-GUI verwaltet werden können. Das System sollte es Benutzern auch ermöglichen, Profilbilder hochzuladen und sich über ein benutzerdefiniertes Login-System anzumelden. |
| Projektauftrag  Beschreibung in 5-6 Sätzen, was im Rahmen dieses Projekts umgesetzt werden soll. | Das Ziel ist es, einen webbasierten Notizen-Management-Service zu entwickeln, der den Benutzern die Erstellung und Verwaltung von Notizen in organisierten Gruppen ermöglicht. Das System wird ein benutzerdefiniertes Anmeldesystem (Email + Passwort) enthalten und die Möglichkeit bieten, Profilbilder hochzuladen. Jede Notiz wird Optionen zur Bearbeitung und zum Löschen direkt im Web-Interface haben. Neben der funktionalen Anforderung wird besonderer Wert auf die Benutzerfreundlichkeit und ein ansprechendes Design gelegt. Im Kern steht die Anwendung von Best Practices wie CRUD-Operationen und die Nutzung der Django ORM für die Datenbankanbindung. |
| Erwartete Schwierigkeiten  Wo können Probleme auftreten? Wo bestehen Unsicherheiten? Was sind die kompliziertesten Arbeiten an diesem Projekt? | Die Implementierung eines benutzerdefinierten Authentifizierungssystems kann kompliziert sein. Das Design der Datenbank kann auch eine Herausforderung darstellen, insbesondere bei der Planung der Beziehungen zwischen Notizen und Gruppen. Die Implementierung eines ansprechenden, benutzerfreundlichen Designs kann ebenfalls eine Herausforderung darstellen. |
| Techniken, Frameworks und Tools  Welche Arbeitsmittel/Programmiersprachen können bzw. sollen für dieses Projekt eingesetzt werden? | Python, Django, HTML, CSS, JavaScript, Django Template Language, PostgreSQL |
| Bemerkungen  Hinweise, Links, Anmerkungen, Skizzen, Wissenswertes, Beispiele, … | Der Hauptfokus dieses Projekts liegt darauf, CRUD-Operationen effektiv zu implementieren und eine Datenbank unter Verwendung des Django ORM einzubinden. Ausserdem legen wir grossen Wert auf gute Praktiken wie die MTV (Model-Template-View) Architektur, die Verwendung der Django Template-Sprache für das Front-End-Design und das Einhalten des DRY (Don't Repeat Yourself) Prinzips, um den Code sauber und wartbar zu halten. |
| Autor/-in  Vor- und Nachname | Pavlo Bohulov |