Task Delivery Form

Task Number	Task Description	Group Members					Status of Completion (%)
		Arthur Milner	Rohaan Aslam	William Barnes	Chaya Moore	Seif Mansour	
	Create a comprehensive product backlog	√	√	√		√	100
	Ensure backlog items have appropriate priorities			√		√	100
	Ensure backlog items have appropriate story points			√		√	100
	Ensure backlog items have comprehensive sub-tasks			√		√	100
	Decide which backlog items will be part of the next SPRINT	√	√	√		√	100
	Create schedule of delivery/Gantt chart for the project					√	100
	Create the database schema.			√			100
	Insert sample data into the tables.			√			100
	Implement database indexing and optimization.			√			100
	Create UniHub logo	√	√				100
	Create initial Django project	√					100
	Create a DockerFile for Django	√					100
	Configure MySQL in docker-compose file	√					100
	Configure Django in docker-compose file	√					100
	Configure MailHog/SMTP server in docker-compose file	√					100
	Ensure Django, MailHog and MySQL containers are communicating	√					100
	Create a React + Vite project		√				100
	Install React dependencies		√				100
	Create a DockerFile for React		√				100
	Configure React in docker-compose file		√				100
	Configure polling in Vite for hot reload		√				100
	Ensure React communicates with Django		√				100

	Create custom Djoser-JWT serializers to appropriately handle JWT tokens	√					100
	Create custom Djoser-JWT views to store the refresh token in httponly cookies	√					100
	Create permissions classes and the views to test them	√					100
	Setup AuthContext to manage authentication state		√				100
	Attach JWT to all API requests for auth		√				100
	Create & Style LoginPage from which users can login and authenticate to use the application with their correct tokens		√				100
	Test Login Page		√				100
	Implement RBAC components for role-based routing		√				100
	Ensure session persistence across page reloads		√				100
	Ensure unauthorised requests are retried with fresh access token		√				100
	Configure Djoser/Django email activation email settings	√					100
	Add custom email activation template for the email's content to point to frontend URL		√				100
	Create & Style AccountActivationPage from which users are redirected to from e-mail for account activation		√				100
	Test Account Activation Page		√				100
	Configure Djoser/Django password reset email settings	√					100
	Add custom password reset template for the email's content to point to frontend URL	√					100
	Create & Style PasswordResetPage from which users can enter their email and request a password reset link/email	√					100
	Create & Style PasswordResetConfirmPage from which allows users to confirm their new password from the password reset link	√					100
	Test Password Reset Pages	√					100

	Create custom user and user manager models, and abstract user create serializer to use email instead of username as the login field, and automatically detect a user’s university	√					100
	Create & Style RegisterPage from which users can enter their names, email and password to create an account.		√				100
	Test Registration Page		√				100
	Create & Style Navigation Menu Component		√				100
	Create & Style ProfilePage from which users can view user information and update their own info	√					100
	Test Profile Page	√	√				100
	Create update profile endpoint	√					100
	Create get profile info endpoint	√					100