

Project Assignment

.NET Academy

Project Future 6

Powered by







FundRaiser - Crowdfund Project



Business Case Description

You have been given the opportunity to work for a big client. The task at hand is to **create a crowdfunding platform** (like <u>KickStarter</u> or <u>Indiegogo</u>) for the Web where one, **the Project Creator**, could share their wildest idea and others, **the backers**, could contribute financially to see it happening.

The Project Creator promotes his idea through a custom **project profile page** where he can **add** a **project title**, **Description**, **photos**, **videos** and **post status updates**, whenever necessary.

In addition to that, the Project Creator should be able to track the project's financial progress through his Dashboard page. To motivate backers to contribute, he/she should offer a number of funding packages where each package is accompanied by a reward to the backer.

A backer should be able to browse all Projects and invest on any one he wishes. He can find Projects either through plain text search or by browsing by category, and he should be able to view the site's trending projects. He/she should also be able to view the projects he funded and observe their progress.

Your client wishes to create a proof of concept of the application, with separate screens for the Project Creator and the backer (authorization not required for the proof of concept), enabling him to showcase it's usage to <u>Angel Investors</u> to receive funding.

Project Requirements

Functional requirements

- 1. Add reward packages (Project creator)
- 2. Fund a project by selecting a reward package (backer)
- 3. View list of available projects (backer)
- 4. View Project's page (backer & Project Creator)
- 5. View a list of funded projects (backer & Project Creator)
- 6. Provide a REST API allowing third party integrations.

Non-functional requirements

- 1. Use EF Core
- 2. Use DI
- 3. Use Coding Conventions
- 4. Source code delivered on GitHub
- 5. Application demonstrated on Azure
- 6. Application of coding and architectural standards

Criteria of assessment

Backend development

- 1. Definition of a Model
- 2. Implementation of Persist Model (using EF Core)
- 3. Definition of project's Services in Interfaces.
- 4. Establishment of project's functionality.
- 5. Exposure of Services to an MVC project

Frontend development

- 1. Usage of Ajax for all form submissions
- 2. Usage of Bootstrap for UI elements
- 3. Handle of double clicks in forms
- 4. Usage of JavaScript coding conventions
- 5. Display of validation messages

Git usage

- 1. Each team member's contribution should be clearly depicted with commits
- 2. Provision of clear and descriptive commit messages
- 3. Usage of branches for each feature development

Azure

- 1. The web application should be deployed to Azure.
- 2. The application's SQL database should be hosted on Azure.

