İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

SOFTWARE DEVELOPMENT DESIGN AND PRACTICE
Project Proposal

Project Name: Bringy
Group Name:

Group Members:

Student Number	First Name	Last Name	Email
1306230116	Sohail Mohamed	Elskhawy	s.elskhawy@ogr.iuc.edu.tr
1306220129	Ebrahim	Alkridi	………………
……………….	……………..	……………..	………………

We Are Planning to divide the app to pages and each page with its functionality among the group members. For example The Login page will be divided into 3 parts and each group member will be working in a part. In That way we all gain experience in frontend and backend and be more productive and each week will be given a report of group member work and progress

Project Description:

“Bringy” is a web-based application of an idea that look like **(Like Getir or Cepte Şok)** There will be a login page for the customer after the customer logs in and the customer will see the products. The customer will be able to filter them, sort them, add to basket and edit the basket, and after the customer confirms his purchases, the customer will be able to checkout and see his order in the purchases page. Our Team is not just copying Getir. We have an idea of integrating the app with AI where the customer will be able to talk to a chatbot that will help him add the items he needs with a click of a button. Here is an example:

Customer: Hello, I want to make Garlic Butter Pasta for dinner and I have only 200 tl

Bot: Sure Here is what you need:

- 200g Ankara Spagetti Makarna: 24 tl
- Sütaş Tereyağı 125 gr: 70 tl
- Sarımsak 100 g: 20 tl

Total is 114 tl. Would You Like To Add the items to the basket ?

2 buttons will show up to prompt the user (Yes or No) if yes the items will be added to the customers basket if no the customer can change his prompt or make a new one

From the admin side there will also be a login page for the admins and there will be only 2 pages one page to create, read, update, and delete the products, and suppliers.
The second page is to track the orders and change the order status to delivered and print a receipt of the order only admins will be able to see the products and order pages and the rest of staff (the delivery guy for example) will only see the undelivered orders

Software Requirements Specifications:

Functional Requirements

- **User Authentication:**
  Admins should be able to login and log out
  Customers should be able to register, login and log out

- **User Authorization:**
  Customers will be able only to read the products in home page
  Admins will be able to (add,edit,read,delete) products in the admin's products page and (edit and read) the orders in the admin's orders page
  Staff will be able to change undelivered order status in the admin's orders page

- **Email Verification:**
  After the customer register for the first time he will receive an email with a code to verify it

- **Product (Adding, Editing, Reading, and Deleting):**
  Admins should be able to add products name, price, image, supplier, category, subcategory
  Admins should be also able to update and delete product
  Customers will be able to read the products in customer's home page

- **Product Filtering, Sorting, Searching**

- **Adding Product To Basket:**
  Customer should be able add products to the basket and also change quantity or remove item

- **Checkout:**
  Customer should be able to checkout after adding the items to the basket
  And see his order after confirming

- **AI Chatbot:**
  the customer will be able to talk to a chatbot that will help him add the items he needs with a click of a button

- **Send Email:**
  When an order is done by a customer an email notification will be sent to admins email
  When Order Status is updated we will notify the customer

- **Change Order Status:**
  Admins can change order status when order added to the database

Non-Functional Requirements

Performance: The system shall respond to the customer and admins requests within 2 or 3 seconds.

Security: The system shall protect user data and prevent unauthorized access.

Usability: The system shall be intuitive and easy to use and comfortable looking.

Maintainability: The codebase shall be well-structured and easy to maintain so we can make future updates.

Tech-Stacks (MERN Stack):

- **Frontend**:
    - Reactjs for building user interface
    - CSS For Component styling
- **Backend**:
    - Nodejs with Express.js to build RESTful API
    - MongoDB For Database Storage
    - OpenAI API for building the chatbot
    - Firebase Authentication
    - Mailjet's API: For Email Sending

URL of the GitHub repo

https://github.com/SohailElskhawy/Bringy.git

GIT Commands and its explanation

```
// initialize the git repo
git init ()

// add readme.md file with text header
echo "# Bringy Delivery App" > README.md

// add readme.md to staging area
git add README.md

// create a commit with a comment of the steps done
git commit -m "Initial commit: Project setup and README"

// link the local repo to remote repo on github
git remote add origin https://github.com/SohailElskhawy/Bringy.git

// rename the current branch to main
git branch -M main

// pushes the main branch to the remote repository and sets it as upstream branch
git push -u origin main
```