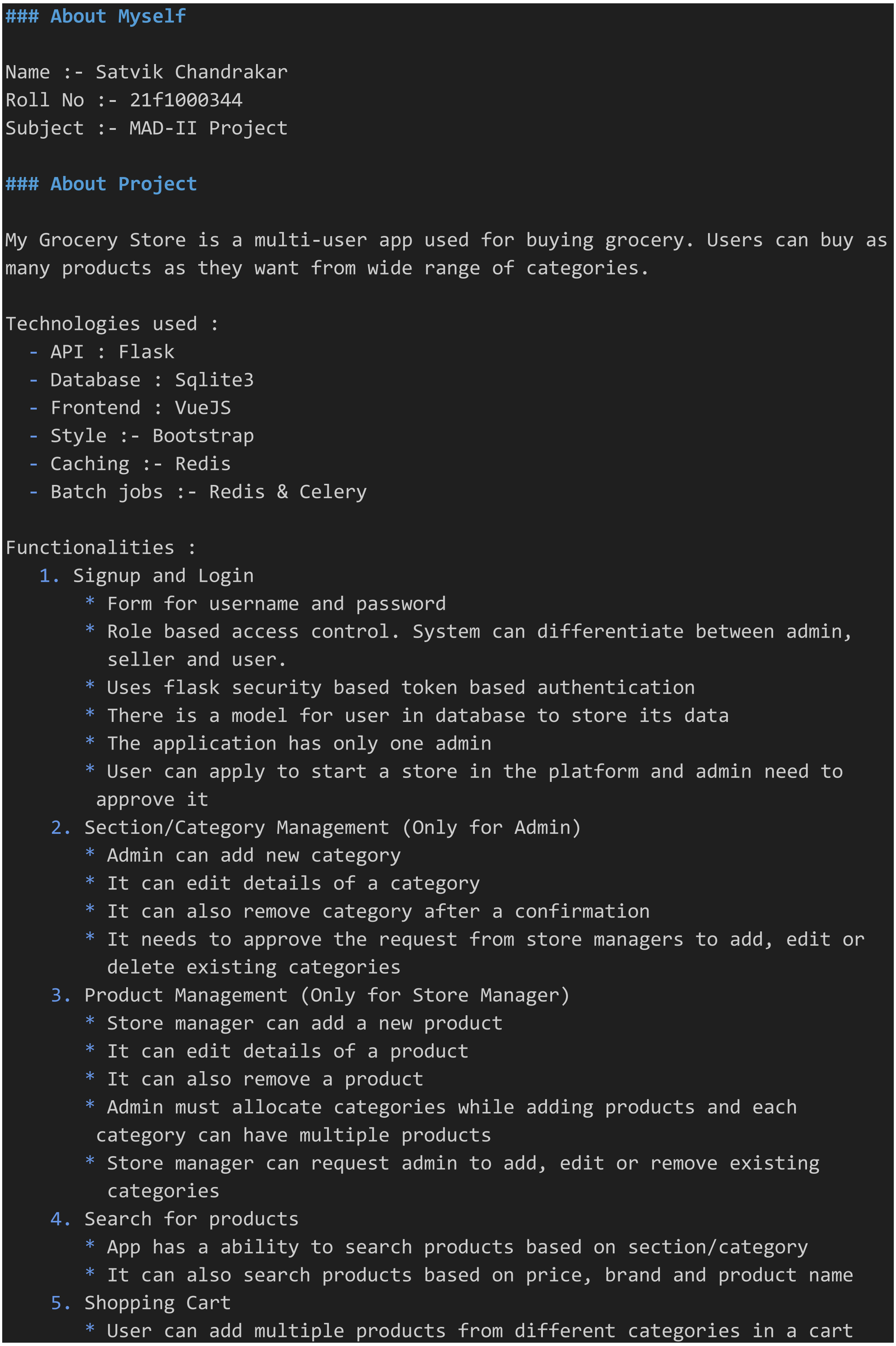
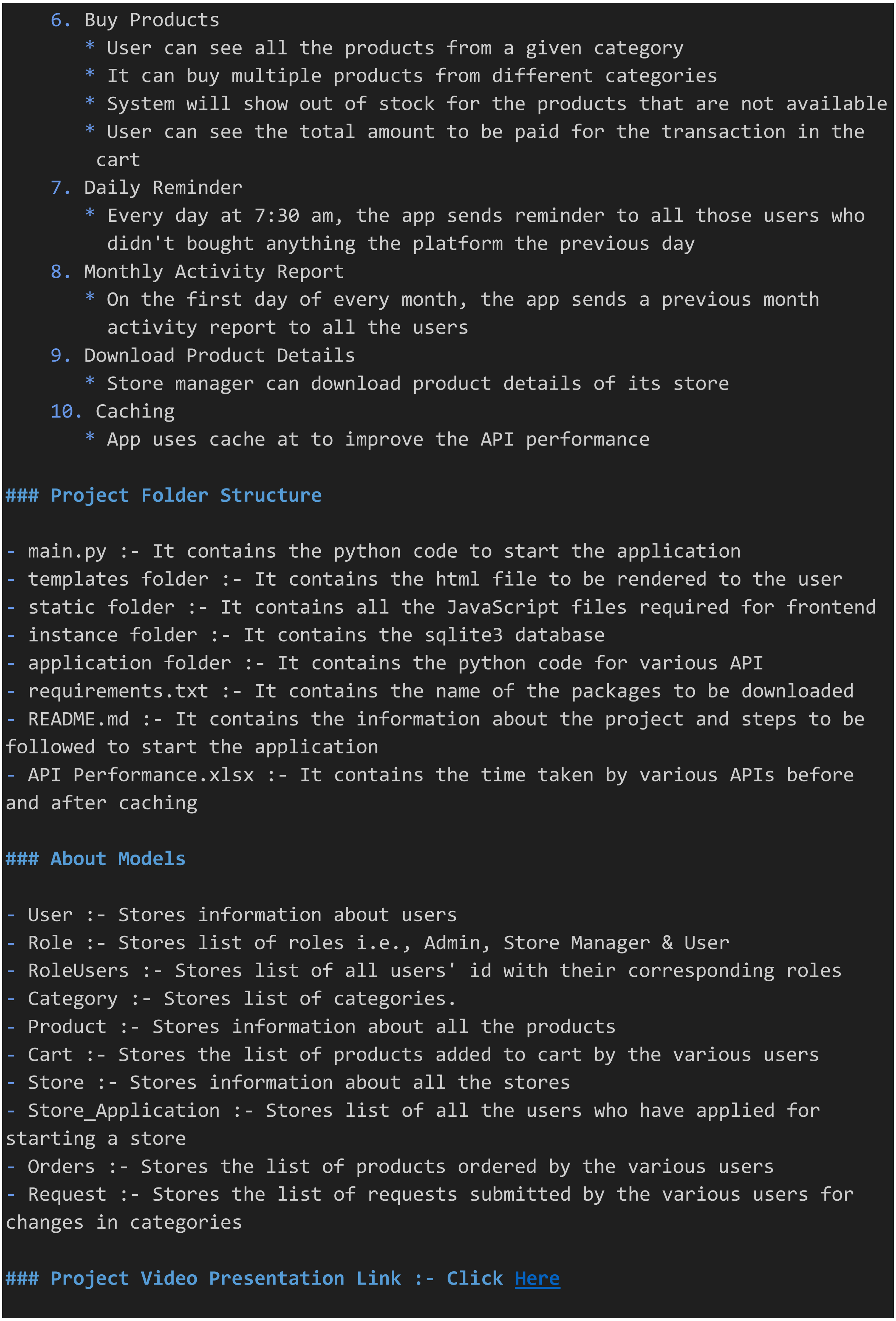


```
### About Myself

Name :- Satvik Chandrakar
Roll No :- 21f1000344
Subject :- MAD-II Project

### About Project

My Grocery Store is a multi-user app used for buying grocery. Users can buy as
many products as they want from wide range of categories.

Technologies used :
  - API : Flask
  - Database : Sqlite3
  - Frontend : VueJS
  - Style :- Bootstrap
  - Caching :- Redis
  - Batch jobs :- Redis & Celery

Functionalities :
   1. Signup and Login
       * Form for username and password
       * Role based access control. System can differentiate between admin,
         seller and user.
       * Uses flask security based token based authentication
       * There is a model for user in database to store its data
       * The application has only one admin
       * User can apply to start a store in the platform and admin need to
        approve it
    2. Section/Category Management (Only for Admin)
       * Admin can add new category
       * It can edit details of a category
       * It can also remove category after a confirmation
       * It needs to approve the request from store managers to add, edit or
         delete existing categories
    3. Product Management (Only for Store Manager)
       * Store manager can add a new product
       * It can edit details of a product
       * It can also remove a product
       * Admin must allocate categories while adding products and each
        category can have multiple products
       * Store manager can request admin to add, edit or remove existing
         categories
    4. Search for products
       * App has a ability to search products based on section/category
       * It can also search products based on price, brand and product name
    5. Shopping Cart
       * User can add multiple products from different categories in a cart
```



```
    6. Buy Products
       * User can see all the products from a given category
       * It can buy multiple products from different categories
       * System will show out of stock for the products that are not available
       * User can see the total amount to be paid for the transaction in the
        cart
    7. Daily Reminder
       * Every day at 7:30 am, the app sends reminder to all those users who
         didn't bought anything the platform the previous day
    8. Monthly Activity Report
       * On the first day of every month, the app sends a previous month
         activity report to all the users
    9. Download Product Details
       * Store manager can download product details of its store
    10. Caching
       * App uses cache at to improve the API performance

### Project Folder Structure

- main.py :- It contains the python code to start the application
- templates folder :- It contains the html file to be rendered to the user
- static folder :- It contains all the JavaScript files required for frontend
- instance folder :- It contains the sqlite3 database
- application folder :- It contains the python code for various API
- requirements.txt :- It contains the name of the packages to be downloaded
- README.md :- It contains the information about the project and steps to be
followed to start the application
- API Performance.xlsx :- It contains the time taken by various APIs before
and after caching

### About Models

- User :- Stores information about users
- Role :- Stores list of roles i.e., Admin, Store Manager & User
- RoleUsers :- Stores list of all users' id with their corresponding roles
- Category :- Stores list of categories.
- Product :- Stores information about all the products
- Cart :- Stores the list of products added to cart by the various users
- Store :- Stores information about all the stores
- Store_Application :- Stores list of all the users who have applied for
starting a store
- Orders :- Stores the list of products ordered by the various users
- Request :- Stores the list of requests submitted by the various users for
changes in categories

### Project Video Presentation Link :- Click Here
```