
Implementation Document

for

DigiCampus

Version 1.0

Prepared by

Group #: 1

Dishay Mehta	200341
Abhishek Pardhi	200026
Samarth Arora	200849
Ankur Kumar	200140
Shashwat Gupta	200923
Ananya Agrawal	200117
Girik Maskara	200387
Sarthak Kohli	200886
Aayush Kumar	200008
Aryan Vora	200204

Group Name: Achievers

dishaymehta952@gmail.com
abhipardhi326@gmail.com
samartharora03@gmail.com
ankurk04@gmail.com
guptashashwatme@gmail.com
ananyaagrawal704@gmail.com
girikmaskara552002@gmail.com
sarthak811kohli@gmail.com
aayushk0072@gmail.com
aryanvora23@gmail.com

Course: CS253

Mentor TA: *Aman Aryan*

Date: 21-03-2022



CONTENTS	II
REVISIONS	II
1 IMPLEMENTATION DETAILS	1
2 CODEBASE	2
3 COMPLETENESS	3
APPENDIX A - GROUP LOG	6

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Dishay Mehta,Abhishek Pardhi,Samarth Arora,Ankur Kumar,Shashwat Gupta,Ananya Agarwal,Girik Maskara,Sarthak Kohli,Aayush Kumar,Aryan Vora	Implementation Document created	20/03/22

1 Implementation Details

Backend Framework: Django

Benefits over other frameworks-

1. MVT architecture- The Model helps handle databases and lets developers create, read, update, and delete objects in the original database. The View fetches data from a model, accepts HTTP requests, applies logic provided by Python class and provides HTTP responses to client requests. The templates are HTML pages, which are used to render data and completely handle the User Interface in tandem with CSS and JavaScript parts.
2. Is highly secure.
3. Great set of libraries and powered by Python.
4. Default Admin Panel- reduces development time and has a lot of room for customization.
5. Django shortcut helps render views easily and enables smooth data transfer from backend to frontend.

Backend Language: Python

Benefits over other languages-

1. It is easy to work with and easy to read.
2. It is well acquainted with versatility and it is well designed.
3. It has many web application frameworks.
4. It provides you with a fast return on investment.

Frontend Languages: JavaScript, CSS, HTML

Benefits over other options-

1. JavaScript has the capacity to reduce response time & slow loading of web pages.
2. Easy debugging from web browsers.
3. Compatible with almost all available web browsers.
4. JavaScript loads & executes web pages quickly.
5. Easy access to all data in the entire html file.

Frontend Software used: Bootstrap

Benefits over other options:-

1. Lightweight and customizable.
2. Easy to use.
3. Consistent framework.
4. Easily portable over many operating systems.
5. Good documentation support.

2 Codebase

Our code can be accessed in the github repository: <https://github.com/Samadeol/CS253>

There are several folders provided within the repository to help navigate through the different avenues of the project.

The '**README.md**' file contains basic information about the repository and its use.

The '**BACKEND**' folder contains the implementation of the backend of our software through Django. This folder contains several subfolders.

The '**Login**', '**Mess**' and '**Hall**' folders contain all the necessary files needed to create the application pages for the login, mess and Hall Entry/Exit functionality respectively.

The '**src**' folder is used to link all the applications together. Contents of src folder:

- '**settings.py**'- contains the configurational settings of our django project.
- '**urls.py**'- calls various django views functions when a particular url path is entered.

The '**statics**' folder contains all the assets required by the frontend (excluding the HTML files themselves)

The '**templates**' folder contains the final HTML files for DigiCampus.

The '**db.sqlite3**' file contains the database of users (in the default format as provided by Django)

The '**manage.py**' file is used to run all the modules, and is the standard file used with Django.

The '**HTML_CSS**' folder contains the frontend implementation of our project, through bootstrap and javascript. This contains many subfolders.

The '**.bsdesign**' files contain designs for various web pages designed on bootstrap.

The '**main**' folder contains the exported HTML files and assets from the main bootstrap design of our project.

The '**sign_up**', '**sign in**', and '**verification**' folders contain the exported HTML files and assets from their respective bootstrap designs.

All '**assets**' folders consist of the fonts and styles (implemented through CSS), javascript codes, and images needed to run their respective HTML codes. All of these are present in their own folders.

3 Completeness

The implementation accomplishes most of the points that were mentioned in the SRS Document.

These main functionalities include :

- Hall Entry and Exit System.
- Mess Extras tracking
- QR Login Portal and Administration by various authenticated people.
- Personalised Page for all admins and students.
- Current and previous month expenses along with expenses incurred during the semester for mess extras

The software requirements have been satisfied:

- The frontend and backend have been mainly developed using Bootstrap and Django respectively.
- The WebApp is made platform independent and can be accessed via any device and operating system.

The DigiCampus satisfies various software quality attributes:

- **Availability** : Once online, the app will be available to everyone.
- **Correctness** : The users can view only their own data
- **Maintainability** : Continuous integration enables bug-fixes without downtime
- **Usability**: Simple and easy interface
- **Interoperability**: System interacts with database provided and makes dynamic changes back in the system
- **Reliability**: Less traffic use-cases ensure failsafe implementations

FUTURE DEVELOPMENT PLAN:

- We are currently running on a dummy database. After testing, we might integrate this with the **Pingala** or **OARS** DataBase. Depending on the fields there, we might wish to include more elements.
- The Performance in terms of loading and refreshing can be enhanced. Currently, the software loads within the prescribed time of 5 seconds

Appendix A - Group Log

- 26/02/22-2/03/22: Samarth, Aryan, Sarthak, Shashwat learnt Django. Samarth and Sarthak learnt operating in javascript. Abhishek, Aayush, Girik, Ankur and Dishay learnt Bootstrap and HTML/CSS.
- 03/03/22: Abhishek created a signup and verification page.
- 04/03/22: Abhishek created a mess and profile page. Also designed temporary logo of website.
- 05/03/22: Samarth and Aryan added initial models in Login app. Configured settings of the project to render html pages in Django using Django Template Library.
- 08/03/22: Abhishek made mess page responsive and also made the lists scrollable.
- 09/03/22: Samarth configured style values of mess.html so that they could be activated using JavaScript. Samarth cleaned Backend folder and separated all CSS, Font, JavaScript files into static mode to run on server.
- 13/03/22: Sarthak added functionalities to mess.html and also buttons to Django views.
- 14/03/22: Aryan experimented with ways of QR Code Authentication.
- 14-18/03/22: Aryan worked on Hall Entry/Exit.
- 18/03/22: Aayush, Dishay created the Hall Entry page.
- 16/03/2022: Ankur made JavaScript file to update the dashboard chart, making it responsive.
- 19/03/22: Samarth and Sarthak created HTML view using values from Django database.
- 16/03/22: Shashwat added initial models of email and further configured the settings of the project to render html pages in Django using Django Template Library.

15/03/22:	Samarth and Sarthak made a Javascript file to update mess_extra items in real time.
17/03/22:	Samarth created api views to call database items in JSON format.
17/03/22:	Samarth and Sarthak used REST framework to update/create data from frontend to backend.
17/03/22:	Samarth, Aryan and Sarthak ran server on different IP to test QR code redirecting.
20/03/22:	Sarthak linked data from database to dashboard.html
12/03/22:	Aryan worked on User authentication and Login
11/03/22:	Aryan worked on Profile Pages
14/03/22:	Abhishek changed the color palettes being used in the pages.
15/03/22:	Abhishek created Dashboard page.
16/03/22:	Abhishek made Dashboard and mess page responsive.
20/03/22:	Aryan created README.md
19/03/22:	Shashwat experimented with an email notification system.
08-12/03/22:	Girik created Frontend pages.
16/03/22:	Girik worked on creating the QR code scanner.
18/03/22:	Shashwat and Girik worked on section 3 of the implementation document.
08-16/03/22:	Abhishek created Frontend pages.
15-20/03/22:	Entire team worked on Implementation Document
20/03/22:	Aayush made confirmation page
16/03/22:	Aayush experimented with QR Code scanner implementation
17/03/22:	Aayush wrote section 2 of the implementation document
17/03/22:	Ananya and Ankur finished Section 1 of implementation document

18/03/22: Abhishek left aligned the Nav-bar and made it responsive in each page.

21/03/22: Abhihsek created QR code page.