Implementation Document

for

Hall Buddy

Version 1.1

Prepared by

Group #: 12 **Group Name: CodeMonk**

Ritesh Hans	**220893**	**riteshhans22@iitk.ac.in**
Mridul Gupta	**220672**	**mridulg22@iitk.ac.in**
Rohan Batra	**210868**	**rohanb21@iitk.ac.in**
Krutuparna Paranjape	**210536**	**krutuparna21@iitk.ac.in**
Mrdul Agarwaal	**210632**	**mrdula21@iitk.ac.in**
Apoorv Tandon	**220192**	**apoorvt22@iitk.ac.in**
Tanishq Maheshwari	**221128**	**tanishqm22@iitk.ac.in**
Taneshq Zendey	**221123**	**taneshq22@iitk.ac.in**
Ayush	**220259**	**ayushs@iit.ac.in**
Samarpan Verma	**220943**	**samarpanv22@iitk.ac.in**

Course: CS253

Mentor TA: *Vaibhav Tanwar*

Date: 18/03/2024

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Implementation Details

Programming Language, Framework and Libraries for:

Backend

Hall Buddy primarily utilizes Python programming language. Django, an open-source framework for web development based on Python, serves as the foundation. Employing the model-view-controller architectural pattern, Django was selected due to its robustness, simplicity, reliability, and scalability. Some key advantages of Django are outlined below:

- Django boasts straightforward syntax, facilitating ease of learning and implementation in web development projects due to its Python-based code.
- It comes equipped with its own built-in server, streamlining the development process.
- Security features such as protection against SQL injection, cross-site scripting (XSS) attacks, and a robust user authentication system are inherent to Django, ensuring the creation of secure and dependable web applications.
- With an object-oriented design, Django fosters code reuse and modularity, enhancing development efficiency.
- Its scalability enables efficient handling of high web traffic, accommodating numerous simultaneous users without compromising performance.
- The built-in admin interface simplifies data model management and data viewing for developers.
- Django facilitates rapid development through scaffolding tools that expedite the creation of a web application's basic structure.
- It includes a built-in SQLite database, simplifying setup and usage.
- Promoting code modularity, Django encourages organized and maintainable programming practices.
- The framework benefits from a supportive community, with a multitude of developers contributing to its enhancement and providing assistance through online forums and documentation.

Additionally, the project utilizes various Python and Django libraries, with a comprehensive list available in the GitHub repository.

Frontend

- HTML (HyperText Markup Language) serves as the fundamental markup language for our web pages, ensuring proper display in web browsers.
- It establishes a foundational structure for web content, encompassing text, images, and various media elements.
- CSS (Cascading Style Sheets) is employed to stylize the appearance of our web pages.
- By separating content from presentation, CSS streamlines the maintenance and updating process for our web pages.

Database

- The SQLite Database Engine has been utilized, which is integrated into Django and offers attributes of being compact, swift, and exceptionally dependable.
- It stands as one of the most prevalent and favored relational database engines globally.
- Its structure, based on files, obviates the necessity for an independent database server, rendering it a straightforward and effective option.

2 Codebase

Provide the link to your github repository.
Mention briefly how to navigate the codebase.

Github Repository: https://github.com/mridul-g/New-HallBuddy

Code Structure

HallBuddy makes use of Django which segregates the project into different apps for the various major use cases. This ensures a well structured and easily accessible code. The project has been divided into following small apps:

1. **Guest Room Booking:** It contains implementation of the guest room booking feature provided by HallBuddy. Residents will be able to see availability of rooms, send booking requests, track their requests and view previous bookings.

2. **Complaint Management:** It contains implementation of functionalities related to complaint management features provided by HallBuddy. Residents will have the options of registering a complaint, viewing complaints and upvoting existing complaints.

3. **Announcements:** It contains implementation of functionalities regarding announcement broadcasting service provided to the admin. Admin will be able to display important announcements on HallBuddy.

4. **Cleaning Management System :** It contains a room cleaning log maintained by the student.Student can mark his room-cleaning status and admin can export data of all the rooms.

5. **Login:** It contains implementation of functionalities related to login, i.e login old user, new user registration and reset password.

6. **Home:** It contains implementation of the homepage. Users can access the various functionalities provided by HallBuddy through the home page.Users can also view Hostel Map, Contacts of Hall Administration and a catalogue of the Hall General Store.

7. **My Profile :** It displays the dues of a student. It also has a payment functionality implemented through a razorpay payment gateway api.

Directories

```
└──HallBuddy_Website
    db.sqlite3
    manage.py
    requirements.txt
    
    ├──authentication
    │   admin.py
    │   apps.py
    │   models.py
    │   tests.py
    │   urls.py
    │   views.py
    │   __init__.py
    │
    │   ├──migrations
    │   │   0001_initial.py
    │   │   __init__.py
    │   │
    │   ├──static
    │       Hall_Background.jpg
    │       Hall_Background.png
    │       logo.png
    │       logo2.png
    │
    │   ├──templates
    │       Login.html
    │       OTP.html
    │       Reset_Password.html
    │       Set_Password.html
    │       SignUp.html
    │       student_login.css
    │
    
    ├──Cleaning
    │   admin.py
    │   apps.py
    │   models.py
    │   tests.py
    │   urls.py
```

```
    │   views.py
    │   __init__.py
    │
    ├───migrations
    │   │   0001_initial.py
    │   │   __init__.py
    │
    ├───static
    │   calendar.css
    │
    ├───templates
    │   calendar.css
    │   Calendar.html
    │   calendar_logic.js
    │   Cleaning_Management.html
    │
    │
    ├───Complaints
    │   admin.py
    │   apps.py
    │   models.py
    │   tests.py
    │   urls.py
    │   views.py
    │   __init__.py
    │
    ├───migrations
    │   │   __init__.py
    │
    └───templates
        check.html
        Complaints_hall.html
        Lodge_Request.html
        Past_Request.html
        Pending_Request.html
        script.js
    │
    ├───django_razorpay
    │   admin.py
    │   apps.py
    │   models.py
    │   tests.py
    │   urls.py
    │   utils.py
    │   views.py
    │   __init__.py
```

```
    │   │
    │   ├───management
    │   │   │   __init__.py
    │   │   │
    │   │   ├───commands
    │   │   │   │   dj_razorpay_init.py
    │   │   │   │   __init__.py
    │   │   │
    │   ├───migrations
    │   │   │   0001_initial.py
    │   │   │   __init__.py
    │   │   │
    │   ├───static
    │   │   └───django_razorpay
    │   │       │   hallbuddy1-1100h.png
    │   │       │
    │   │       ├───css
    │   │       │       bootstrap-datepicker.min.css
    │   │       │       bootstrap-select.min.css
    │   │       │       bootstrap.min.css
    │   │       │       bootstrap.min.css.map
    │   │       │
    │   │       └───js
    │   │               bootstrap-datepicker.min.js
    │   │               bootstrap-select.min.js
    │   │               bootstrap.bundle.min.js
    │   │               bootstrap.bundle.min.js.map
    │   │               jquery.min.js
    │   │
    │   ├───templates
    │   │   └───django_razorpay
    │   │           adhoc_payment.html
    │   │           base.html
    │   │           checkout.html
    │   │           manual_transaction.html
    │   │           membership_fee.html
    │   │           payment_status.html
    │   │           transactions.html
    │   │
    │   ├───templatetags
    │   │   │   dj_razorpay.py
    │   │   │   __init__.py
    │   │   │
    │   │
    │
    ├───guestroom
```

```
    │   admin.py
    │   apps.py
    │   models.py
    │   tests.py
    │   urls.py
    │   views.py
    │   __init__.py
    │
    ├───migrations
    │   │   0001_initial.py
    │   │   __init__.py
    │   │
    │
    ├───templates
    │       booking_aprooved.html
    │       booking_pending.html
    │       guestroom.html
    │       guestroom_temp.html
    │       pastbookings.html
    │
    │
    ├───HallBuddy_Website
    │   │   asgi.py
    │   │   settings.py
    │   │   urls.py
    │   │   wsgi.py
    │   │   __init__.py
    │   │
    │
    ├───Home
    │   │   admin.py
    │   │   apps.py
    │   │   models.py
    │   │   tests.py
    │   │   urls.py
    │   │   views.py
    │   │   __init__.py
    │   │
    │   ├───migrations
    │   │   │   0001_initial.py
    │   │   │   __init__.py
    │   │   │
    │   │
    │   ├───static
    │   │       hall_map.png
    │   │
    │   ├───templates
```

```
        Announcements.html
        Announcements_admin.html
        Contact.html
        Home.html
        Map.html
        Shop.html
        Dues.html

    templates
    .DS_Store
    admin_base.html
    Error.html
    package.json
    student_base.html

    static
    .DS_Store
    contact.css
    favicon.ico
    GuestRB.png
    hallmap.png
    page.css
    student_base.css
    style.css
    styles.css

    public
    .DS_Store
    hallbuddy1-1100h.png

    external
    bellfill3136-dwbm.svg
    buildingstore5515-p6yh.svg
    dashboardblack24dp113132-ln5i.svg
    ellipse2123132-5lk-200h.png
    fibsmapmarker3136-kqig.svg
    firsphonecall3136-u7a8.svg
    hallbuddy-200h.png
    line13136-3b7w.svg
    rectangle123131-9n5u-400w.png
    rectangle53136-qddo-200h.png
    rectangle73134-y8t-200w.png
    vector3133-tpso.svg
    vector3134-2hos.svg
    vector3134-8g9f.svg
    vector3134-q2cg.svg
    vector3135-4t7q.svg
```

└──external1
bellfill3136-dwbm.svg
buildingstore5515-p6yh.svg
dashboardblack24dp113132-ln5i.svg
dot2148-3kyl-200h.png
dot2148-mof5-200h.png
dot2148-rf15-200h.png
dot2149-0xpr-200h.png
dot2149-9url-200h.png
dot2149-fnns-200h.png
dot2149-oq7p-200h.png
dot2149-r0p-200h.png
dot2149-uq7b-200h.png
ellipse2123132-5lk-200h.png
email12148-5ts.svg
email12148-b4hc.svg
email12149-77w.svg
email12149-tmg.svg
email12197-6wfi.svg
fibsmapmarker3136-kqig.svg
firsphonecall3136-u7a8.svg
hallbuddy-200h.png
line13136-3b7w.svg
placeholder2148-j52m-200h.png
placeholder2149-5udk-200h.png
rectangle123131-9n5u-400w.png
rectangle13803144-92tj.svg
rectangle13813145-d6qh-200h.png
rectangle13823145-jwlb-200h.png
rectangle13833144-gow-200h.png
rectangle13853146-ct1v-200h.png
rectangle53136-qddo-200h.png
rectangle73134-y8t-200w.png
rectanglei219-5rkm-200h.png
rectanglei219-5rrh-200h.png
rectanglei219-5ws-200h.png
rectanglei219-ejmj-200h.png
rectanglei219-jyka-200h.png
rectanglei219-pn5-200h.png
rectanglei219-vsqc-200h.png
rectanglei219-wlzs-200h.png
rectanglei219-x8u5-200h.png
rectanglei219-zn28-200h.png
vector2148-3tfm.svg
vector2148-4ank.svg
vector2149-lux.svg

vector2149-npu2.svg
vector2197-9wob.svg
vector3133-tpso.svg
vector3134-2hos.svg
vector3134-8g9f.svg
vector3134-q2cg.svg
vector3135-4t7q.svg
whatsappimage20240125at030713144-1jjvf-1400h.png

How to run the application globally

- Clone the Github repository using the following command: git clone
- Run the following commands in the sequence as described below
- **python manage.py migrate** (to pull and integrate new changes in the database model)
- **python manage.py runserver** (runs the application and hosting the server on localhost)

3 Completeness

Provide the details of the part of the SRS that have been completed in the implementation.

Provide the future development plan by listing down the features that will be added in the (may be hypothetical) future versions.

Details of the SRS that have been completed :

- **Login:** A portal for login has been created.
- **OTP :** User receives an OTP before registration
- Complaint management
- Lodging a complaint
- Viewing previous complaint
- Viewing complaints of other users
- Guest room booking
- Selecting from 4 different guest rooms
- Admin has the option to approve/reject the guest room booking
- Both admin and users can view announcements
- User can mark which dates were cleaned which weren't
- Admin can see the whole user database

Status/Future Development plan of features

SRS Features	Status	Future Development Plan
Profile login / registration	Completed	Addition of Profile images and connecting the User Profile with Student Search.
Complaint Management	Completed	Addition of a feedback form after the complaint has been resolved
Guest Room Booking	Completed	Incorporation of Hall Facility available in future for booking.
Announcements	Completed	
Cleaning	Completed	Option of giving an overall review of the cleaning process

Appendix A - Group Log

All the group members were in constant touch with each other and the TA through the WhatsApp group. Suggestions were given by the TA regarding some corrections and elaboration in the architecture diagram, class diagrams, and state diagram. All those suggestions have been seriously incorporated into the document. Work was divided and team-meetings were held regularly.

Date	Members Present	Description
26/2/2024	All Group Members	Distribution of work and Decided the utilities and libraries to use.
	All Group Members	Discussion on frontend progress.
1/3/2024	TA and all Group Members	Addressed and resolved TA's questions and suggestions regarding our software
2/3/2024	All Group Members	Discussion on initial models and views for Login, Signup Pages.
3/3/2024	All Group Members	Discussion on initial models and views for Guest Room Booking.
4/3/2024	All Group Members	Discussion on initial models and views for Complaint Managements.
5/3/2024	All Group Members	Databasing of software.
6/3/2024	All Group Members	Discussion on final models and views for all Pages
7/3/2024	All Group Members	Finalized overall software backend as well as frontend also modified to achieve better results.
8/3/2024	TA and All Group Members	Checked on software functionality, performance, reliability, accessibility, efficiency, correctness, usability, integrity.
10/3/2024	All Group Members	Locating and Fixing bugs in the software.
13/3/2024	TA Group Members	Locating and Fixing bugs in the software.
15w/3/2024	All Group Members	Worked on the Creation of Implementation Document.