IIIT Hostel Management Portal developed using WEB2PY

 $\mathrm{May}\ 5,\ 2015$

Contents

- 1. Brief description of the web application.
- 2. Features
 - Functions
 - UI Screenshots
 - Schema of tables

Part I

Brief Description:

This web app has been developed as a part of our ITWS 2 project work and made using WEB2PY.

IIIT Hostel Management Portal is a simple portal where students can Log In and then Book their rooms before the start of the semester, post complaints, subscribe for newspapers, fix an appointment with a Doctor and even register a Lost or Found entry.

Salient Features:

- Room Booking
- Complaint Entry
- Laptop Registration
- Lost and Found Entry
- Newspaper Subscription
- Feedback

Note

- When booking a room a person can book a room only once and then changes can only be overidden by the Admin.
- Complaints entered by a person are visible to the person who made the entry and the Admin.
- Similar goes for newspaper subscriptions and laptop registration.
- Feedback once given cannot be undone.
- A person must sign up and be logged in to use any of the above services.
- To ensure a person is a reliable source and a student of IIIT he/she must first confirm his/her registration during sign up by clicking on the link sent to their email address.

Part II

Main Features:

- 1. Complaint Here users can add complaints and view complaints.
- 2. **Laptop** Here users have to register their laptops.
- 3. **Medical** To fix an appointment with a doctor or in case of an Emergency contact the Admins.
- 4. Lost&Found To post if an item gets lost.
- 5. Room Allocation Here users can book a room.
- 6. **Newspaper** Users can subscribe for a newspaper.
- 7. Feedback Users can give their feedbacks.

Functions:

These are present in the default py file in the Controllers.

- 1. write_complaint(): This function creates a form to add the complaint to the complaint table.
- 2. view_complaint(): This function displays the contents of the complaint table of the user currently logged in or all the complaints of the registered users to the Admins.
- **3.** my_complaint():This function returns a grid to display a particular complaint.
- 4. medical_request(): This function creates a form to schedule an appointment with a doctor by specifying the convenient time.

 In case of an emergency a user can click on the Emergency button and this will automatically send a mail to the admins informing them of an emergency.
- 5. my_requests(): This functions returns a grid which displays the medical requests of the user currently logged in or all the requests of the registered users to the Admins.
- **6. view_request():**This functions returns a grid which displays a particular medical request.
- 7. laptop_entry(): This function creates a form to add the details of a

student's laptop to the Laptop table.

- 8. my_laptops():This function returns a grid to view the details of the user's laptop entries or the laptop entries of all the registered users by the Admins.
- **9.lostandfound_entry():** This function creates a form to add a lost entry and this is added to the table Lost.
- 10.view_lostandfound(): This returns a grid containing the details of the table Lost where the registered users can view all the lost entries and can make comments.
- 11.view_lost():To view a particular lost entry and comment.
- 12.room_allocation(): This function creates a form to select room no from any floor and any block D,E or OBH. Here each user can select a room only onceand it can be changed only by the Admin.
- **13.newspaper():**This function creates a form where users can subscribe for a newspaper.
- 14.my_newspapers():This function creates a grid where users can view there newspaper subscriptions and Admin can view the subscriptions of all the registered users.
- 15.feedback_entry(): This function creates a form to enter feedback.
- **16.view_entry():** This function returns a grid to enable the Admin to view the feedback given by the users.
- 17.adminship(): This function defines the adminship of the users like Superadmin, Admin or regular users.

Schema of the Tables:

These tables are present in db.py file in Models

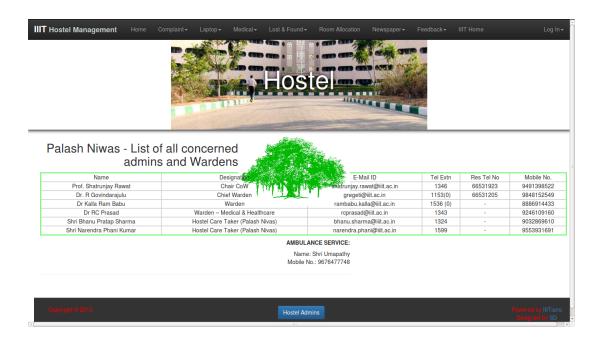
```
1. \ db. define\_table ('complaint', Field ('roll', 'integer', writable=False, requires=[IS\_INT\_IN\_RANGE (2000)] \\ Field ('category', requires=IS\_IN\_SET (CAT)), Field ('subject', requires=IS\_NOT\_EMPTY ()), Field ('description', 'text', requires=IS\_NOT\_EMPTY ()), Field ('image', 'upload', uploadfield='myblob') \\ Field ('myblob', 'blob'), auth. signature)
```

- $2. \ db. define_table ('Laptop', Field ('roll', 'integer', writable=False, requires=[IS_INT_IN_RANGE (20070Field ('category', requires=IS_IN_SET (brand)), Field ('Serial', requires=IS_NOT_EMPTY ()), Field ('description', 'text'), Field ('image', 'upload', uploadfield='myblob'), Field ('myblob', 'blob'), auth.signature)$
- 3. db.define_table('Lost', Field('roll', 'integer', writable=False, requires=[IS_INT_IN_RANGE(20070100 Field('Item', requires=IS_NOT_EMPTY()), Field('Place_Where_Lost', requires=IS_NOT_EMPTY()), Field('description', 'text', requires=IS_NOT_EMPTY()), Field('image', 'upload', uploadfield='myblob') Field('myblob', 'blob'), auth.signature)
- 4. db.define_table('comm', Field('complaint', 'reference complaint'), Field('body', 'text'), auth.signature)
- 5. db.define_table('comm2', Field('Lost', 'reference Lost'), Field('body', 'text'), Field('founditem', 'boolean'), auth.signature)
- 6. db.define_table('med', Field('roll', 'integer', writable=False, requires=[IS_INT_IN_RANGE(20070100 Field('description', 'text'), Field('doctor', requires=IS_IN_SET(MED)), Field('availablefrom', 'time', requires=IS_TIME(), default='6:00'), Field('emergency', 'boolean'), auth.signature)
- 7. db.define_table('medcomm', Field('med', 'reference med'), Field('body', 'text'), auth.signature)
- 8. db.define_table('news', Field('roll',writable=False), Field('newspaper',requires=IS_IN_SET(NEWSField('validfrom','date',default=request.now.date), Field('validupto','date'), auth.signature)
- 9. db.define_table('obh', Field('floorno',writable=False,requires=IS_IN_SET(FLOOR)), Field('room',writable=False), Field('person',requires=IS_IN_DB(db,db.auth_user.Rollno)),)
- $10. \ db.define_table('d', Field('floorno', writable=False, requires=IS_IN_SET(FLOOR)), \\ Field('room', writable=False), Field('person1', requires=IS_IN_DB(db, db. auth_user.Rollno)), \\ Field('person2', requires=IS_IN_DB(db, db. auth_user.Rollno)))$

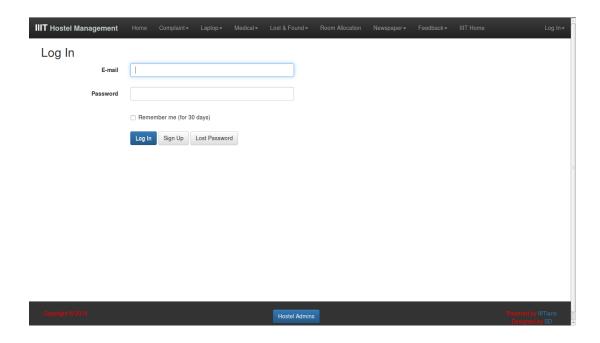
- 11. db.define_table('med', Field('roll', 'integer', writable=False, requires=[IS_INT_IN_RANGE(2007010] Field('description', 'text'), Field('doctor', requires=IS_IN_SET(MED)), Field('availablefrom', 'time', requires=IS_TIME(), default='6:00'), Field('emergency', 'boolean'), auth.signature)
- 12. db.define_table('Feedback', Field('roll', 'integer', writable=False, requires=
- $IS_INT_IN_RANGE (200701000, 201547000), IS_NOT_IN_DB (db,'Feedback.roll'), IS_NOT_EM (db,'Fe$
-), $Field('Room_Rating', requires=IS_IN_SET(rates)), \\Field('Washroom_Rating', requires=IS_IN_SET(Field('Extra_Comments', 'text'), \\ auth.signature)$

UI Screenshots

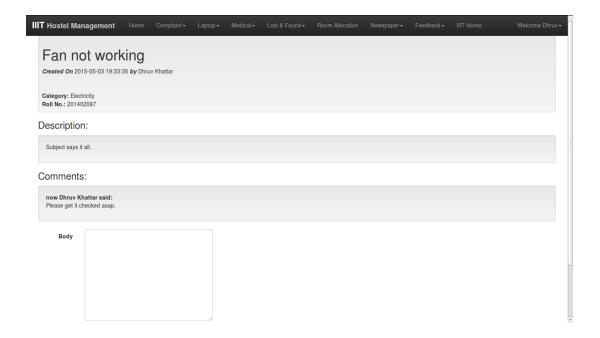
1.HomePage

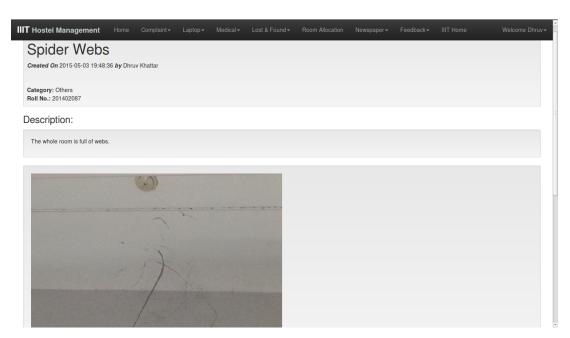


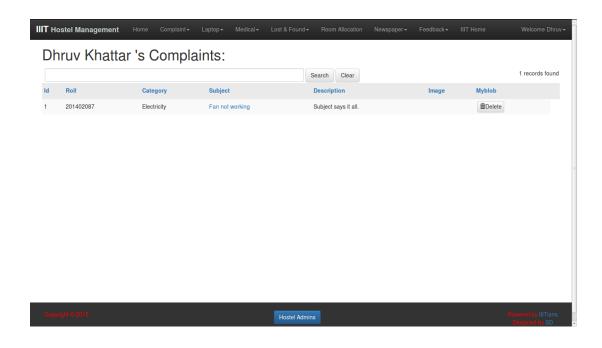
2.Login

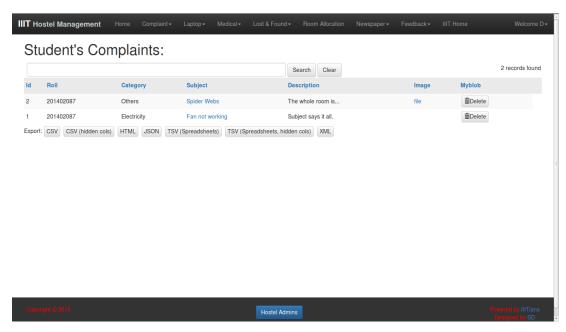


$3. \\ Complaints$

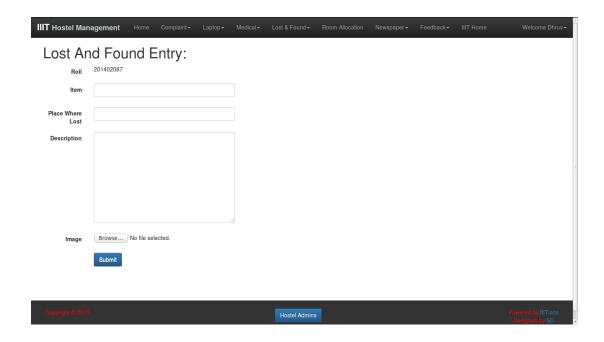


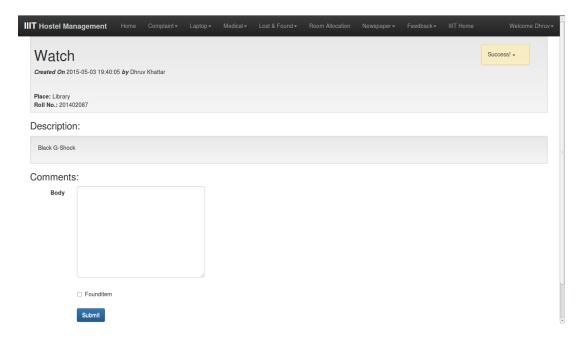




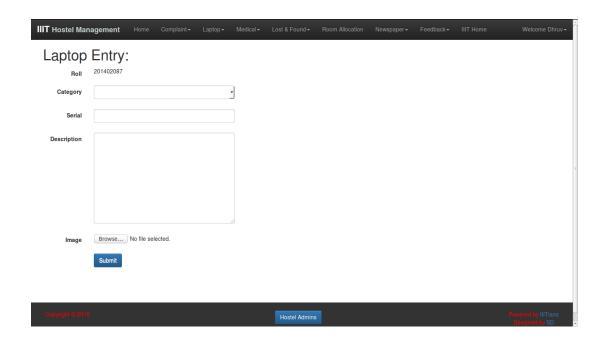


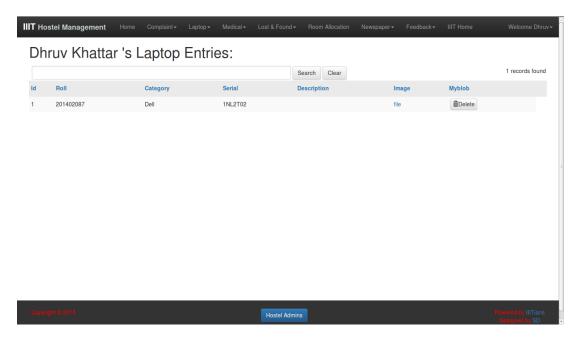
4.Lost&Found



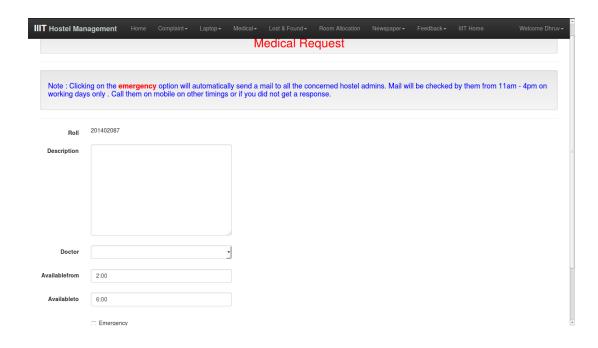


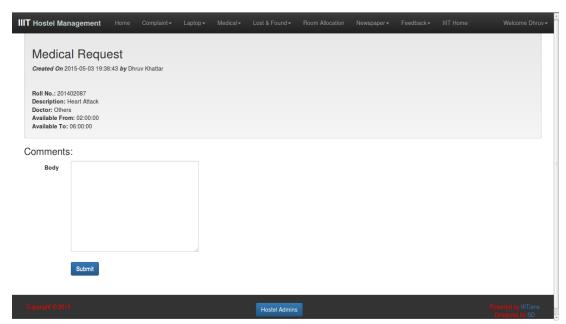
5.Laptop Entry and Views

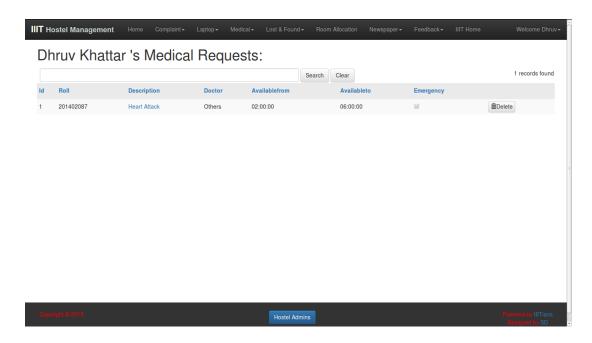




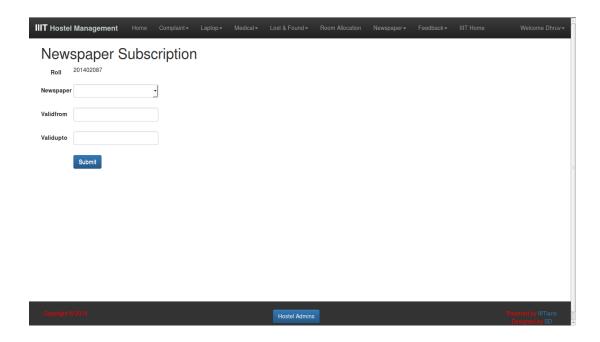
6.Medical Requests

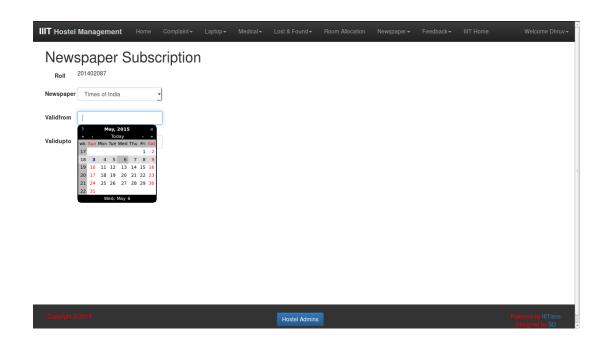


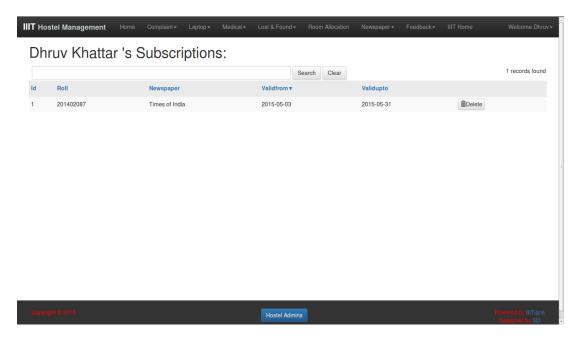




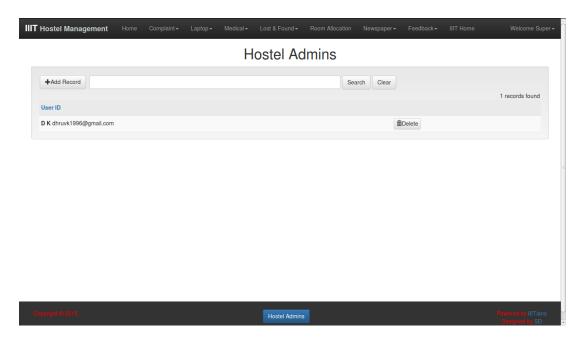
$7. New spaper \ Subscriptions$



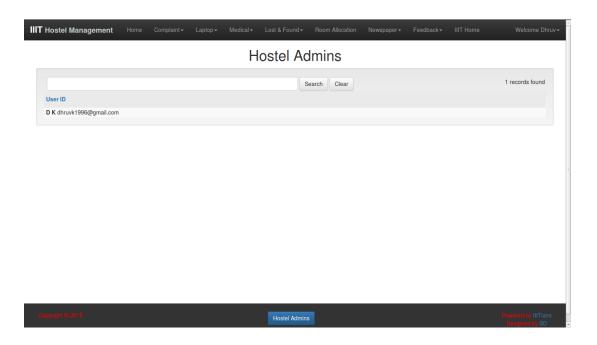




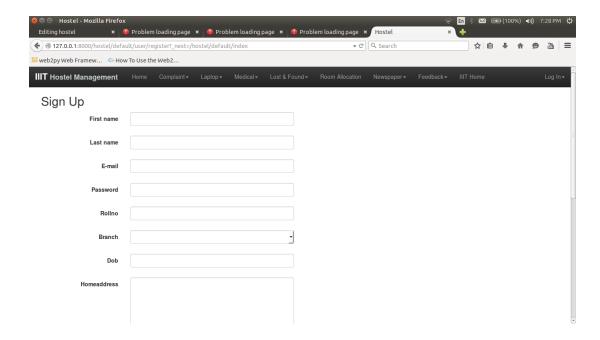
8.Super Admin



9.Admins



$10.\mathrm{Sign}~\mathrm{Up}$



END

This web application was developed by Dhruv Khattar (201402087) and Siddhartha Gairola (201402068). Thank you.

