

# Hestia

*ITWS-II: Web2py Application*

Developed By:  
Sriharsh Bhyravajjula and Isha Mangurkar

# Hestia

Sriharsh Bhyravajjula and Isha Mangurkar

May 4, 2015

## 1 About Hestia

Hestia is a Web2py application which acts as a private image-enabled blog entry platform as well as an event manager. It has been constructed using SQL databases, HTML, CSS, JavaScript and inbuilt Web2py functions, with the documentation being carried out using Latex. It has been hosted on [pythonanywhere.com](http://pythonanywhere.com)

It has a homepage which introduces the functions of the application, along with an elaboration on the idea behind the project. The menubar initially consists of a link to the homepage on the top-left, with a log-in menu on the top-right, which leads to a login/sign-up page. Once signed in, access is provided for four more menus, two for entering diary-entry/images and events, and two for showing the logged in user's previous entries in both fields respectively. The entry pages have forms with validated, typeset entry fields whose valid input is stored in databases corresponding to the users. The display pages have an option of exporting their tabular list of contents to various formats. Each individual entry can be viewed, edited and deleted as per the given requirements.

Along with a neatly designed website, and well used functions and databases, the highlight of this application is the privacy provided to individual users as per their login. The data provided by one user cannot be accessed by any other, and hence Hestia keeps the data private with respect to each user, thus providing a neat, pleasant and productive experience to its users.

## 2 Feedback

We hope you will enjoy using this release as much as we enjoyed creating it. If you have comments, suggestions or wish to report an issue you are experiencing - contact us at: [s.bhyravajjula@research.iiit.ac.in](mailto:s.bhyravajjula@research.iiit.ac.in).

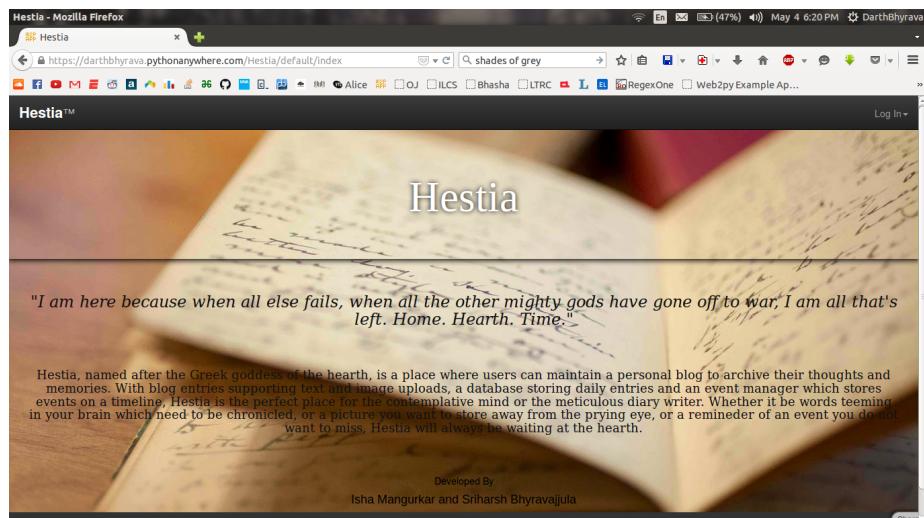


Figure 1: The Index Screen

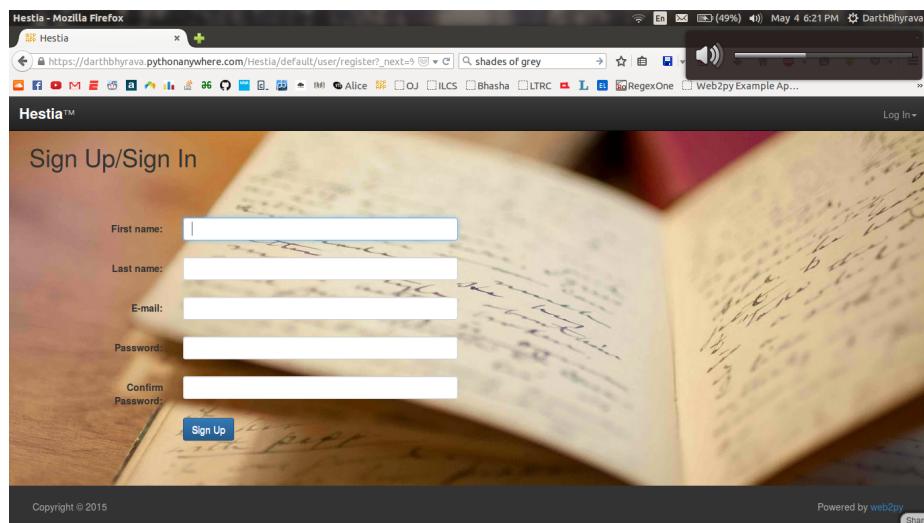


Figure 2: Sign Up Screen

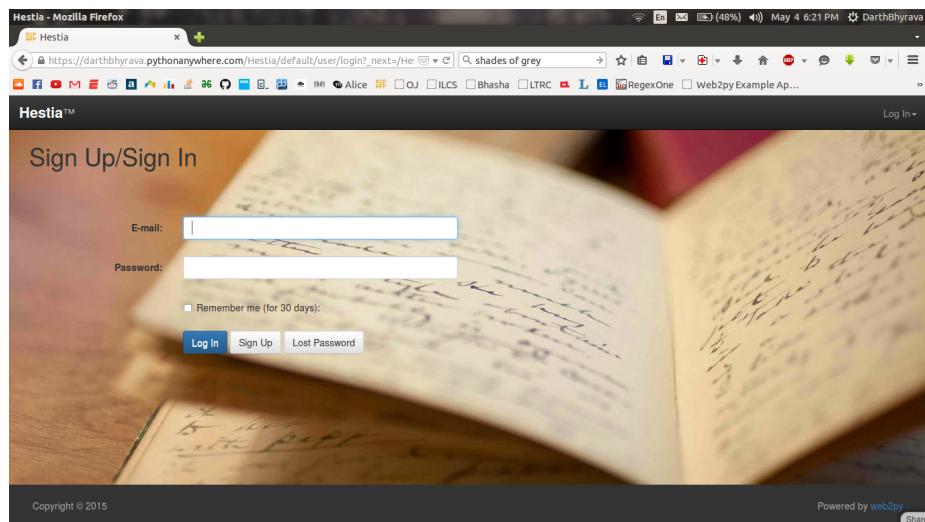


Figure 3: Login Screen

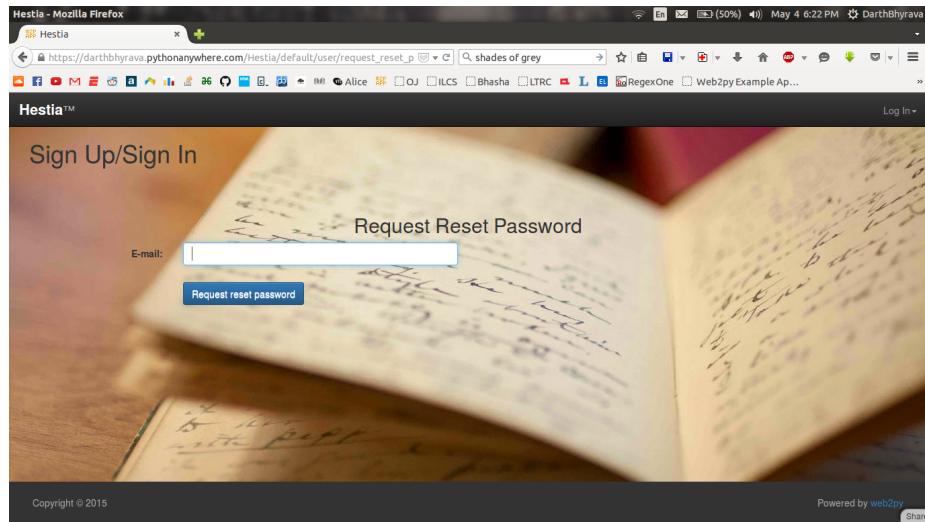


Figure 4: Forgot Password

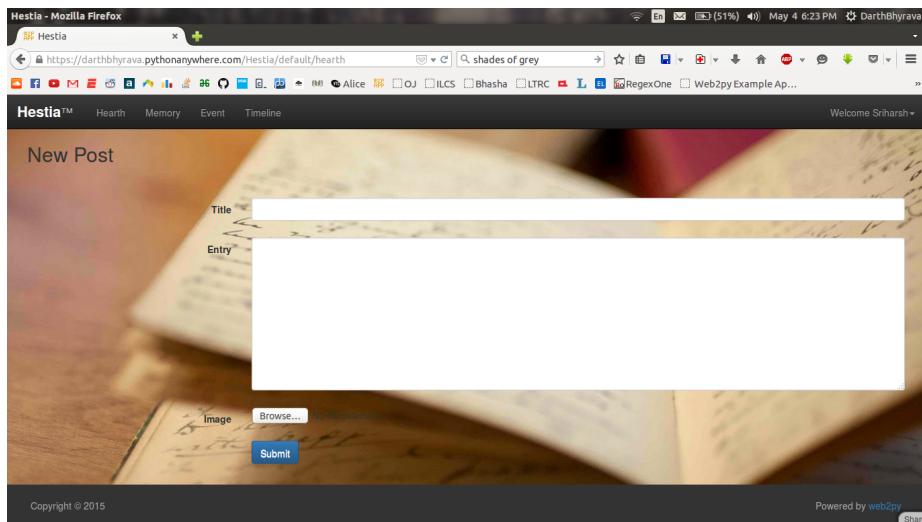


Figure 5: Text/Image Entry

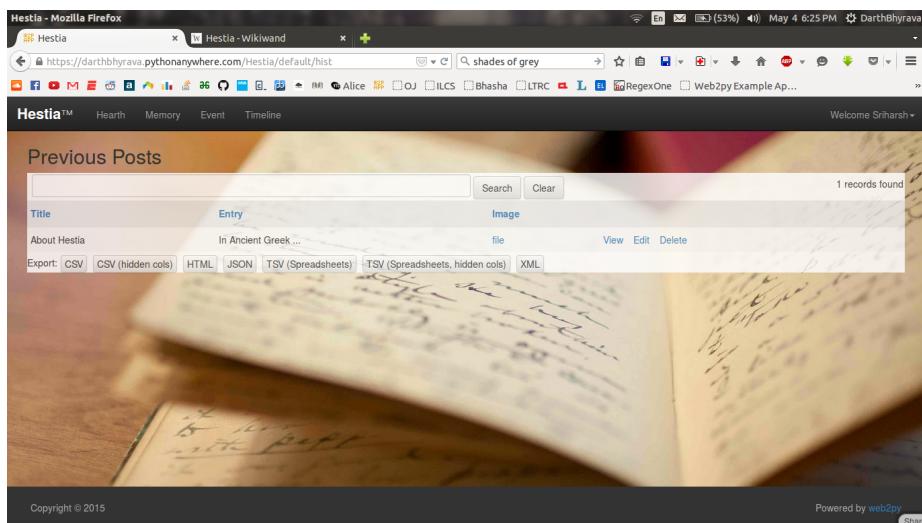


Figure 6: List of Entries

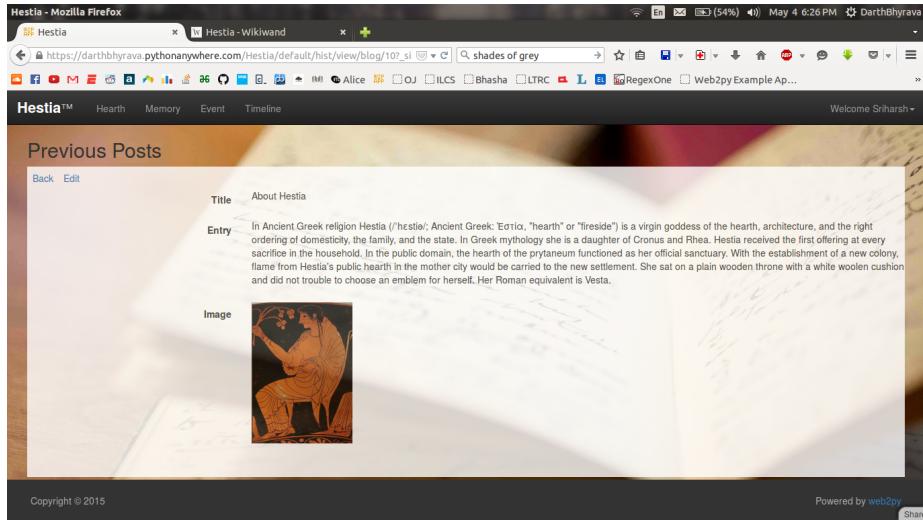


Figure 7: Entry Display

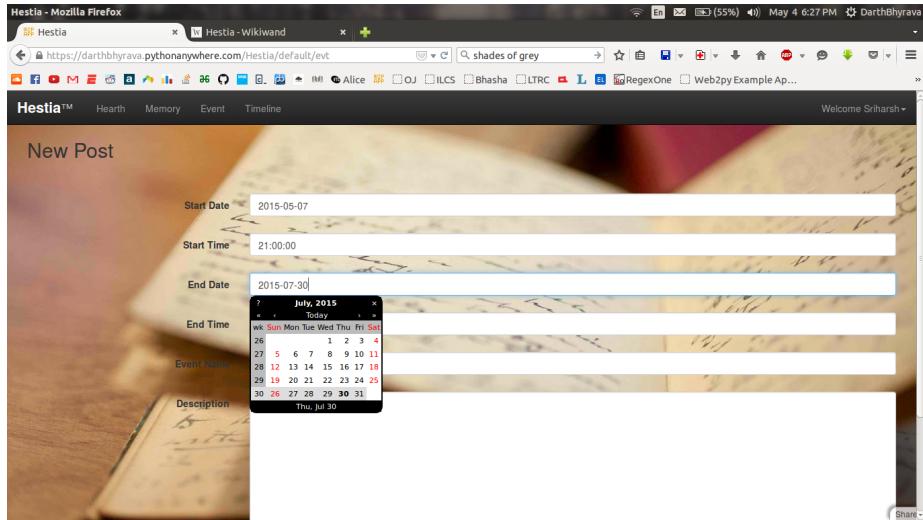


Figure 8: Event Creation

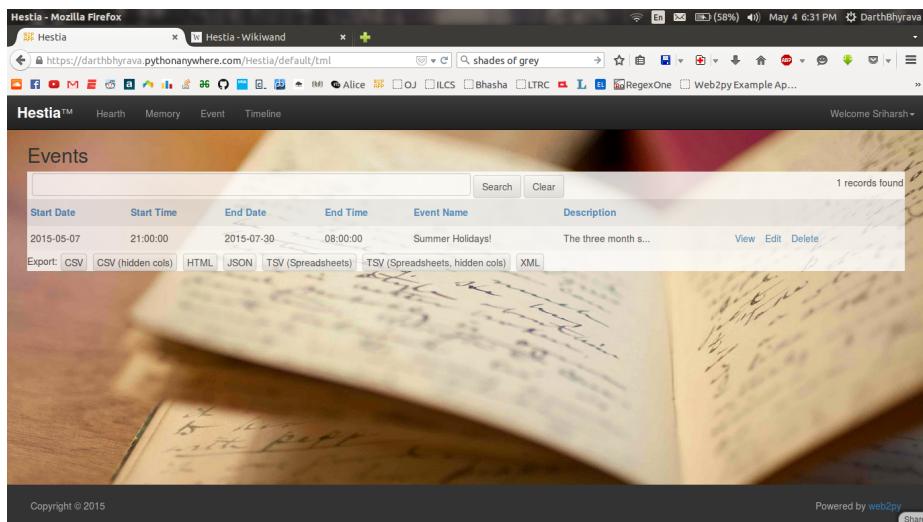


Figure 9: List of Events

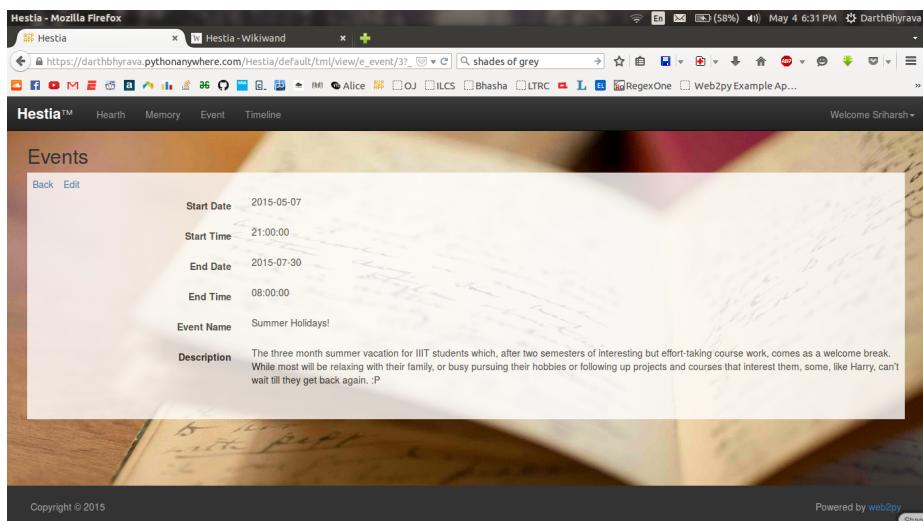


Figure 10: Event Display

# **Future Scope of the Application**

## **1 Our Suggestions**

- Facilitating sign-in/sign-up using third-party sites like Facebook, Google+, Twitter, etc. using Janrain.
- Sharing of images/events/entries via Twitter, Facebook, Google+, Instagram etc.
- Allowing email alerts for notifications for events
- Creating an in-built calendar application for event-handling
- Sharing of entries/events with other users of the app so that they can comment/share in turn
- An option to upload entries/events in form of tables as a whole

# Schema of Tables Used

## 1 The Blog Table

The 'Blog' table is defined thus in db.py:

```
db.define_table('blog',Field('f_title','string',label=T('Title'),requires=IS_NOT_EMPTY()),Field('f_desc','text',label=T('Description')),Field('f_image','upload',label=T('Image')),auth.signature)
```

The Blog table consists of three input fields - Title(of type 'string'), Entry(of type 'text'), Image(of type 'Upload') - along with auth.signature fields.

## 2 The Events Table

The 'Events' table is defined thus in db.py:

```
db.define_table('e_event',Field('e_start','date',label=T('Start Date'),requires=IS_NOT_EMPTY()),Field('e_end','date',label=T('End Date'),requires=IS_NOT_EMPTY()),Field('e_stime','time',label=T('Start Time')),Field('e_etime','time',label=T('End Time')),Field('e_name','string',label=T('Event Name'),requires=IS_NOT_EMPTY()),Field('e_desc','text',label=T('Description')),auth.signature)
```

The e\_event table consists of six fields - Start Date(of type 'date'), End Date(of type 'date'), Start Time( of type 'time'), End Time( of type 'time'), Event Name(of type 'string'), Description (of type 'text') - along with auth.signature fields.

# **Features**

## **1 Sign-Up/Log-In**

A feature which allows the user to sign up using name, email and a password. If user is already registered on the application, then he/she can sign in using email and password. In case the user has forgotten the password, there is an option for resetting it via email.

## **2 Entry/Image Form**

An entry form accessed via the 'Hearth' button on the menu-bar. It provides a title, an entry field and an image upload option. The title bar has a not\_empty validator to ensure non-garbage values.

## **3 Entries Display - List and Individual**

List of entries with images accessed via the 'Memory' button on the menu-bar. It has options to search for specific names, and against each entry name, there are options for viewing, editing and deleting the specific entry. Clicking on the View Button enables a display screen for each entry.

## **4 Event Form**

Another input form for the events, accessed via the 'Event' button the menu-bar with the input fields consisting of date and time pop-ups when selected for easier ease of inputs. Also contains name and description of event.

## **5 Events Display - List and Individual**

Selected via the 'Timeline' button on the menu-bar, this displays a list of all stored events in the application's database. The list has options to search for

specific names, and against each entry name, there are options for viewing, editing and deleting the specific entry. Clicking on the View Button enables a display screen for each entry.

## **6 Privacy**

Each user can access only his or her events and entries via this application when they are logged in, hereby maintaining the privacy of the individual's data.

## **7 Upload Image; Search, View, Edit, Delete Entries**

As mentioned earlier, the application allows users to upload their pictures online onto the app, and also allows the options of searching, viweing, editing, and deleting the app's entries and events.