

Cloud Azure App

Aditi (Partap, Singla, Gupta)

November 2017

Our website is deployed on <http://instabooks.azurewebsites.net/>. The key design principles implementation details are as follows.

1 Databases

Two collections in Cosmos DB :

- users : Stores all profile info and post ids of each user. Also, people followed by you and those who follow you. The structure of the datatype user is as follows :
 - user_name
 - password (encrypted)
 - following : List of user_name, name, pic_link
 - followers : List of user_name, name, pic_link
 - posts : List of date, text, file sorted chronologically
 - profile : dob, gender, name, pic_link (URL to the blob of his profile pic)
 - chats : Dictionary of chats, with key as the other person. Each chat is an array of messages date, text, sender_uname, receiver_uname, sender_name, receiver_name.

2 Blob Storage

A storage account was created on azure. We made a new container to store the photos and files shared by people. New images are added as blobs, and images are converted to base64 text and saved. CORS origins were set up to allow local host and the azure website to access the blobs.

3 APIs

The following APIs were implemented in NodeJS to extend the functionality of the app. Json Web Tokens are used to maintain the user's session.

1. login_user : Log in the user. A session token is sent back as part of response.

2. register : Register into our site. All the info such as DOB, name, username (unique email id), gender and a profile photo is required. Also, the user is logged in & the server sends back a token.
3. get_user_info : Get all info of a particular person whom you are following.
4. add_post : User can add a post to his page, which would be visible to all the people who are following him.
5. upload_file : User can upload an image or video or pdf. (All formats are supported, since we convert to base64) These are also visible to all his followers.
6. send_message : When a user sends a message to another user, this message is added to the chats of both the users.
7. get_user_list : List of name and profile pic of all users whom the person is not following. This API is needed so that the user can see the list of people whom he can follow.
8. follow_user : Updates the database collection of users based on who wants to follow who.

4 App Work-flow

User can register first. Then, he can log in with his email id and password.
The user will see his own page, and there are buttons to view the list of people he is following (Following), and also the list of people who follow him (Followers).
Also, the user can click on 'Follow' button to see the list of all the other users whom he can follow. A user can see someone's page only if either one of them is following the other.
The user can upload new posts on his page, and he can also upload a photo or video or any text file on his page.
The user can also chat with any other user, and the chats are private and hence, not visible to anyone else.

5 Screenshots

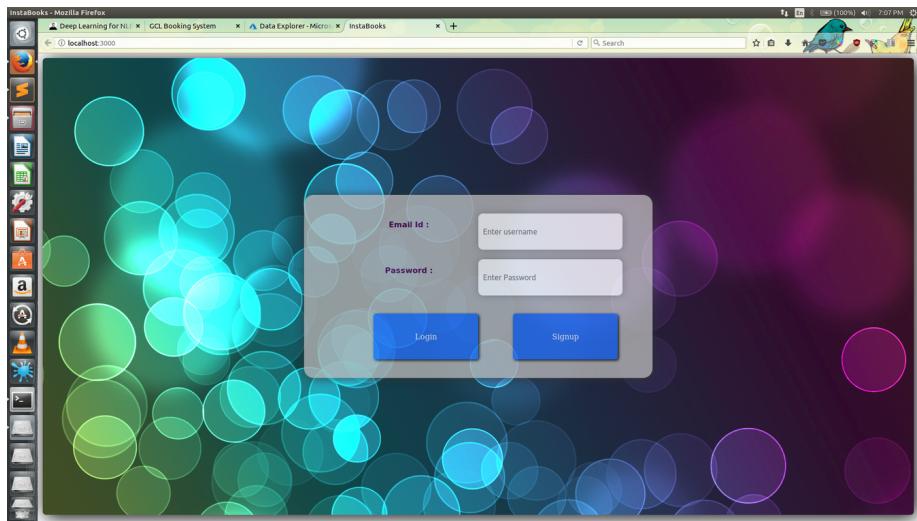


Figure 1: Login Page

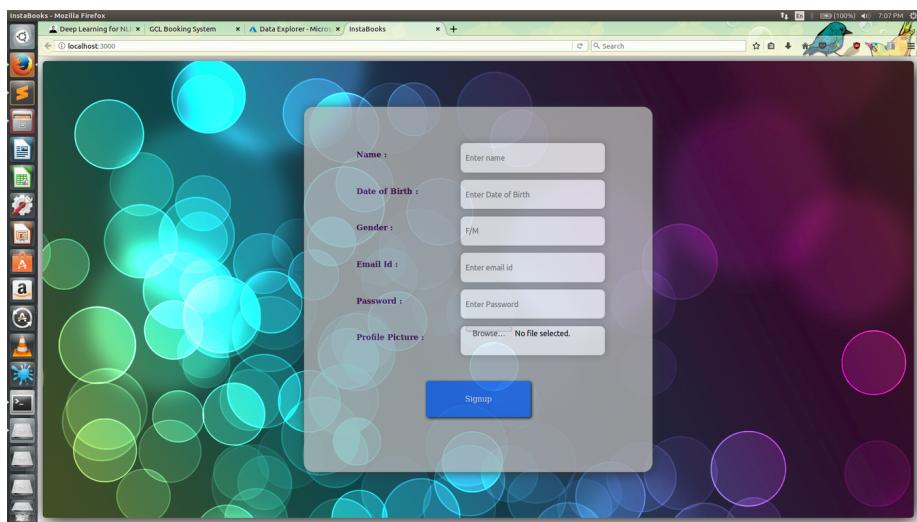


Figure 2: Signup Page

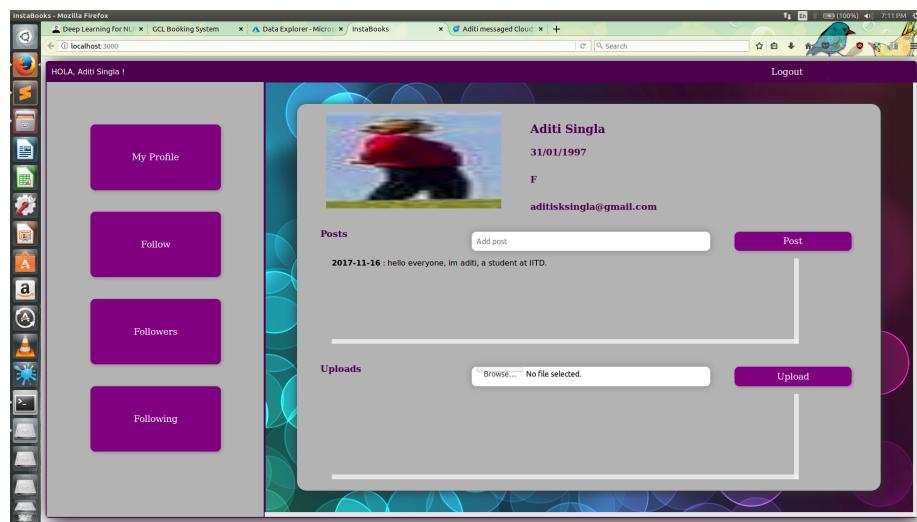


Figure 3: Homepage of a User

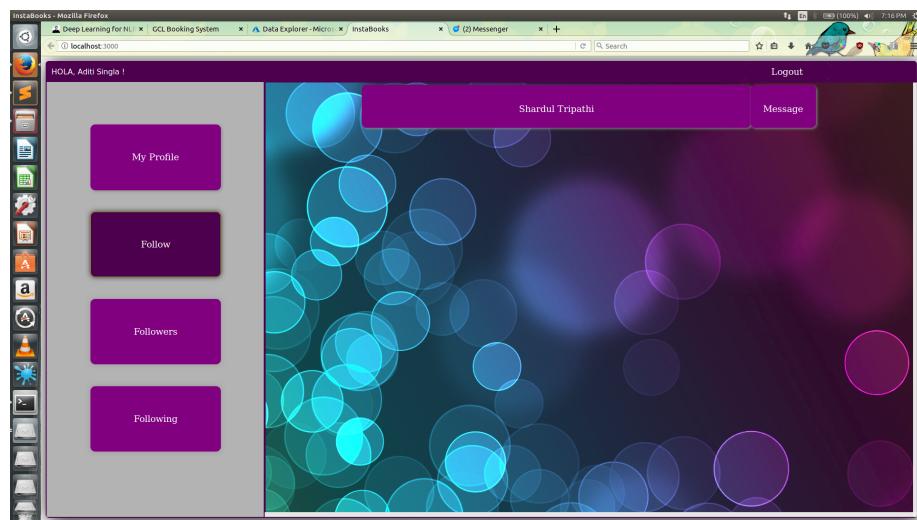


Figure 4: Follow Page

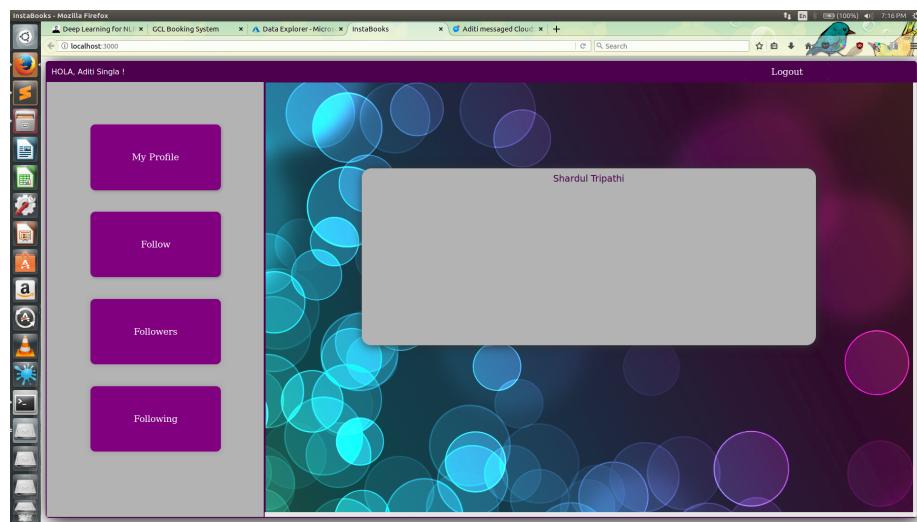


Figure 5: Followers Page

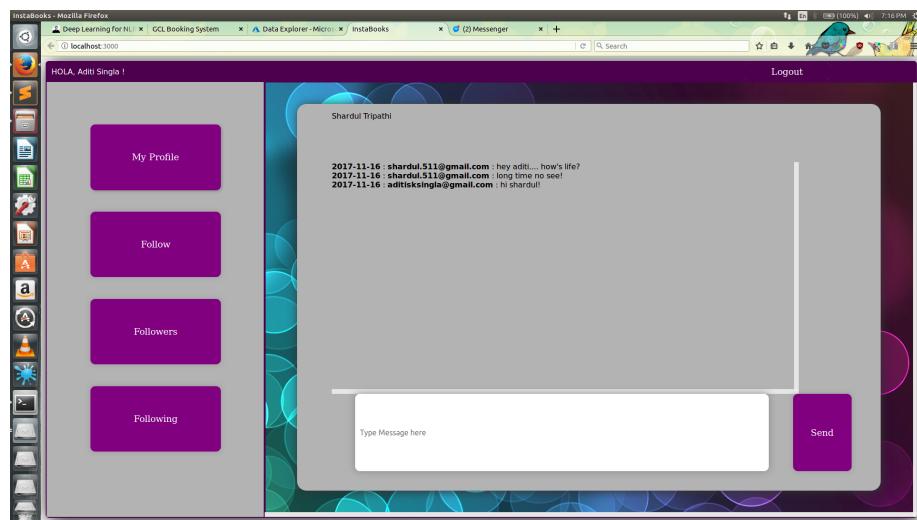


Figure 6: Messaging Page