

INITIAL SET UP FOR UNIBUILDER PROJECT

- **Setup**

- **Technical Terms**

1. Uni admin – Full control on Uni builder portal
2. Builder admin – Full control on the Builder Portal
3. Owner – Client of a particular Builder admin
4. Sub-Contractor – Sub contractor of a particular Builder admin
5. Builder Users – Users created by the particular Builder admin
6. Account Type – (Builder admin / Internal User, Owner and Sub-Contractor)

- **Configure ASSETS / LOG URLs**

1. Assets (images / js / css / logs) urls need to keep as configurable- Either CDN / From Running Server – **Done (Log added)**

- **Database**

1. Multiple DB connection need for the following operations - **Done (Scenario: if developer is inserting data in read DB how will we handle this)**

- Write
- Read
- Session

- **Session**

1. Session to be validated at the time of ajax request
2. Session Management in database. Based on user type the session values to be stored. tables fields below - **Done - CI session table structure**

- session_id
- ip_address
- user_agent
- last_activity
- user_data

- **Validations**

1. Server side validation library to be created
 - Need place holder for server side / common validation
2. JQuery side common validation setup to be done – JQuery rules to be used
 - Submit validation will be done
 - UI team to provide the mandatory validation
 - No JS alert for validation
 - Field names to be same as the respective database field names

- **Errors**

1. Errors All errors to be enabled in development and testing - **Done**
2. Errors to be disabled in production - **Done**
3. 4XX, 5XX error page for user
4. Only for the critical error the application has stop
5. Need to use exception handling in each code (Try , catch and throw) – (Currently we have done this as common)
6. **Page not found / OOPs page need to designed – It will be same for all the pages**

- **Logs**

1. Need to log only the **warning** and **critical error** in **Error Log** file
2. Insert / update / delete query need to track in **Query log** file
3. Logs are tracked in the following files
 - Error log file
 - Query log file
 - Tracking will be done on hierarchical directory format, File path and enabling or disabling of error log is configurable
 - YYYY
 - Month
 - 01_error_log.txt

- 01_query_log.txt

- **Transaction Management**

1. For Bulk / Multiple operation should use either transaction in php model or transaction in stored procedure

- **Profiler**

1. Profiler to be enabled to view the performance of the application, If a profiler is enabled it will be enabled for the whole application not for a specific page

- **Encrypted URL**

1. All page URL's should be encrypted. Example: - http://www.unibuilder.com/Bxbqc97r8q4XE99D7b20

- **SMS**

1. For SMS, need to implement common library / model
2. Based on the user's country, need to selected the SMS service provider and send the SMS
3. There will be 2 countries SMS service provider API going to implement currently – Provider details will confirm soon.

- **Mails**

1. All mails should go as per respective SMTP settings.
2. SMTP mail settings classified as
 - a. DEFAULT
 - b. SUPPORT
 - c. DONOTREPLAY
 - d. Etc.
3. How track mails
 - a. If there is any smtp connection issue we need to insert mail details to mail table and need to show to the user saying that, your message cannot be sent due to following reason, it will be there in the queue, will send it soon.
 - b. Failed emails only tracking in database table
4. How fetch mail works?
 - a. If a mail is coming to mail server, it will trigger / push / hit the mail to our server specific url (This location we have to give and it will get configure in the mail server).
 - b. In that url we will be getting the email as text / eml file
 - c. Our code will read that content and thread to specific message using the uniquely generated email id

- d. Mail server hits to our server url but it might pass extra params / details to for each hits
 - i. Eg: Configure url will be www.unibuilder.com/fetchmail/
 - ii. www.unibuilder.com/fetchmail/252asdfasdf258asdfa2sdf5
 - iii. www.unibuilder.com/fetchmail/asdfasdgdf324523sghy436

- **Mail Template**

- There are two type of mail templates are require
 1. Master Mail Template : It holds all default templates of uni builder portal
 2. Builder Mail Template : When Builder admin register, all default master templates would be copied to builder admin
 3. Builder admin / builder user can edit all mail templates which are there in the builder portal.
 4. If the builder admin / builder user want to revert it to default (old one), at that time we need to copy it from master mail template and update it.

- **ON / OFF**

1. **Error**

2. **Logs**

3. **Profiler**

- **Memcache**

1. Not implementing now, Will be using session to store the values, which need to get across the portal once a user is logged in.
2. Following are the details will be storing initially in session
 - User information's
 - Access levels
 - Menus
 - Mail preferences

- **Template**

1. Common template –Load common file (bootstrap CSS / JS, Validator files etc.)
2. Headers / Footer / Left side bar / will be separate for the Account types (Builder Admin / Builder user, Owner & Sub contractor)
3. Relevant JS / CSS will be loaded in that page view files

- **Svn Setup**

1. 10.0.0.212 - Unibuilder design dev
2. 10.0.0.215 - Unibuilder development dev

- **Delete**

1. Default builder user do a delete action, it will get removed from the database table, for this we will be using ON CASCADE DELETE concept in mysql innodb
2. There will be some records we should not remove from the database when a builder user does a delete action. That details will let you know soon.

- **Triggers**

1. Triggers to be implemented, depends on scenario, mainly for tracking the activities (history)

- **Date Format**

1. **Datetime** datatype need to use **gmdate** function
2. **Date** datatype need to use **date** function

- **Code Checker**

1. Explanation will be provided soon.

- **PHP Docs**

1. Need to document the php classes, functions.
2. Execution process will explain soon.

- **Data tables**

1. Data table implementation fully with jQuery data table / Code igniter and jQuery data table need to finalize
2. Pipelining data to reduce Ajax calls for paging : Will try to implement this concept, if not happening we can think of server request for each pagination click

- **Schedulers**

1. There are two types of schedulers
 1. Default system schedulers, example below
 - a. Only one schedule is available now- notify before the Builder admin subscription getting expire
 2. Builder Portal Notification schedules. This has to be implemented with the help of thread. Example of notification as follows
 - a. Schedules notification

- b. Bid package notifications
- c. Etc.