

**Jashore University of Science and Technology**

**Department of Computer Science and Engineering**

**Course Code:** CSE-4108

**Course Title:** Web Development Project Laboratory

**A project report on**

Just undergraduate final admission system

|  |  |
| --- | --- |
| Submitted to | Submitted by |
| [Mostafijur Rahman Akhond](https://just.edu.bd/t/mra" \t "_blank)  Assistant Professor,  Department of Computer Science and Engineering  Jashore University of Science and Technology | **Chayan Das**  Roll: 180147 4th Year 1st Semester Session: 2018-2019 Dept. of Computer Science and Engineering Jashore University of Science and Technology. |

**Submission Date:**  09.07.2023

**Index**

**Chapter 1: Introduction --------------------------------------------------------------------------------- 3**

* 1. **Background-------------------------------------------------------------------------------------------- 3**
  2. **Motivation---------------------------------------------------------------------------------------------- 3**
  3. **Objectives---------------------------------------------------------------------------------------------- 4**

**Chapter 2: Software requirement specification------------------------------------------------------ 5**

* 1. **Introduction------------------------------------------------------------------------------------------- 5**
  2. **Overall description----------------------------------------------------------------------------------- 6**
  3. **External Interface Requirements------------------------------------------------------------------ 7**
  4. **System features--------------------------------------------------------------------------------------- 8**

**Chapter 3: System Design------------------------------------------------------------------------------- 11**

* 1. **System design----------------------------------------------------------------------------------------- 11**
  2. **Use case design--------------------------------------------------------------------------------------- 13**
  3. **Activity diagram------------------------------------------------------------------------------------- 14**
  4. **Class diagram----------------------------------------------------------------------------------------- 16**
  5. **ER diagram-------------------------------------------------------------------------------------------- 20**
  6. **Schema diagram------------------------------------------------------------------------------------- 21**

**Chapter 4: Implementation details-------------------------------------------------------------------- 23**

* 1. **Authentication---------------------------------------------------------------------------------------- 23**
  2. **Dashboard--------------------------------------------------------------------------------------------- 25**
  3. **Application page------------------------------------------------------------------------------------- 26**

**Chapter 5: Testing----------------------------------------------------------------------------------------- 36**

* 1. **White Box Testing----------------------------------------------------------------------------------- 36**
  2. **Black Box Testing------------------------------------------------------------------------------------ 37**

**Chapter 6: Deployment---------------------------------------------------------------------------------- 38**

**Conclusion-------------------------------------------------------------------------------------------------- 39**

**Abstract**

In JUST, there is an undergraduate admissions committee. They select the applicants through an admission test and publish a merit list. Then the applicants are given the subject order according to their choice. Based on these data an applicant is given a subject and this admission system is very difficult, time-consuming, and expensive to do manually. An automated system has been developed on how to get rid of this problem. So that the waiting list for admission can be published very soon and admission can be done by verifying the information. For this, an automated system is needed. Any system can quickly publish applicants' waiting lists for admission based on merit position, subject order, and vacant department and can see admission documents. This automated system can easily solve manual problems, which will eliminate a lot of time, money, and additional complications for the university. This application work fully automatically.

**Chapter 1**

**Introduction**

* 1. **Background**

Admission of a university is an indispensable activity in every educational activity and is as old as education itself. With more students applying for admission than in previous years, admission process for admissions officers has become increasingly time-consuming. More applications resulted in heavier paperwork and processing challenges. Every admission is routed through various faculty for evaluation, and the manual admission process caused difficulty in admission, and archiving. In order to speed up and simplify the admission process, the development of the Automated Applicants Admission System is to enact document redesign and an automated management program. The proposed ‘’undergraduate final admission system’’ would store, route, and retrieve prospective and current application documents. The development of science and technology has immensely contributed to the growth of computers and the internet which has adversely increased the use and need for an automated applicant’s admission system. Universities, Colleges, Companies, organizations, etc. are seeking better ways to distribute and disseminate information throughout the world. In order to achieve this, they develop and rely on online applications, which help them input, update, process, and disseminate information. The internet has increased the use of online applications because data could be easily and readily accessed as well as documented. This project is targeted at allowing admission of applicants' decisions to be made faster and more efficiently. It looks at improving the present traditional system of the manual process (paper-based) which is manually documented. This type of project is a hot topic in the whole world to digitalize our education System.

* 1. **Motivation**

There are some insufficient small systems in our university to manage the whole admission system. But those are not well managed and haven’t a central database. Jashore University of Science and Technology has at least five different offices and steps to complete the admission system. The offices have their own system to manage the applicant’s data. When they need to transfer data with each other they use pen drives, hard drives and mails. But now days, it is not sufficient. In this system, the communication of data is a big challenge. Currently, JUST has a web application “JUST Admission 2022" that consists of some of the features. But they are not packed into a complete application. Our new system will add all possible features including online application form fill up, keep records, and access a central admission database which is more secure than the existing system.

* 1. **Objectives**

The main objective of this system is to give these facilities.

1. **Efficiency:** Digitalize the admission system will save time and resources than the manual system. All receipts will generate automatically, so to time can be saved.
2. **Accuracy:** All data will save accurately in a central database. It will ensure the maximum security of all sensitive information. It will eliminate the risk of lost important papers and records.
3. **Consistency:** Automated admission ensures that all data and applications are transferred consistently and accurately, reducing the risk of errors or data loss.
4. **Security:** After completing the project, it will connect with the main server of our university. So all data will be safe and secure.
5. **Performance:** Performance of the whole entire process will increase undoubtedly.

**Chapter 2**

**Software requirements specification**

* 1. **Introduction**
     1. **Purpose**

The main purpose of this system is to make the admission system easier and smooth. Admission means the management of huge amount of data. This type of system can be the better solution than the old file system. On the other hand, the accessibility of admission data at any time and desired format is also make a great opportunity for the authority. We said that it is not the implementation of full requirement, it’s only a partial fulfillment of the full project.

This project includes these implementation.

1. All users authentication.
2. Online application process.
3. User interface front- end design for users etc.
   * 1. **Document conventions**

There some shortcut will use on this document like

1. **JUST-** Jashore University of Science and Technology.
2. **SRS** – Software Requirement Specification
3. Bold lines are used for important lines.
   * 1. **Intended Audience and Reading Suggestions**

We have at least five type of users

1. Applicant
2. Dean office
3. Hall
4. Medical and
5. Register

All of them will review this document. And also our project stakeholders, future implementation team, project supervisor will study this report.

* 1. **Overall description**
     1. **Product perspective**

From a product perspective, an automated admission system should have a number of features and capabilities to make it effective. These may include:

* The ability to automatically admit applicants to a new system based on predefined criteria or rules.
* The ability to automatically retrieve data from database.
* The ability to update data at any time.
* The ability to handle a large volume of applicants concurrently, to ensure that the process is efficient and effective.
* The ability to be easily configured and customized to meet the specific needs and requirements of the university using the system.
  + 1. **Product function**

The major functions that should be allowed by the JUST Admission System are listed below,

* Admin to manage user information.
* Applicant can update information
* Applicant will receive notification
* Applicant may view a tutorial where application fill-up details can be added.
  + 1. **User classes and characteristics**

**Applicant:**

* Update data
* Insert data
* View notifications
* Download Application Form
  + 1. **Operating environment**

As if it is a web based application so it can be used in any operating system which support internet browsing.

* + 1. **Design and Implementation Constraints**
* All the information of applicants and users must be stored in the database. And all the information is accessible by the system.
* React.js, Bootstrap is used for Frontend Design.
* Node.js, Express.js is used for the Backend server.
* MSSQL server is used as a database
  1. **External Interface Requirements**

There are a number of external interface requirements that an automated admission system may need to consider in order to be effective. These may include:

* + 1. **Integration with the existing system:**

The automated admission system will need to be able to access and retrieve data and other assets from the existing system or platform, and may also need to be able to write data back to the existing system in certain cases.

* + 1. **Integration with the new system:**

The automated admission system will also need to be able to access and write data to the new system or platform, in order to facilitate the application process. JUST Undergraduate Final Admission System.

* + 1. **User authentication and authorization**

The automated admission system may need to be able to authenticate users and ensure that they have the appropriate permissions to access and use the system.

* + 1. **User interface**

The automated admission system will typically include a user interface that allows administrators to configure and manage the system. This interface must be easy to use and intuitive, with clear instructions and guidance for administrators.

* 1. **System features**
     1. **Applicant management**

**Description and priority:**

This allows fixed users to get access to the system as applicants. The users can access the system with their admin-permitted given userid which is none other than GST ApplicantId. The admin controls the Information.

**Stimulus/Response Sequence:**

* Users enter their given admin userid and password in the sign in field.
* The system stores detailed information about the users and provides access to sign-in in the system.
  + 1. **Functional Requirements**
       1. **Authentication**

**Input:** The user will enter the admin-given email and password for sign-in.

**Output:** access to sign in.

* + - 1. **Application Fill-up**
* **Personal Information:**

**Input:** Put the user all personal information in this form.

**Output:** Successfully inserted or updated personal information.

* **Address:**

**Input:** Put the user address information with permanent and present address in this form.

**Output:** Successfully inserted Address.

* **Education:**

**Input:** Put the user educational information in this form.

**Output:** Successfully inserted/updated Educational information (SSC and HSC).

* **Others:**

**Input:** Put the user all information in this form.

**Output:** Successfully inserted/updated Additional information if given.

* **Guardian:**

**Input:** Put the user guardian information in this form.If father is absent as guardian then legal guardian option should fill-up.

**Output:** Successfully inserted guardian information.

* + - 1. **Application Form download**

**Input:** Fetch all data from database

**Output:** Application form successfully generated as pdf form.

* + 1. **Non-Functional Requirements**

**Performance:**

This includes requirements related to the speed, scalability, and responsiveness of the system. For example, a requirement that the system should be able to handle a certain number of concurrent users or process a certain amount of data within a specific time frame.

**Security:**

This includes requirements related to the protection of the system and its data from unauthorized access, as well as the ability to detect and recover from security breaches.

**Usability:**

This includes requirements related to the ease of use and understandability of the system for the end-users.

**Reliability:**

This includes requirements related to the system’s ability to function correctly and consistently under normal and abnormal conditions.

**Maintainability:**

This includes requirements related to the ease of maintaining the system, including testing, debugging, and modifying the system.

**Portability:**

This includes requirements related to the ability of the system to be easily transferred to different hardware or software environments.

**Compliance:**

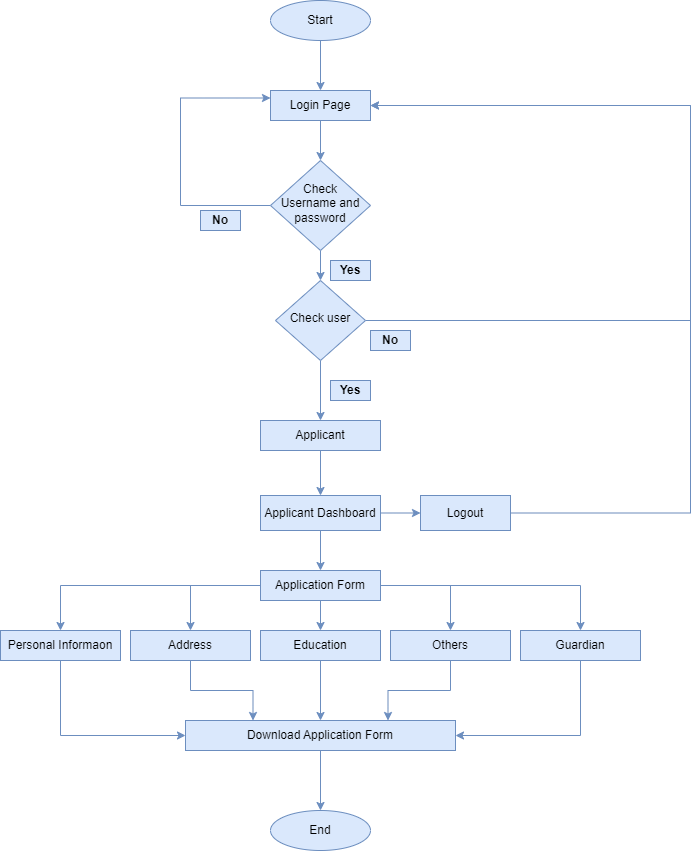
This includes requirements related to adherence to laws, regulations, industry standards, or company policies.

**Chapter 3**

**System Design**

* 1. **System design**

This system is a web based system. This is an overall design of the project.



**Fig 3.1:** System flowchart.

Figure- 3.1 shows the flowchart diagram of the project. Here,

1. At first applicants can see the Login page.
2. They will insert their userid and the password for login.
3. The system will check the userid and password with the database. If it matches then applicant will reach to Applicant Dashboard.
4. Now applicants will go to the applicant component and fill their application form.
5. Application form has five sections like Personal Information, Address, Education, Others and Guardian. Applicant should fill all the required field of every section.
6. After that they can download the application form.
   1. **Use case design**

This is the use case diagram of the applicant.

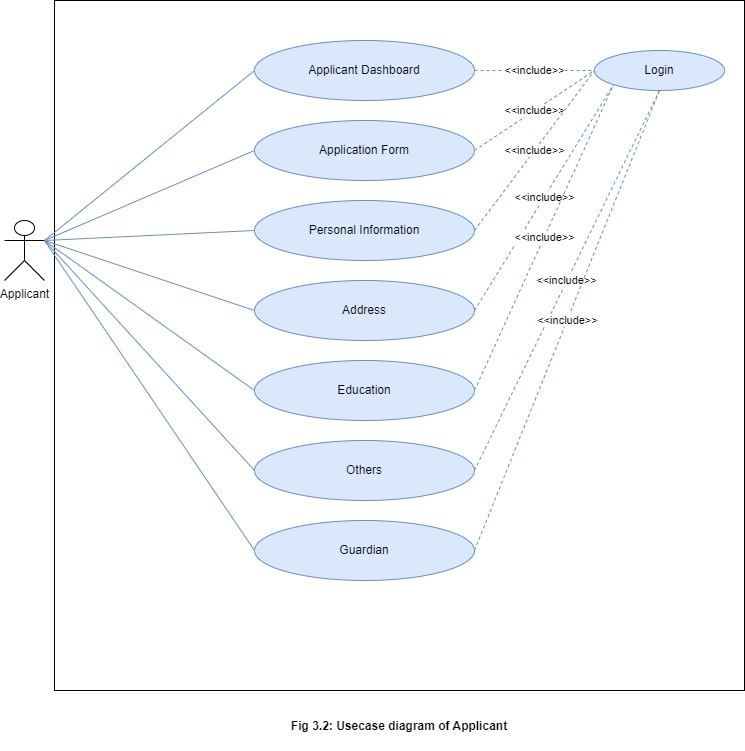


Fig- 3.2 shows the use case diagram of this project. In this diagram we have one actor named Applicant. The activities of the actor are

1. At first applicants will Login page. It is a initiating activity for the system. They will insert their userid and the password for login.
2. The system will check the userid and password with the database. If it matches then applicant will reach to Applicant Dashboard.
3. Now applicants will go to the applicant component and fill their application form.
4. Application form has five sections like Personal Information, Address, Education, Others and Guardian. Applicant should fill all the required field of every section. All of this sections activity is dependent on login. Without login anyone can’t access them.
   1. **Activity diagram:**

Here is the activity diagram of this system.

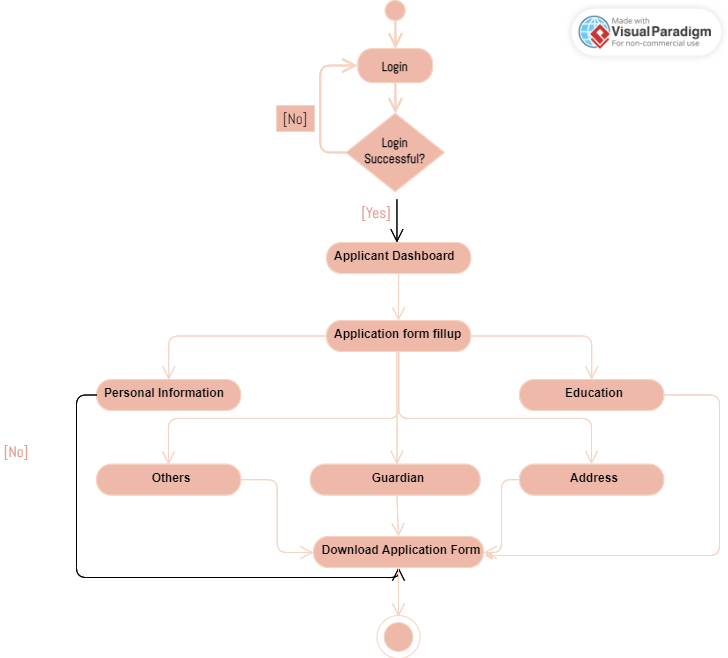
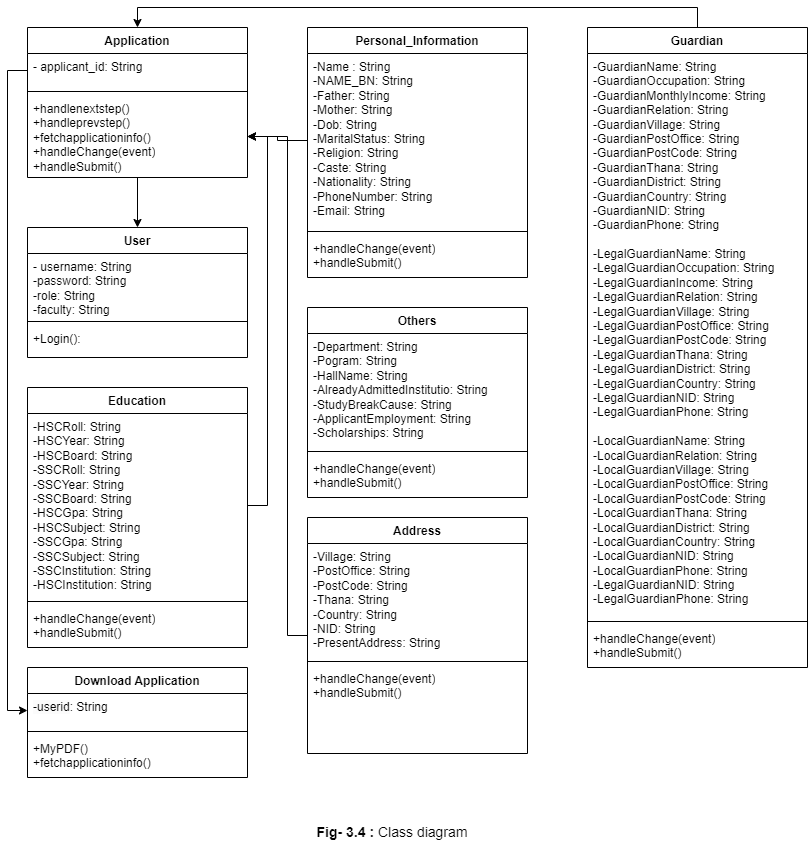


Fig- 3.3: Applicant’s activity diagram.

Fig – 3.3 represent an activity diagram. In this diagram we will see the activity of Applicant. Here,

1. Activity will start from login page
2. Click login button and logged in.
3. Click Application Form button and go to Application.
4. Click submit button and go to next step.
5. Click Download Application form and download Application form pdf.
6. Click logout button and logout.
   1. **Class diagram:**

Class diagram of the system give below.



According this class diagram we can notice 8 classes in this system.

**User:**

This class handles Users. The User class includes public login() method to verify the login of the user. When the user clicks the Login in button login() method is called and the user gets access to the system.

**Application:**

This class has an applicant\_id string value and handlenextstep () method is used to go to next step like education to others. ‘handleprev ()’ method is used to go to previous steps. Here fetchapplicantinfo () method is used to fetch applicant data from database.

**Personal Information:**

This class has some string values they are

* Name: '',
* NAME\_BN: '',
* Father: '',
* Mother: '',
* Dob: '',
* Gender: '',
* MaritalStatus: '',
* Religion: '',
* Caste:'',
* Nationality: '',
* PhoneNumber: '',
* Email: '',

These values are used to insert or update data into database. And handlesubmit() method is used for submit data to database. Another method handlhange(event) is used to handle data change in those strings.

**Address:**

This class has some string values they are

* Village: '',
* PostOffice: '',
* PostCode: '',
* Thana: '',
* District: '',
* Country: '',
* NID: '',
* PresentAddress: '',

These values are used to insert or update data into database. And handlesubmit() method is used for submit data to database. Another method handlhange(event) is used to handle data change in those strings.

**Education:**

This class has some string values they are

* HSCRoll: '',
* HSCYear: '',
* HSCBoard: '',
* SSCRoll: '',
* SSCYear: '',
* SSCBoard: '',
* HSCGpa: '',
* HSCSubject: '',
* SSCGpa: '',
* SSCSubject: '',
* SSCInstitution: '',
* HSCInstitution: '',

These values are used to insert or update data into database. And handlesubmit() method is used for submit data to database. Another method handlhange(event) is used to handle data change in those strings.

**Others:**

This class has some string values they are

* Department: '',
* Pogram: '',
* HallName: '',
* StudyBreakCause: '',
* AlreadyAdmittedInstitutio: '',
* ApplicantEmployment: '',
* Scholarships: '',

These values are used to insert or update data into database. And handlesubmit() method is used for submit data to database. Another method handlhange(event) is used to handle data change in those strings.

**Guardian:**

This class has some string values they are

|  |  |
| --- | --- |
| * GuardianName: '', * GuardianOccupation: '', * GuardianMonthlyIncome: '', * GuardianRelation: '', * GuardianVillage: '', * GuardianPostOffice: '', * GuardianPostCode: '', * GuardianThana: '', * GuardianDistrict: '', * GuardianCountry: '', * GuardianNID: '', * GuardianPhone: '', * LegalGuardianName: '', * LegalGuardianOccupation: '', * LegalGuardianIncome: '', * LegalGuardianRelation: '', * LegalGuardianVillage: '', | * LegalGuardianThana: '', * LegalGuardianDistrict: '', * LegalGuardianCountry: '', * LegalGuardianNID: '', * LegalGuardianPhone: '', * LocalGuardianName: '', * LocalGuardianRelation: '', * LocalGuardianVillage: '', * LocalGuardianPostOffice: '', * LocalGuardianPostCode: '', * LocalGuardianThana: '', * LocalGuardianDistrict: '', * LocalGuardianCountry: '', * LocalGuardianNID: '', * LocalGuardianPhone: '', * LegalGuardianPostOffice: '', * LegalGuardianPostCode: '', |

These values are used to insert or update data into database. And handlesubmit() method is used for submit data to database. Another method handlhange(event) is used to handle data change in those strings.

**Download Application:**

This class has a user\_id string value and fetchapplicantinfo () method is used for fetch applicant data from database. And MyPDF() method is used to generate pdf with applicant data. This document will be in (.pdf) form.

* 1. **Entity – Relationship diagram:**

The ER diagram of this system is given below.

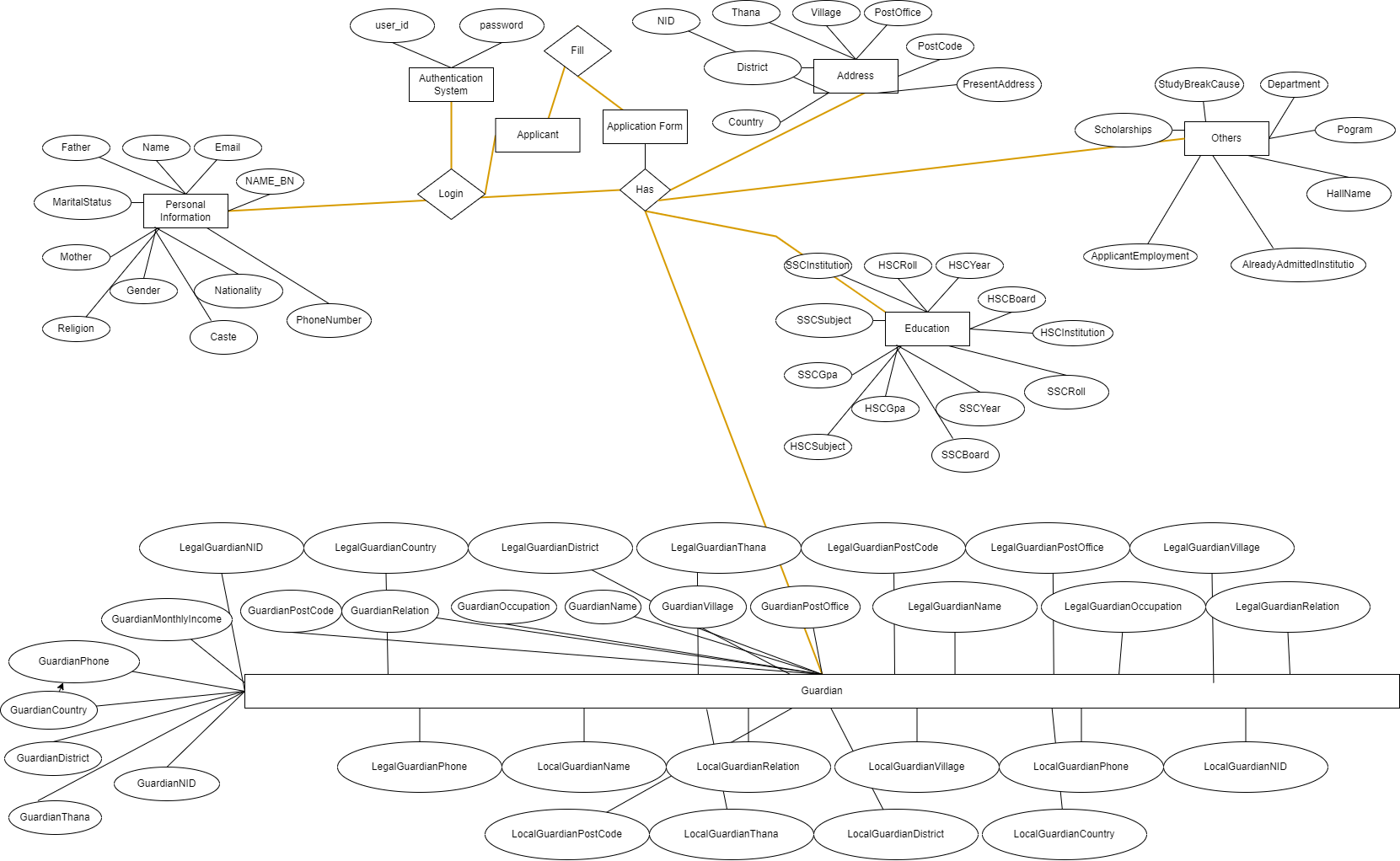


Fig – 3.5: ER diagram

This ER diagram make relation with the different entities. We have the relation like an Application can be filled by one applicant. On the other word we can say that the relation between w entities is one to one. Like that, Application entity has a relationship with

* Personal information
* Address
* Education
* Others and
* Guardian entity.

Every entity has a relationship with other entities and it is must for any system database.

* 1. **Schema diagram:**

The Schema diagram of this system is given below.

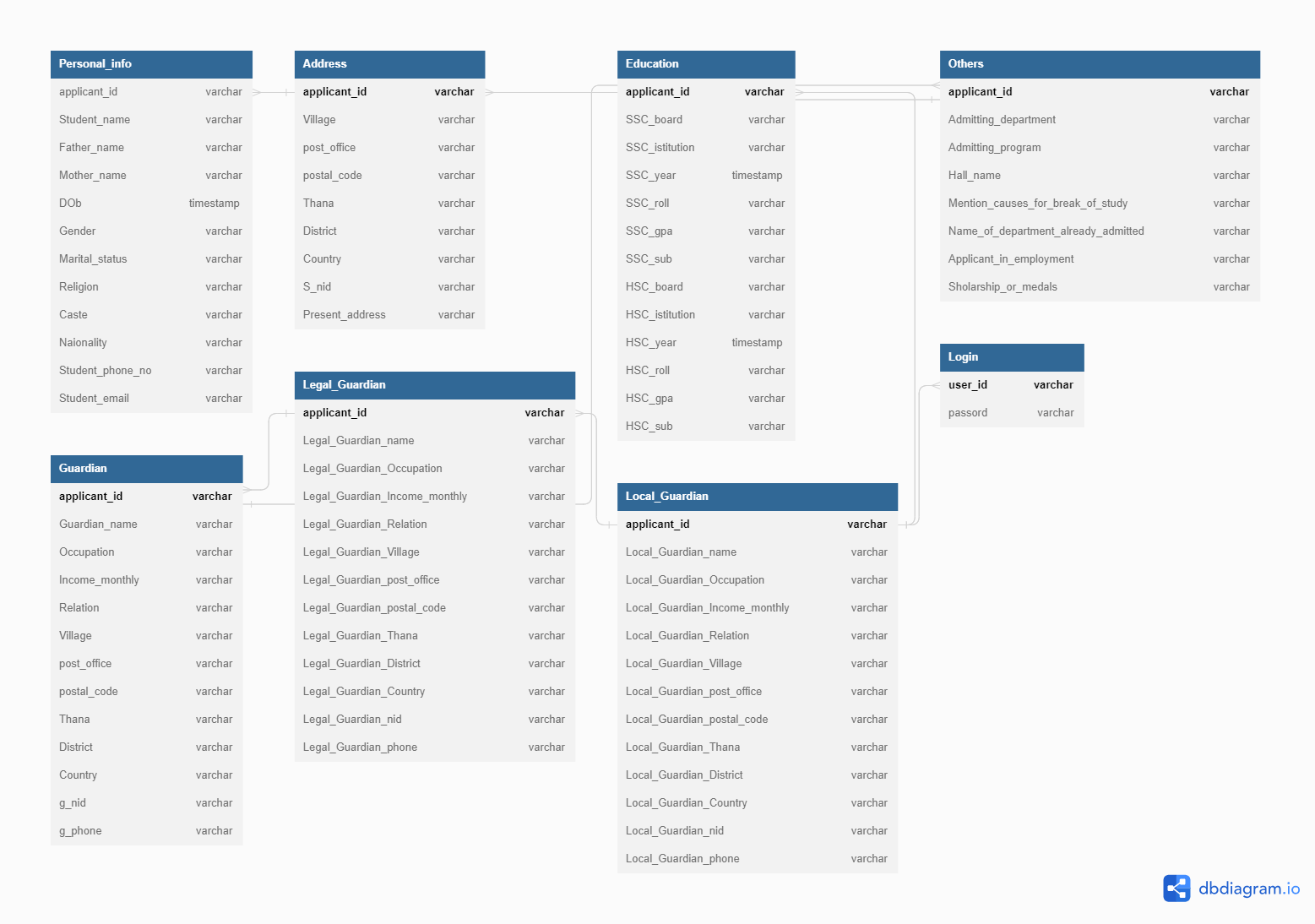


Fig – 3.6: Schema diagram

In this diagram we have primary keys and foreign keys. References of primary keys and foreign keys are stated bellow.

Ref: Personal\_info.applicant\_id > Address.applicant\_id

Ref: Address.applicant\_id > Others.applicant\_id

Ref: Others.applicant\_id > Guardian.applicant\_id

Ref: Guardian.applicant\_id > Legal\_Guardian.applicant\_id

Ref: Legal\_Guardian.applicant\_id > Local\_Guardian.applicant\_id

Ref: Education.applicant\_id > Local\_Guardian.applicant\_id

Ref: Login.user\_id > Local\_Guardian.applicant\_id

**Chapter 4**

**Implementation details**

* 1. **Authentication**

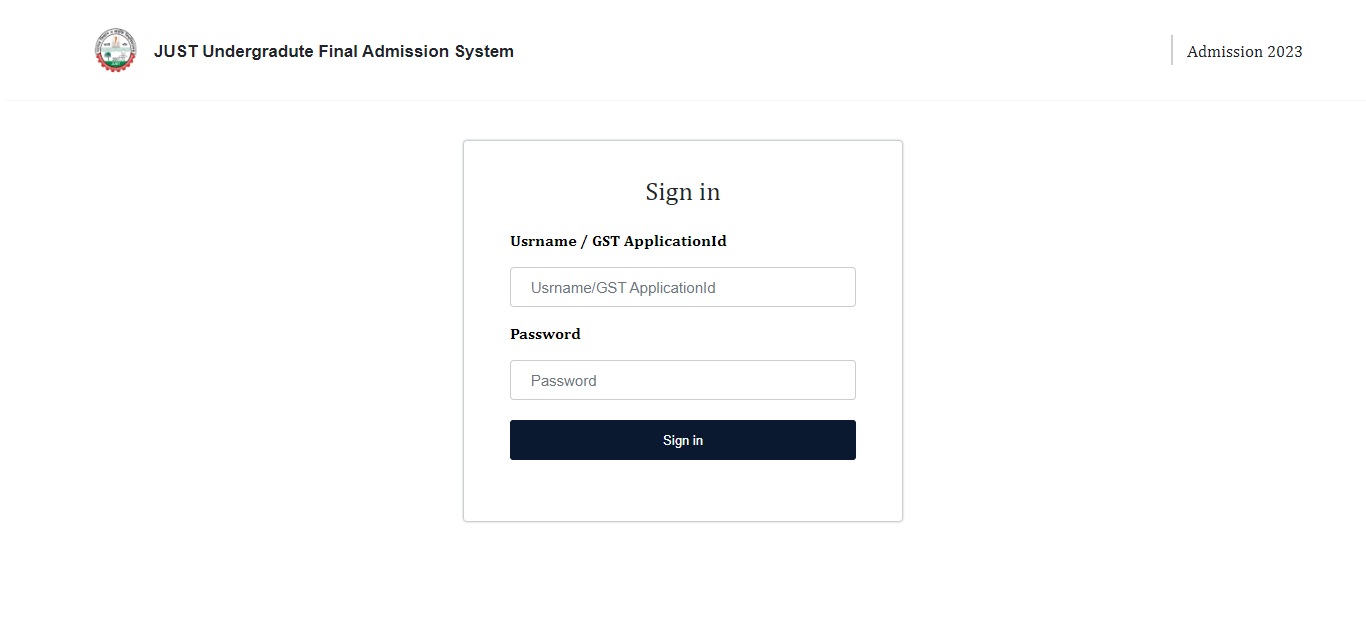


Fig: 4.1.1: Login page

* Login page has two input field named Username and Password.
* Here, for applicant GSTApplicantId is the username and GST password is the password.
* After that Sign in button will submit username and password and took the applicant to the main dashboard.

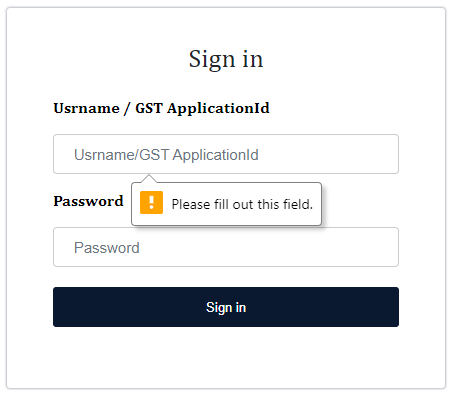


Fig: 4.1.2: Login page with empty username

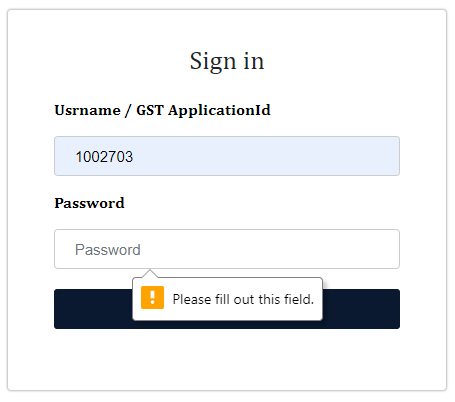


Fig: 4.1.3: Login page with empty password

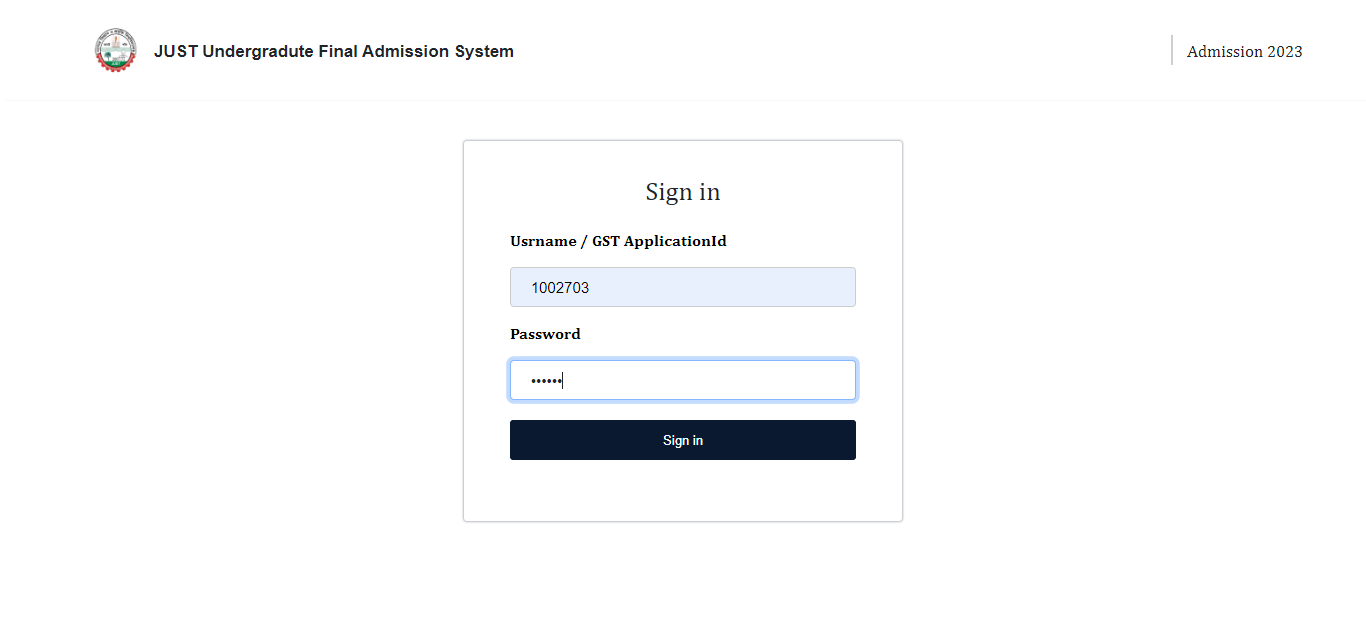


Fig: 4.1.4: Login page with username and password

* Here in figure 4.1.2 and 4.1.3 show that with empty field if we try to login then we will face this kind of scenario.
* On the other hand, in figure 4.1.4 shows the filled username and password with correct form. It can be logged in if this username and password was stored in database.
  1. **Dashboard**

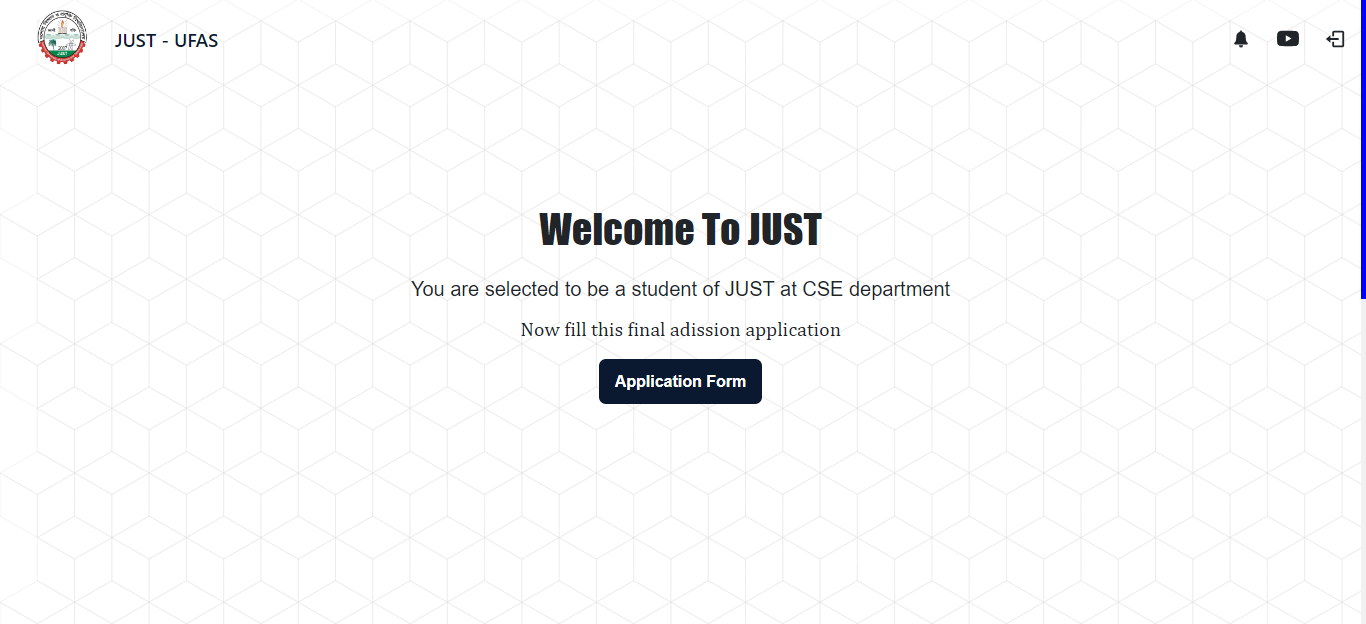


Fig: 4.2: Main dashboard

This is the main dashboard here we have three buttons on the header

* First button is created for showing notification send y authority.
* Second button is for a video tutorial of Application fillip. It is basically a YouTube video.
* And the last is used for logging out from the dashboard.

The application form button will navigate user to the application form.

* 1. **Application Page**

Application page has some buttons and steps. They are

* Back to home button – This button is used for go to main dashboard given in fig- 4.2. It is also called scroll up button.
* Here we have five steps this application. They are

1. Personal Information – here we have only applicant’s personal information like name, father’s name, mother’s name, DOB and so on.
2. Address – here we have only applicant’s address like village, post office, zip code, Thana, district, present address etc.
3. Education – here we have only applicant’s educational qualification like SSC and HSC exam result, roll, institution, subject etc.
4. Others – here we have only applicants additional and some vital information like department, program, job details (if any), and scholarships and so on.
5. Guardian – This step is the biggest and important step which contains applicant’s guardian’s information.

* We can go to any step in any time and update any information at any time. For this we need to press those steps to go to the step.
* Most importantly any of field can be in read only mode that mean hose information’s are not allowed to change.

**Personal Information:**

This is the first interface of application page. It is the first and starting interface which is Personal Information.



Fig: 4.3.1: Personal Information (1)

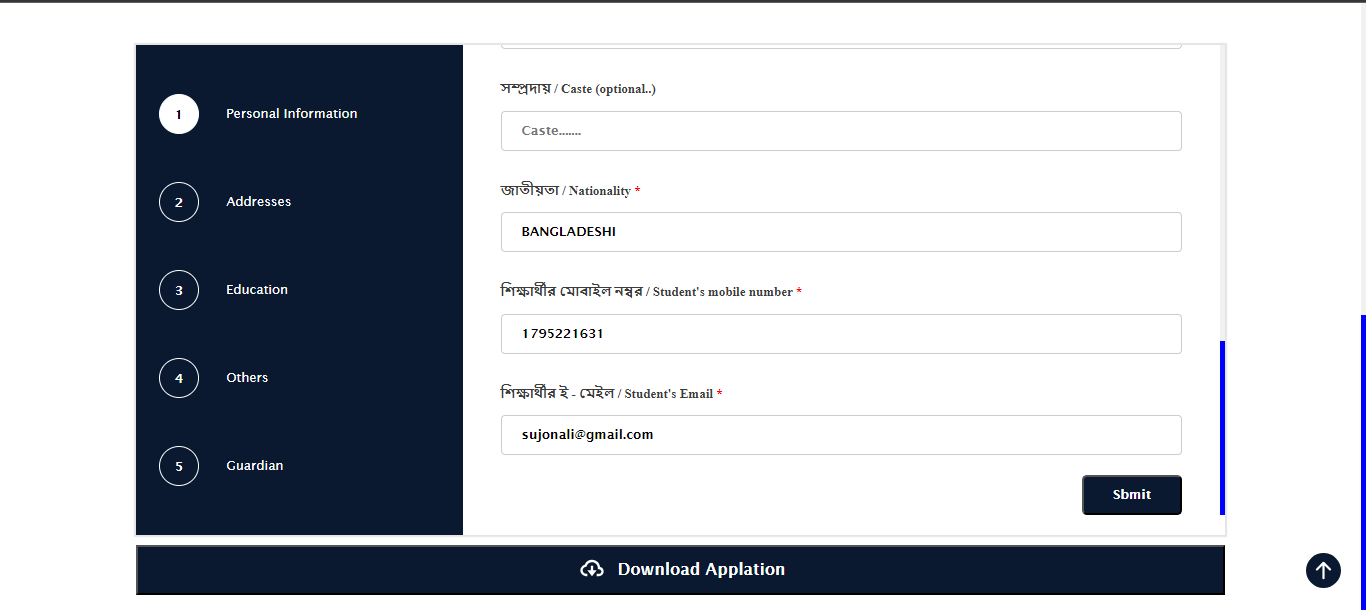


Fig: 4.3.2: Personal Information (2)

* In this step we may have some information like name, father’s name, DOB, gender, phone number and others. This will be on read-only mode. That mean we can’t change them.
* After filling this step we need to press the submit button shows in Fig: 4.3.2.

**Address:**

This is the second interface of application page. Here we will record the applicant’s address.

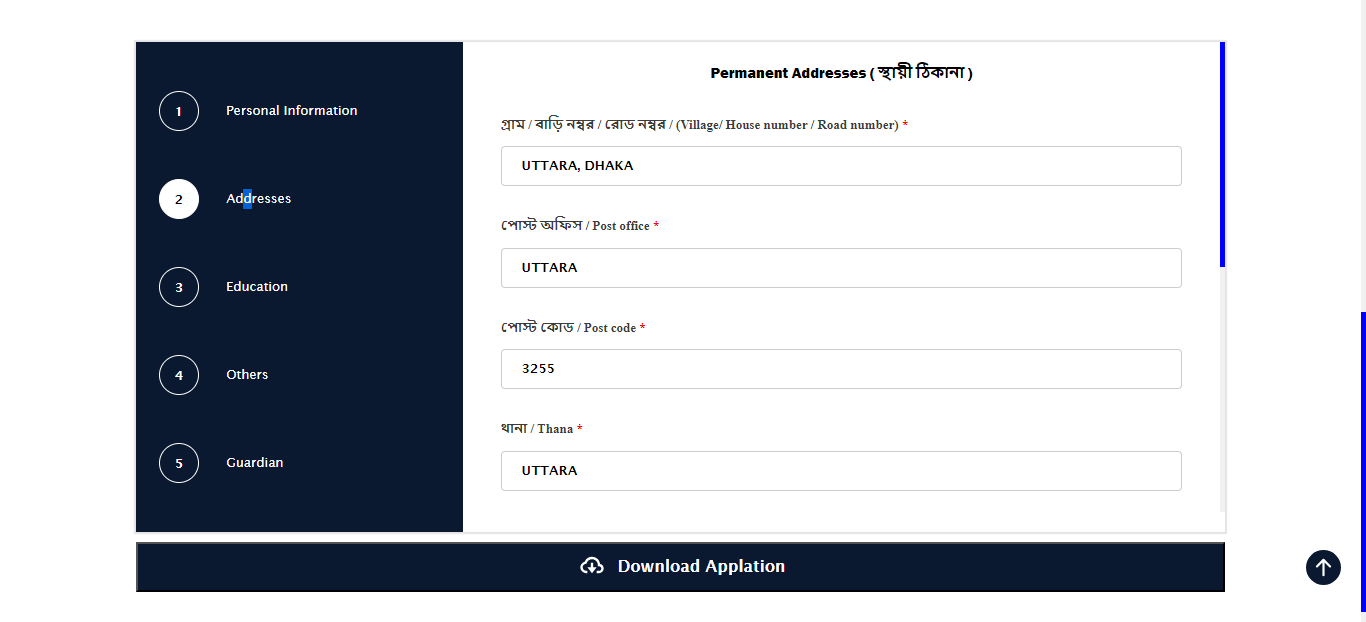


Fig: 4.3.3: Address (1)

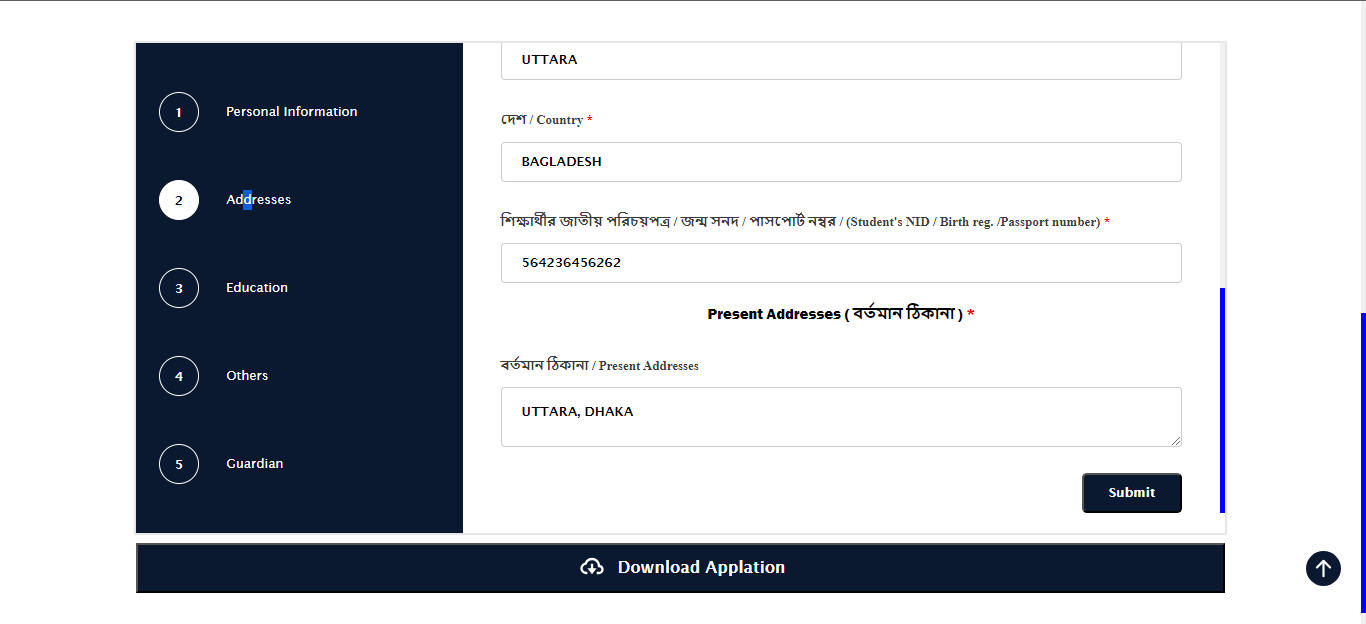


Fig: 4.3.4: Address (2)

* In this step we have addresses of applicant present and permanent.
* Permanent address contains with Village, Post Office, Postcode: '', Thana: '', District: '', Country: '', NID: '',
* Submit button is used to submit those information to database.
* Submit button is also used forgo o next step too.

**Education:**

This is the third interface of application page. Here we will record the applicant’s educational qualifications of SCC and HSC exam.

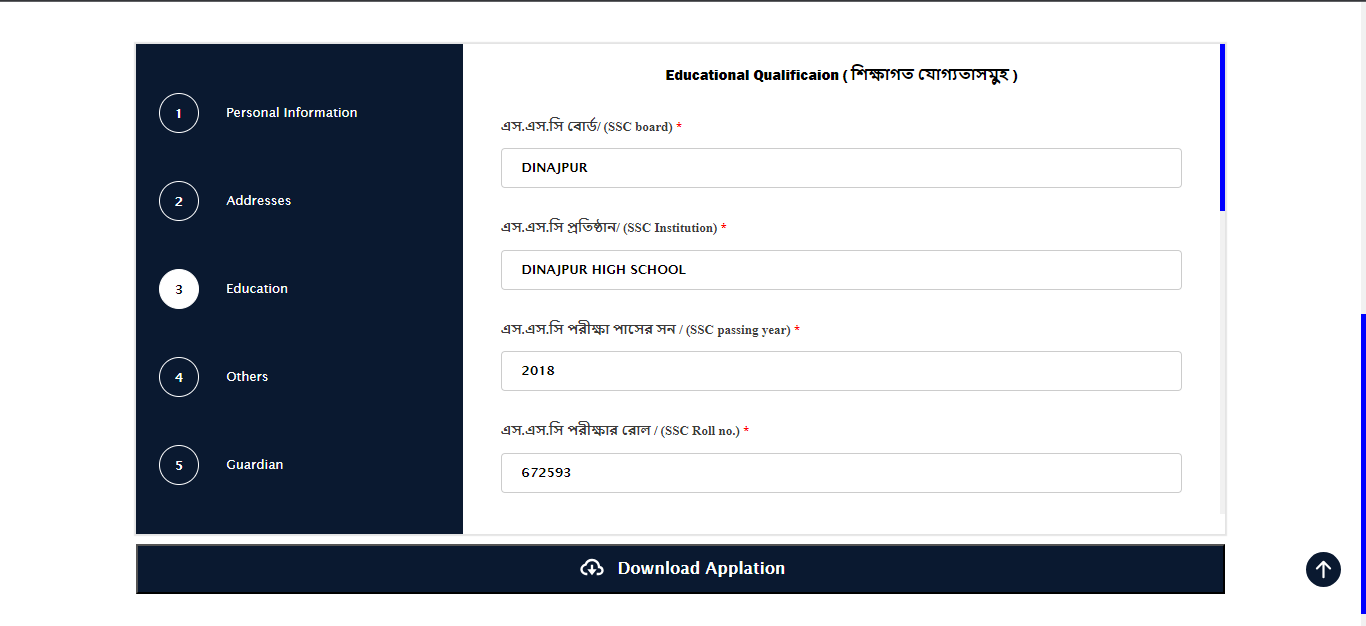


Fig: 4.3.5: Education (1)

* Here we have SSC exam information field where most of the fields are protective.
* Only institution name can be inserted / updated.



Fig: 4.3.6: Education (2)

* Here we have HSC exam information field where most of the fields are protective.
* Only institution name can be inserted / updated.
* Here in Fig: 4.3.4 and Fig: 4.3.5 has these fields HSCRoll: '', HSCYear: '', HSCBoard: '', SSCRoll: '',  SSCYear: '', SSCBoard: '', HSCGpa: '', HSCSubject: '', SSCGpa: '', SSCSubject: '', SSCInstitution: '', HSCInstitution: '',
* Submit button is used to submit those information to database.
* Submit button is also used for going to next step too.

**Others:**

This is the fourth interface of application page. Here we will record the applicant’s additional and important information too.

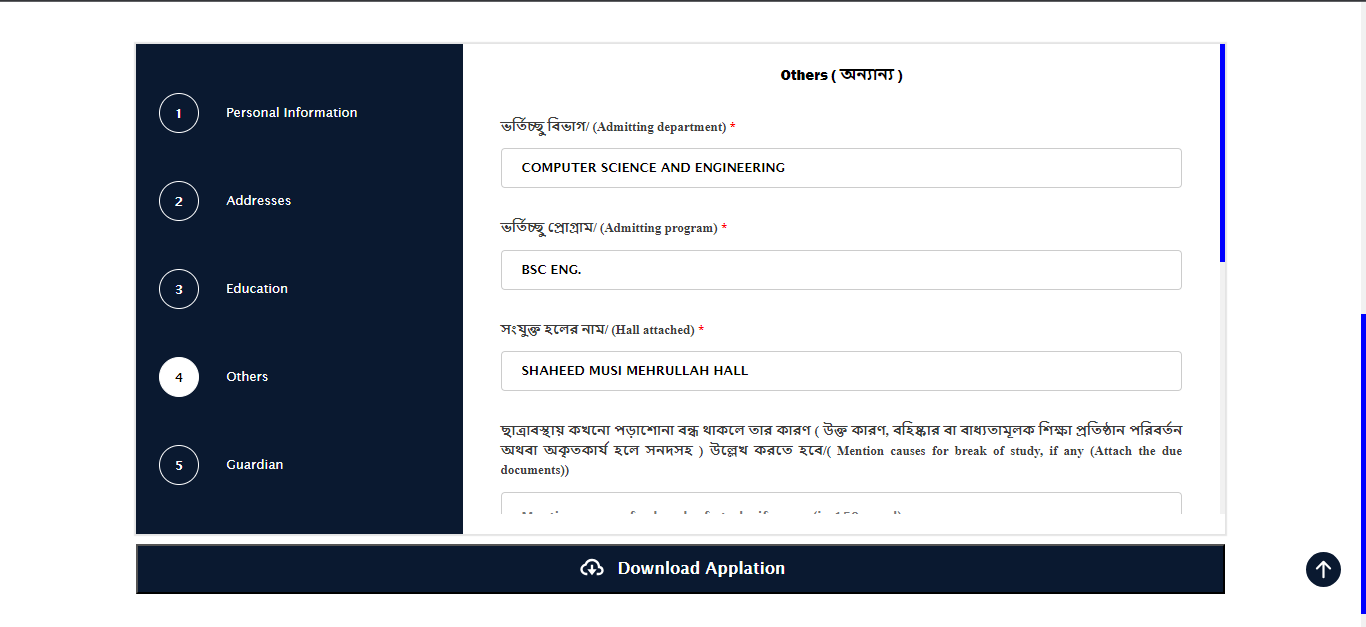


Fig: 4.3.7: Others (1)

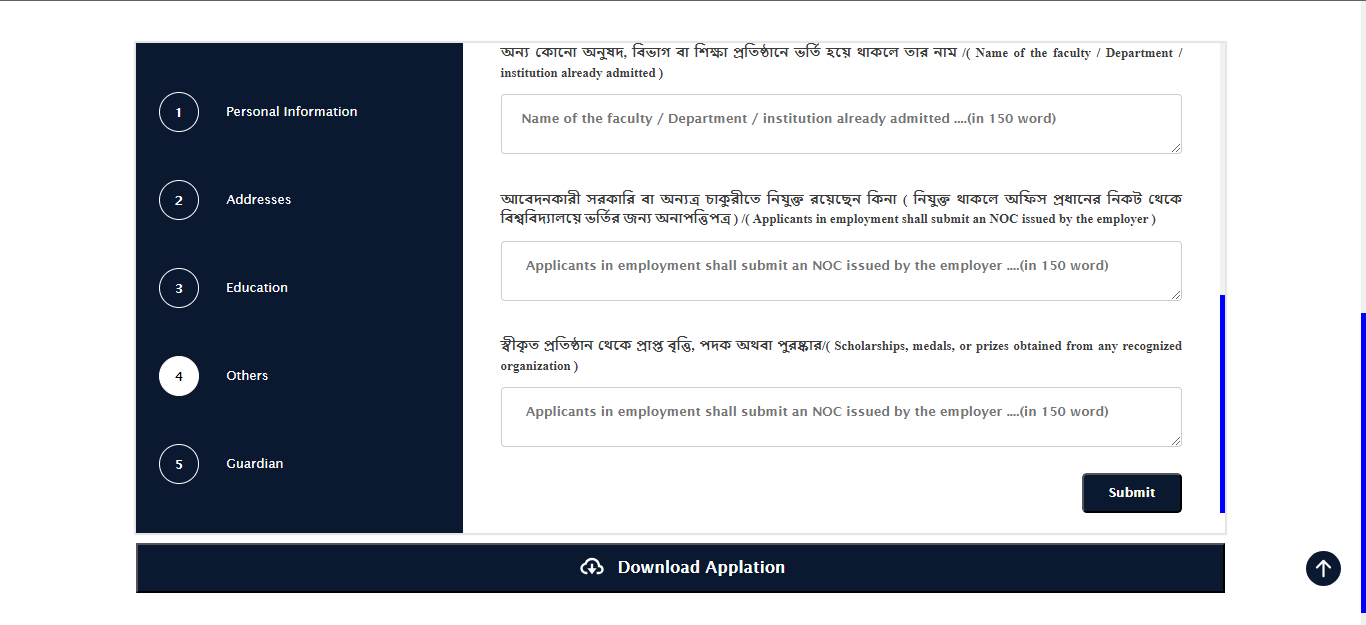


Fig: 4.3.8: Others (2)

* The vital information in others are Department: '', Pogram: '', HallName: '',
* And the additional information are StudyBreakCause:” ”, AlreadyAdmittedInstitutio: '',ApplicantEmployment: '', Scholarships: '',

**Guardian page:**

This is the fifth and last interface of application page. Here we will record the applicant’s guardian’s and important information too. This step is the biggest and important step which contains applicant’s guardian’s information.

We have three types of information here.

1. Guardian:
2. Legal guardian in the absence of father
3. Local guardian.

**Guardian:**



Fig: 4.3.9: Guardian (1)



Fig: 4.3.10: Guardian (2)

Here we have guardian’s information like GuardianName: '',GuardianOccupation: '',GuardianMonthlyIncome: '', GuardianRelation: '',GuardianVillage: '', GuardianPostOffice: '',GuardianPostCode: '',GuardianThana: ''GuardianDistrict: '', GuardianCountry: '', GuardianNID: '', GuardianPhone: '',

**Legal Guardian:**

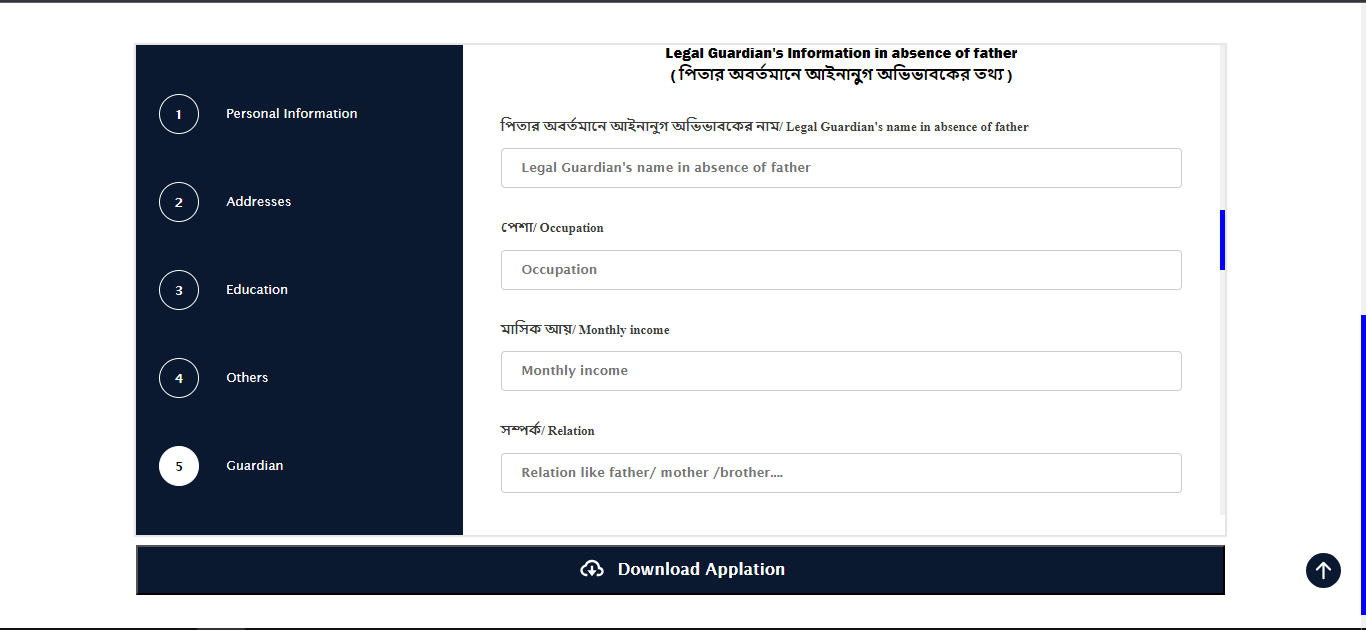


Fig: 4.3.11: Legal Guardian (1)

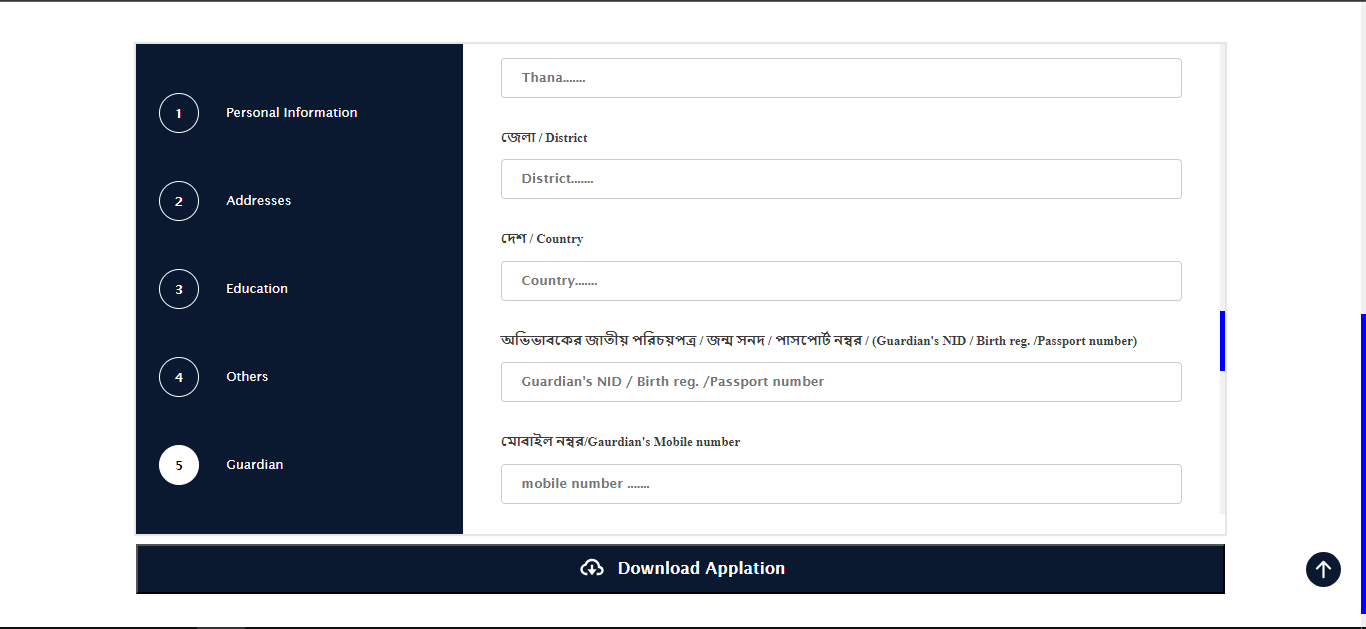


Fig: 4.3.12: Legal Guardian (2)

Here we have guardian’s information like LegalGuardianName: '',LegalGuardianIncome: '',LegalGuardianOccupation: '',LegalGuardianRelation: '',LegalGuardianVillage: '',LegalGuardianPostOffice: '',LegalGuardianPostCode: '',LegalGuardianThana: '',LegalGuardianDistrict: '',LegalGuardianCountry: '',LegalGuardianNID: '',LegalGuardianPhone: '',

This field is only required hen applicant’s father is absent as guardian.

**Local Guardian:**

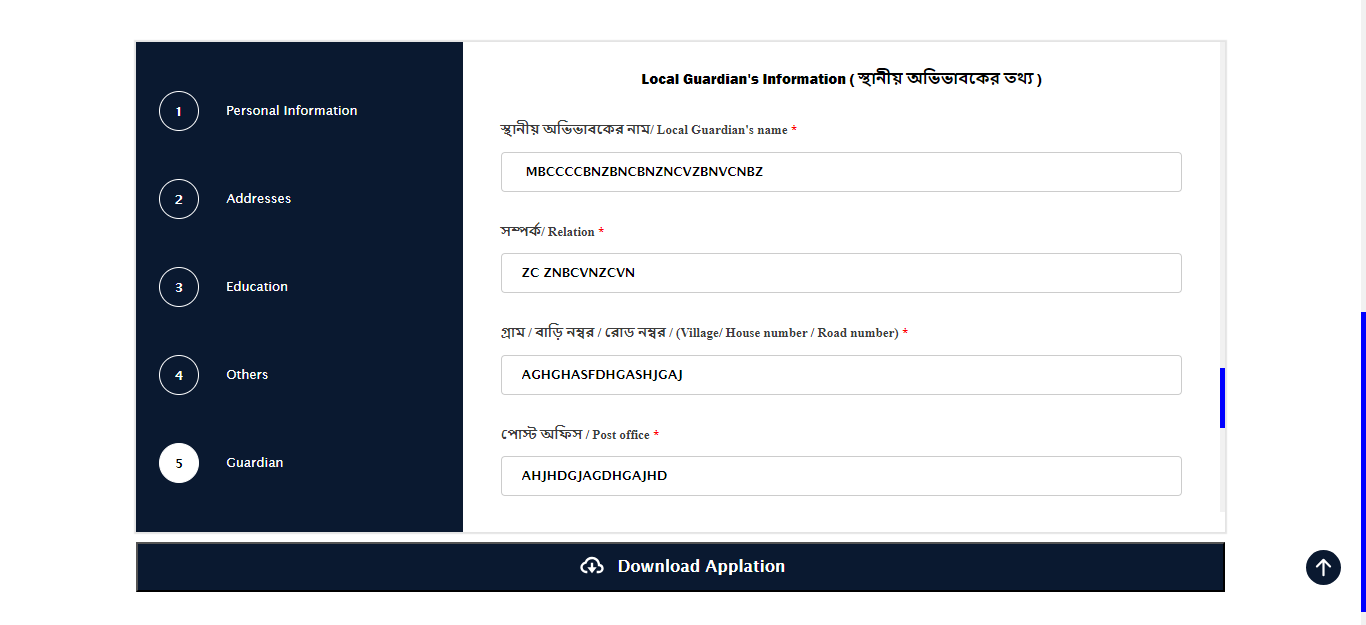


Fig: 4.3.13: Local Guardian (1)

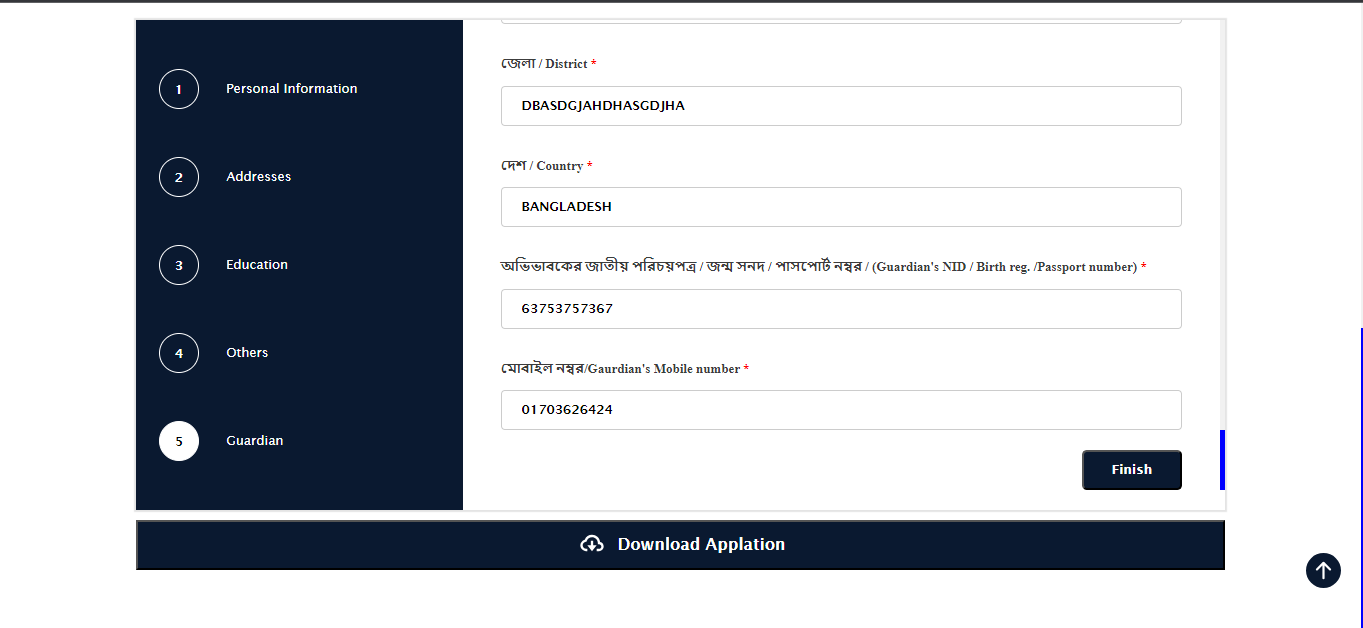


Fig: 4.3.14: Local Guardian (2)

Here we have guardian’s information like LocalGuardianName: '',LocalGuardianRelation: '',LocalGuardianVillage: '',LocalGuardianPostOffice: '',LocalGuardianPostCode: '',LocalGuardianThana: '',LocalGuardianDistrict: '',LocalGuardianCountry: '',LocalGuardianNID: '',LocalGuardianPhone: '',

**Chapter 5**

**Testing**

* 1. **White Box Testing**

White box testing is a type of testing that focuses on the internal structure of a software application. In the context of web applications, white box testing involves testing the application's code, including the algorithms, data structures, and other implementation details. There are several approaches to white box testing for web applications, including:

**Unit testing:**

This involves testing individual units of code, such as functions or methods, to ensure that they are working correctly.

**Integration testing:** This involves testing how different units of code work together, to ensure that they are integrated correctly and produce the expected results.

**Code coverage testing:** This involves testing how much of the code is executed during testing, to ensure that all code paths are exercised and that the entire application is tested.

**Mutation testing:** This involves making small, deliberate changes to the code (called "mutations"), and then rerunning the tests to ensure that they fail. This helps to ensure that the tests are actually able to detect defects in the code.

Overall, the goal of white box testing is to ensure that the code of a web application is correct and free of defects and that it is implemented according to the design specifications. In our project, the two types of white box testing are unit testing and integrating testing. All functions and methods of the system worked perfectly in unit tests. It is giving correct output according to the input. Integrating tests are done by combining functions. And giving correct output as per input.

* 1. **Black Box Testing**

Black box testing is a type of testing that focuses on the external behavior of a software application, without looking at its internal structure or implementation details. In the context of web applications, black box testing involves testing the application from the user's perspective, by providing inputs and verifying the outputs. There are several approaches to black box testing for web applications, including: JUST Admission Auto Migration System 47 Functional testing: This involves testing the application's functions and features to ensure that they are working correctly and as expected. Usability testing: This involves testing the user experience of the application, to ensure that it is easy to use and understand. Compatibility testing: This involves testing the application on different devices, browsers, and operating systems to ensure that it works correctly in different environments. Performance testing: This involves testing the application's performance, such as its response time and stability under different load conditions. Overall, the goal of black box testing is to ensure that the application is working correctly from the user's perspective and meets the specified requirements. All the above types of black box testing are done in my project. All performed well in black box testing. As a result, my project is still giving good performance.

**Chapter 6**

**Deployment**

Deploying a web application typically involves the following steps:

**Building and testing the application:**

This typically involves writing and testing the code, and packaging it into a deployable form (such as a WAR file for Java applications).

**Setting up the production environment:**

This involves setting up the servers, databases, and other infrastructure needed to run the application in production. This may involve configuring load balancers, web servers, and other components.

**Deploying the application:**

This involves transferring the application code and any necessary dependencies (such as libraries or configuration files) to the production environment and installing it on the servers.

**Configuring the application:**

This involves setting any necessary application-specific configuration, such as database connection strings or environment-specific settings.

**Testing the deployment:**

After the application is deployed, it is important to test it to ensure that it is working correctly in the production environment. This may involve running automated tests or manually testing the application's functions.

**Maintaining and monitoring the application:** After the application is deployed, it is important to monitor it for any issues or performance problems, and to perform regular maintenance to keep it up to date and running smoothly.

Now, my project is deployed on localhost and its activities are being managed. When we deploy this project, we will follow the above steps to deploy our site as well as maintain my project.

**Conclusions**

In general, the "JUST Undergraduate Admission System" potentially offers several benefits, such as the online application process and central database use working more quickly and accurately than manual processes. This system has reduced the risk of errors that occur with manual processes. The admission system have reduced the extra risk, that resulting in smooth process. It is important to note that the specific benefits and conclusions of the admission system will depend on the specific needs and goals of the university implementing it.

**Future Work**:

Previously we said that is a partially fulfillment of admission system of JUST. SO, future plans for this project. Is to complete all requirement and office works for undergraduate admission.