

SSDI Project Report

Online Application to Predict Home Loan Eligibility & Estimated Amount

*Suhail Lala(slala@uncc.edu),
Venkat Bandaru (ybandaru@uncc.edu),
Deep Prajapati(dprajap1@uncc.edu),
Ajithesh Navaneethakrishnan(anavanee@uncc.edu)*

Table of Contents

1. Project Description.....	3
2. Software Process Model.....	4
3. Requirements Analysis.....	4
4. Design and Implementation.....	5
5. UML Diagrams.....	12
6. Testing (Use cases)	20
7. Step by Step Illustration of Functionality (with screenshots)	21
8. Results and conclusion.....	31
9. Github URL - Package Explorer.....	35

Online Application to Predict Home Loan Eligibility & Estimated Amount

1. Project Description

Today an applicant has to go through several formalities to obtain a loan. This effort would be for naught if his application gets denied.

Through this project, we provide a web application where an applicant can enter his/her details online and get to know if his/her loan application would be approved or not. If approved, the web application also estimates the eligible loan amount for the applicant.

Apart from this basic functionality, our application also contains a homepage, help, about-us, faqs and contact-us page. Moreover, we used responsive design to ensure that our website works well with different form-factors/channels **like mobile browser support**.

This application is targeted towards any individual who wants to find out his/her chances/eligibility of getting a loan and the estimated amount that he/she may receive from the loan.

2. Software Process Model-Incremental Model

We have chosen the Incremental model where we divided the whole requirement into various builds. We also know the major requirements definition, so this model was apt for us. We carried out multiple development cycles and we divided each cycle into smaller, easily managed modules. We followed this process till the whole project is complete. We chose the Incremental model as we found out the following advantages:

- Generates working software quickly and early during the software life cycle.

- It is more flexible and less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customers can respond to each build.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

3. Requirements Analysis

Requirement 1: Interface - System should predict a person's loan repayment ability.

Support: System will use pre-trained Machine Learning (ML) models which can predict the loan eligibility. System will be deployed through cloud computing platform(s). An Interface will be provided to interact with the model.

Requirement 2: Interface - System should estimate the loan amount.

Support: System will use pre-trained ML models which can also predict estimated eligible loan amount for the applicant. System will be deployed through cloud computing platform(s). An Interface will be provided to interact with the model.

Requirement 3: System should have Loan eligibility form.

Support: System will have a form to get applicant details to predict home loan eligibility.

Requirement 4: System should display Decision page (loan result).

Support: System will display results to the applicant, output from the ML models.

Requirement 5: System should have a Help & FAQ page to support application users.

Support: System will have a static help & faq page, guiding how to fill application forms.

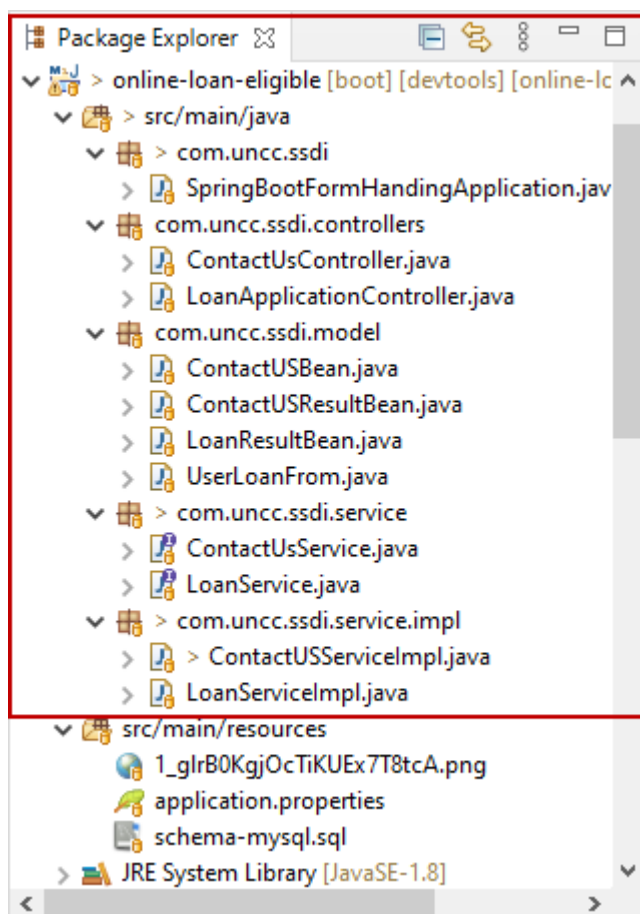
Requirement 6: System should have a Contact Us page where users should be able to contact when the need arises.

Support: System will have a contact us page; users will have to fill the contact us form.

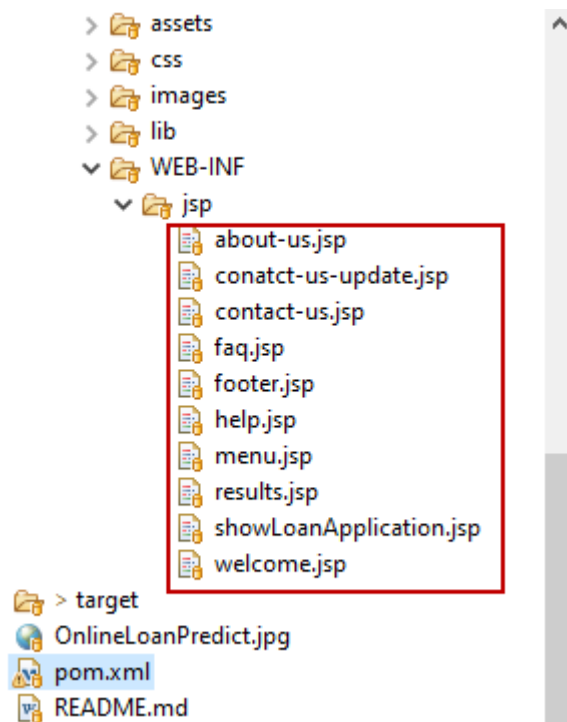
4. Design and Implementation

Overall Design

We have used Java Programming language along with Spring Boot , JSP as our front end and used Spring Model View Controller Framework as backend with 4 model classes, two services classes and two controller class and 10 jsp pages....below is package explorer view.



JSPs



Other technologies used - Java 1.8, J2EE, JSP/HTML5, Spring Boot server, Cloud technologies, Restful APIs, Spring boot Microservices. (Other tools JS/jQuery.).

High level design details

A user typically interacts with the loan eligibility form, putting in all his details. This data is processed in the backend, formatted, and sent to the **Azure endpoint** where we have deployed Machine Learning models. Within Azure, this data is validated and passed through a model which determines the user's loan eligibility. If the user is eligible, the data is passed through a second model which estimates the loan amount that the user is expected to receive. Finally, Azure returns the loan value or 0, depending upon the eligibility of the user. Our application catches this output and displays it on the results page.

Azure endpoint for Machine Learning Interface - <http://52.167.63.182:80/api/v1/service/ml-deploy/score>

Below fields are required per interface

- **CODE_GENDER**
- **FLAG_OWN_CAR**
- **CNT_CHILDREN**
- **AMT_INCOME_TOTAL**
- **AMT_CREDIT**
- **NAME_EDUCATION_TYPE**
- **NAME_FAMILY_STATUS**
- **DAYS_BIRTH**
- **DAYS_EMPLOYED**

- **FLAG_PHONE**
- **OCCUPATION_TYPE**

The web application is calling above ML interface to determine Loan eligibility and estimated loan amount approvals.

Application URL - <http://localhost:8081/welcome>

This project involved the following development aspects:

- Developed a web form that a user can fill if he wants to obtain information about his loan eligibility.
- Developed a page to display results per user filled information.
- Developed an interface to connect and send/receive data from the ML models.
- Deployed ML models to Azure, configuring them, and creating endpoint and token to connect with the web application.
- Developed help, faq, contact us and about us pages.

Developed Responsive WebSite

Application is using **Java X mail API** for sending mail as part of Contact US.

Technologies	<p>Use the following technologies to build the requirement</p> <p>HTML5</p> <ul style="list-style-type: none"> • <i>Design a form</i> • <i>Provide a button etc.</i> <p>CSS3</p> <ul style="list-style-type: none"> • <i>Apply background image for the HTML Body with appropriate class name</i> • <i>Apply background color for the header.</i> • <i>Apply transition effect.</i> <p>Bootstrap</p> <ul style="list-style-type: none"> • <i>Make a responsive border table.</i> • <i>Make the page responsive.</i> <p>JavaScript</p> <ul style="list-style-type: none"> • <i>Validate the HTML form using JavaScript code</i> <p>JQuery</p> <ul style="list-style-type: none"> • <i>Add controls to the buttons.</i> • <i>Iterate and display the data</i> • <i>Toggle the buttons (save or update) based on the action(save or edit).</i> <p>AJAX</p> <ul style="list-style-type: none"> • <i>Call the API's for read, save, edit, delete user info using AJAX.</i> • <i>Call the API' to interact with ML Model for prediction and eligible amount</i> <p>API Calls</p> <ul style="list-style-type: none"> • <i>To get the data from ML model –</i> <ul style="list-style-type: none"> ○ <i>Call: http://xxxxxxx</i> ○ <i>method GET</i> • <i>User Details</i> <ul style="list-style-type: none"> ○ <i>Call: <a href="http://xxxxx/user/<id>">http://xxxxx/user/<id></i> ○ <i>All CRUD methods</i> ○ <i>In the place of the <id> substitute the id of the user as applicable</i> <p>Cloud Technologies</p> <p>Azure/AWS</p>
---------------------	---

Project Swagger Documentation

The Entire project can be generated and be viewed under Swagger Documentation few sample screenshot below..

Swagger URL - <http://localhost:8081/v2/api-docs>
<https://editor.swagger.io/>

The image shows the Swagger Editor interface in a web browser. The address bar displays `editor.swagger.io`. The editor has a dark theme and a sidebar on the left for file management. The main area is split into two panes: a code editor on the left and a visual API explorer on the right.

Swagger Specification (Left Pane):

```
1 swagger: '2.0'
2 info:
3   description: Api Documentation
4   version: '1.0'
5   title: Api Documentation
6   termsOfService: 'urn:tos'
7   contact: {}
8   license:
9     name: Apache 2.0
10    url: 'http://www.apache.org/licenses/LICENSE-2.0'
11 host: 'localhost:8081'
12 basePath: /
13 tags:
14   - name: basic-error-controller
15     description: Basic Error Controller
16   - name: contact-us-controller
17     description: Contact Us Controller
18   - name: loan-application-controller
19     description: Loan Application Controller
20 paths:
21   /about-us:
22     get:
23       tags:
24         - loan-application
```

API Explorer (Right Pane):

The right pane displays the API endpoints for the `contact-us-controller`. The endpoints are listed with their HTTP methods, paths, and descriptions:

- GET** `/contact-us` `contactusPage`
- HEAD** `/contact-us` `contactusPage`
- POST** `/contact-us` `processContactUSRequest`
- PUT** `/contact-us` `contactusPage`
- DELETE** `/contact-us` `contactusPage`
- OPTIONS** `/contact-us` `contactusPage`
- PATCH** `/contact-us` `contactusPage`

← → ↻ editor.swagger.io

Swagger Editor Supported by SMARTBEAR File Edit Generate Server Generate Client

```
1 swagger: '2.0'
2 info:
3   description: Api Documentation
4   version: '1.0'
5   title: Api Documentation
6   termsOfService: 'urn:tos'
7   contact: {}
8   license:
9     name: Apache 2.0
10    url: 'http://www.apache.org/licenses/LICENSE-2.0'
11 host: 'localhost:8081'
12 basePath: /
13 tags:
14   - name: basic-error-controller
15     description: Basic Error Controller
16   - name: contact-us-controller
17     description: Contact Us Controller
18   - name: loan-application-controller
19     description: Loan Application Controller
20 paths:
21   /about-us:
22     get:
23       tags:
24         - loan-application-controller
25       summary: aboutusPage
26       description: About Us Page
27       operationId: aboutusPageUsingGET
28       produces:
29         - '*/*'
```

DELETE /help helpPage

OPTIONS /help helpPage

PATCH /help helpPage

GET /submitLoanApplication show

POST /submitLoanApplication processRequest

GET /welcome firstPage

HEAD /welcome firstPage

POST /welcome firstPage

PUT /welcome firstPage

DELETE /welcome firstPage

editor.swagger.io

Swagger Editor. Supported by SMARTBEAR. File Edit Generate Server Generate Client

```

1 swagger: '2.0'
2 info:
3   description: Api Documentation
4   version: '1.0'
5   title: Api Documentation
6   termsOfService: 'urn:tos'
7   contact: {}
8   license:
9     name: Apache 2.0
10    url: 'http://www.apache.org/licenses/LICENSE-2.0'
11 host: 'localhost:8081'
12 basePath: /
13 tags:
14   - name: basic-error-controller
15     description: Basic Error Controller
16   - name: contact-us-controller
17     description: Contact Us Controller
18   - name: loan-application-controller
19     description: Loan Application Controller
20 paths:
21   /about-us:
22     get:
23       tags:
24         - loan-application-controller
25       summary: aboutusPage
26       description: About Us Page
27       operationId: aboutusPageUsingGET
28       produces:
29         - '*/*'

```

POST /contact-us processContactUSRequest

Process Contact US Form

Parameters

Name	Description
email	email
string (query)	
message	message
string (query)	
name	name
string (query)	
subject	subject
string (query)	

editor.swagger.io

Swagger Editor. Supported by SMARTBEAR. File Edit Generate Server Generate Client

```

1 swagger: '2.0'
2 info:
3   description: Api Documentation
4   version: '1.0'
5   title: Api Documentation
6   termsOfService: 'urn:tos'
7   contact: {}
8   license:
9     name: Apache 2.0
10    url: 'http://www.apache.org/licenses/LICENSE-2.0'
11 host: 'localhost:8081'
12 basePath: /
13 tags:
14   - name: basic-error-controller
15     description: Basic Error Controller
16   - name: contact-us-controller
17     description: Contact Us Controller
18   - name: loan-application-controller
19     description: Loan Application Controller
20 paths:
21   /about-us:
22     get:
23       tags:
24         - loan-application-controller
25       summary: aboutusPage
26       description: About Us Page
27       operationId: aboutusPageUsingGET
28       produces:

```

POST /submitLoanApplication processRequest

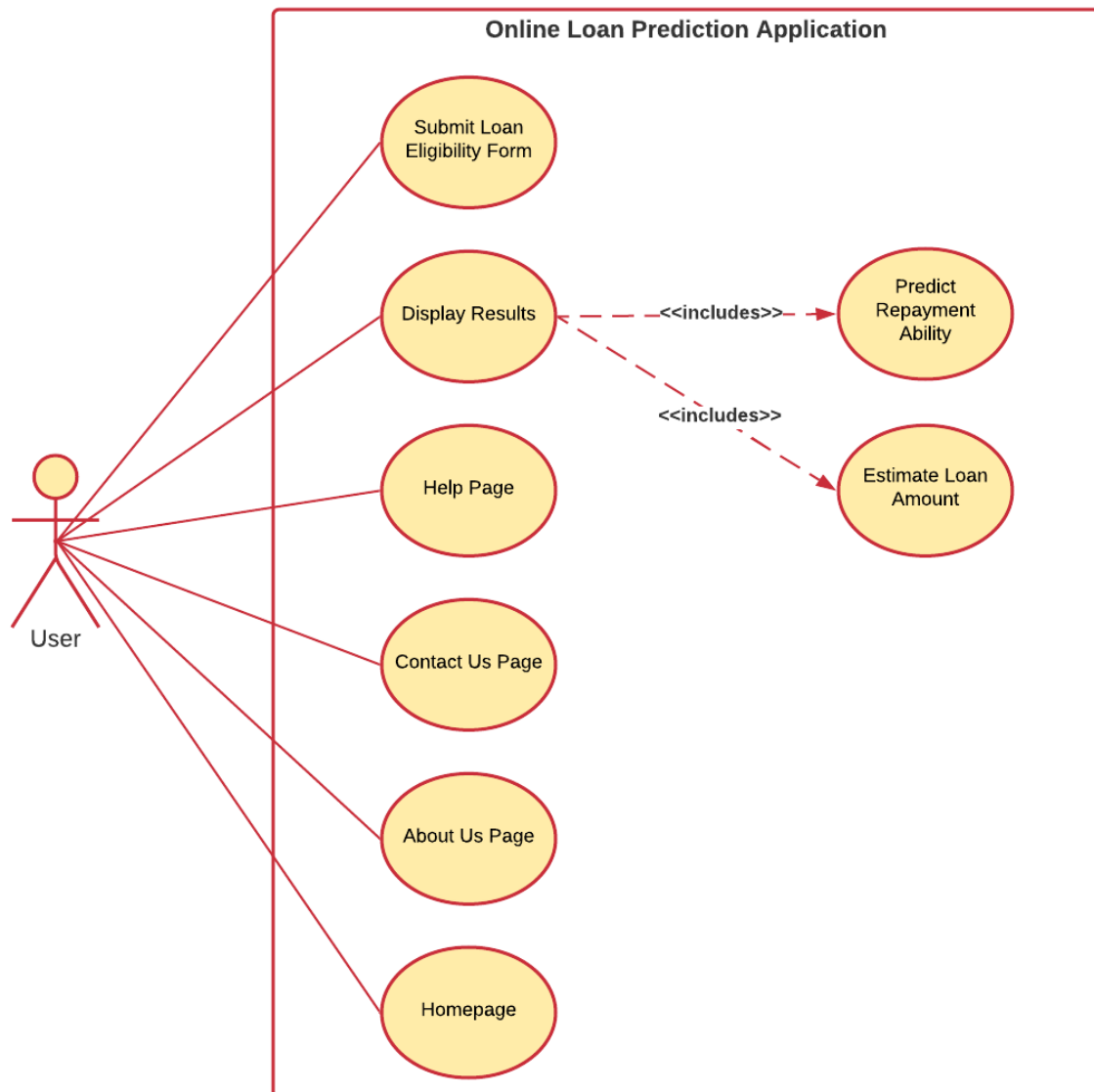
Process Loan Application

Parameters

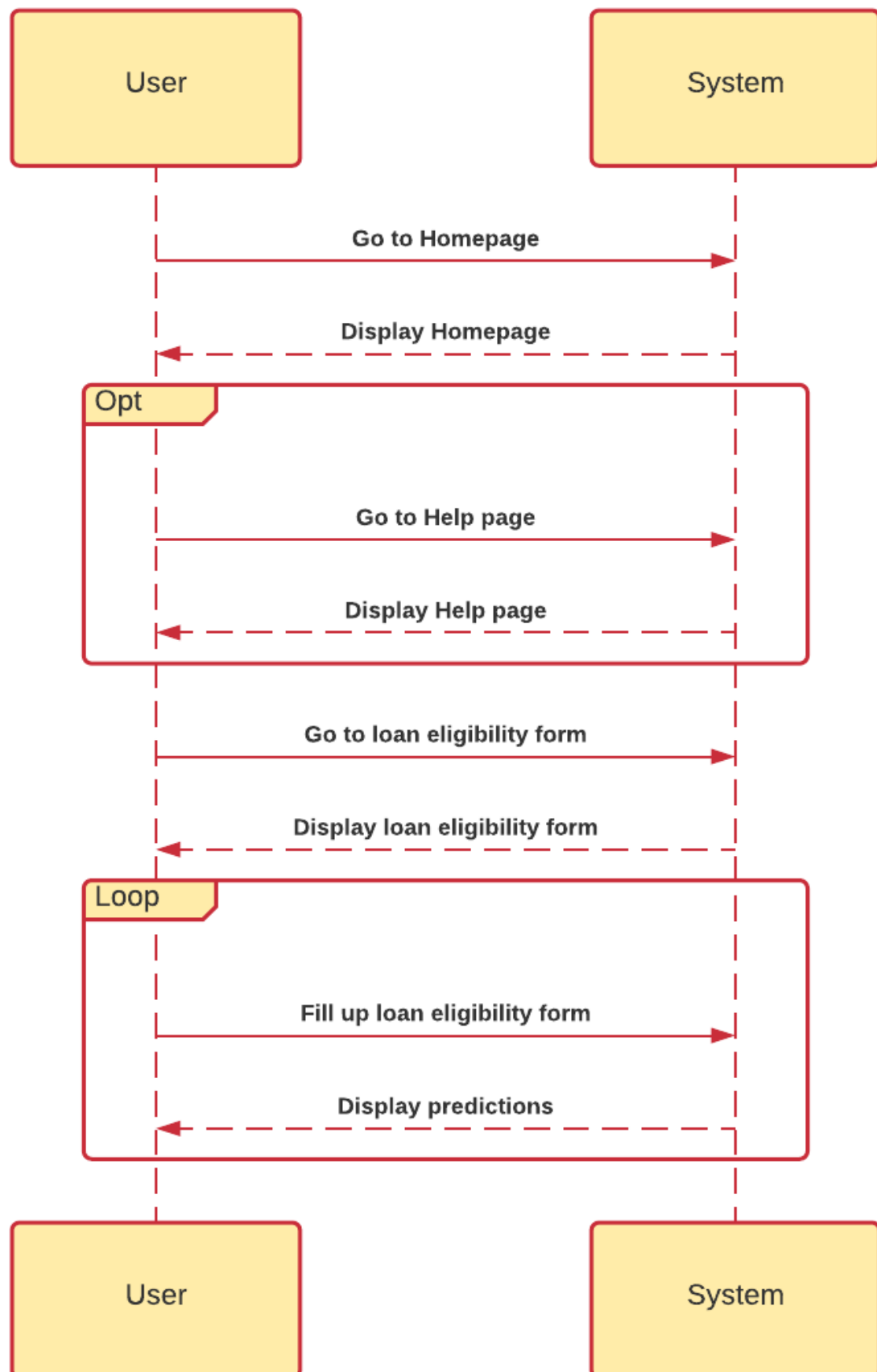
Name	Description
dob	dob
string (query)	
educationType	educationType
string (query)	
employedDays	employedDays
string (query)	
familyStatus	familyStatus
string (query)	

5. UML Diagrams

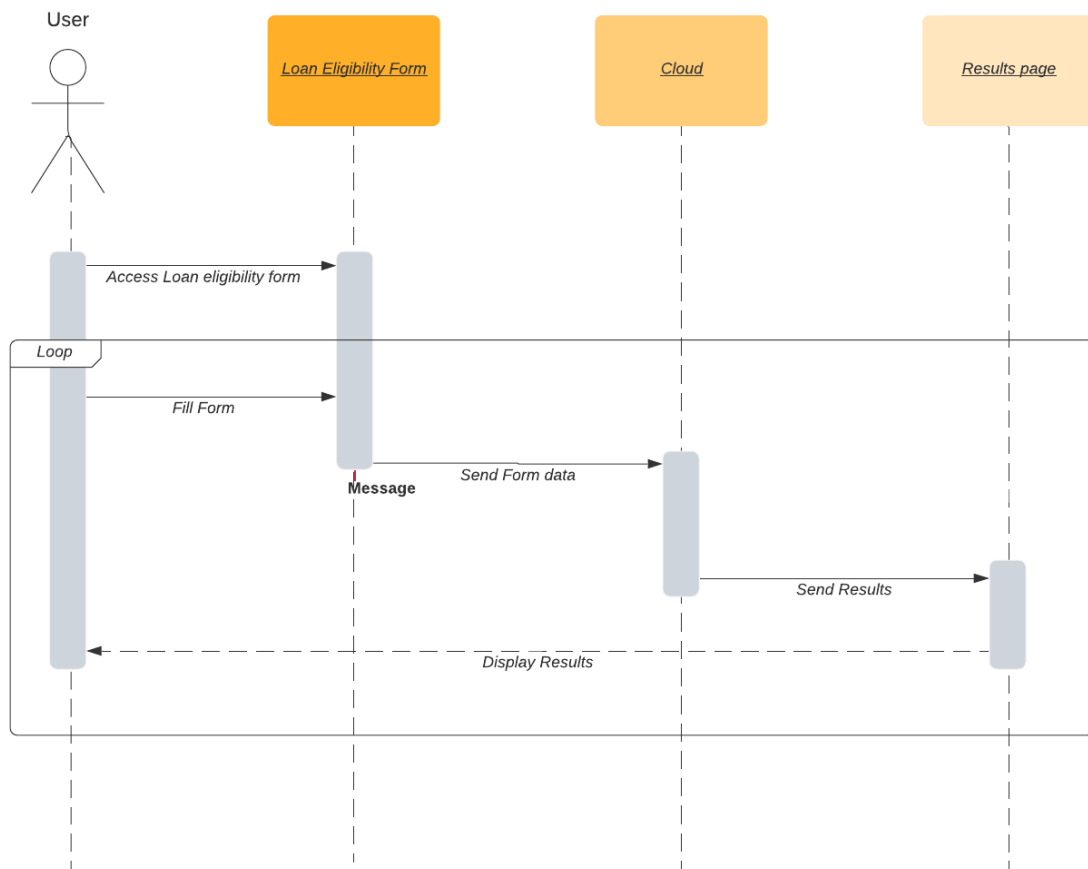
Use Case Diagram



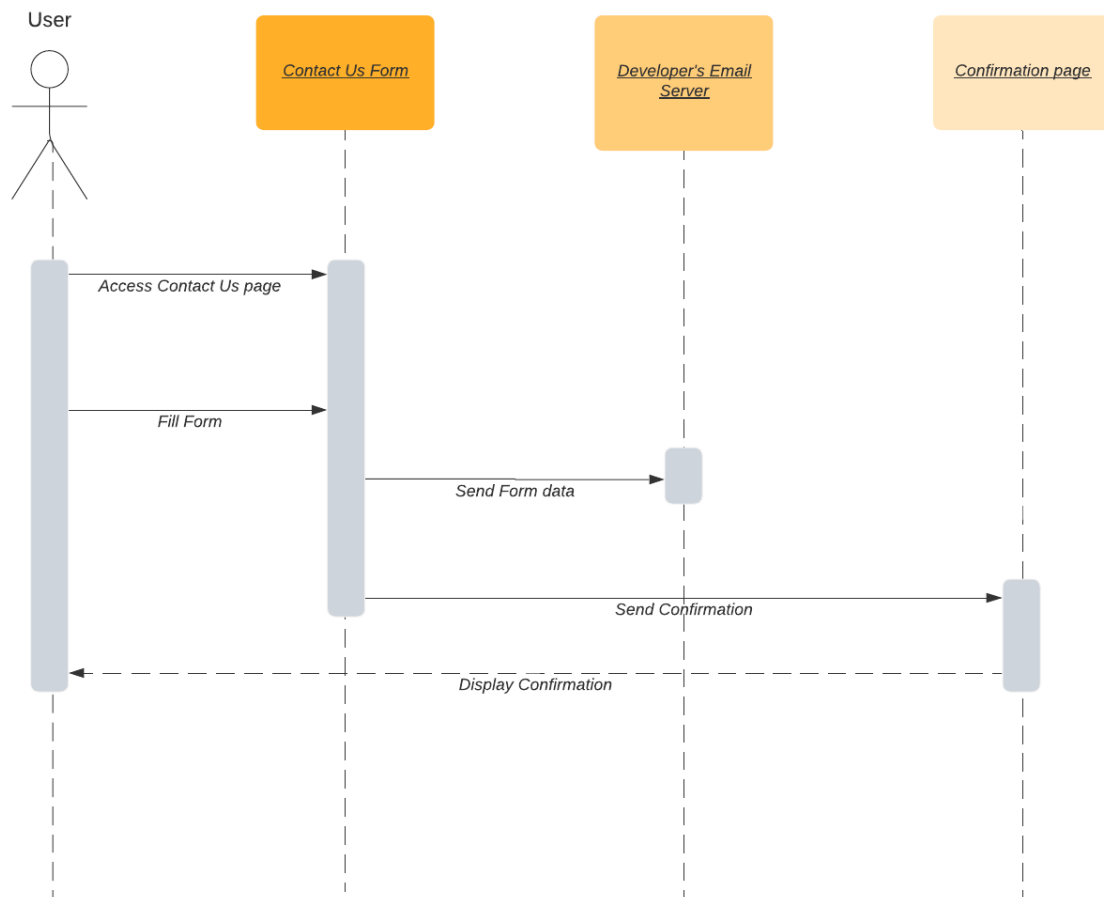
Sequence Diagrams: System Sequence Diagram



Sequence Diagram - Determination of Loan Eligibility & Amount

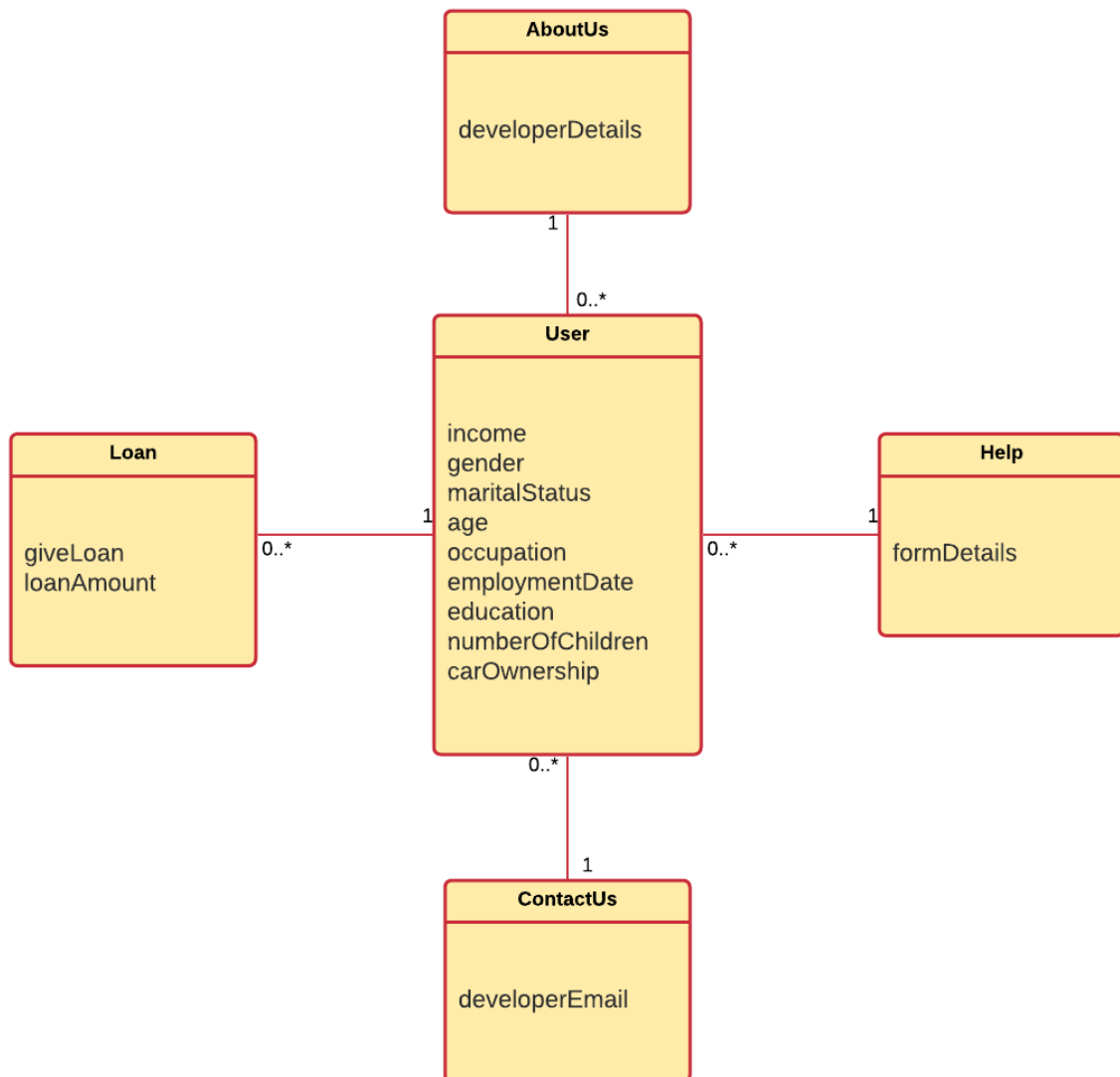


Sequence Diagram - Contact Us Page

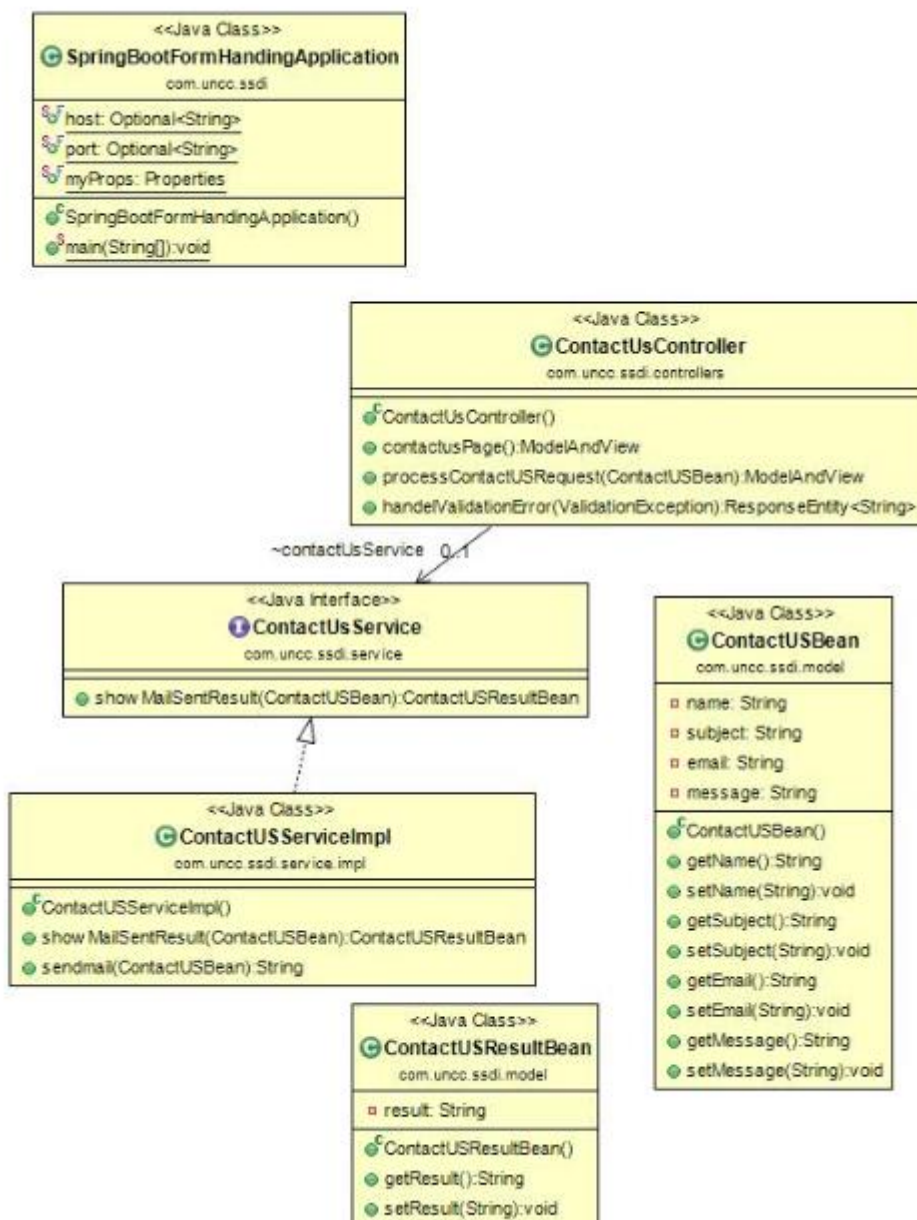


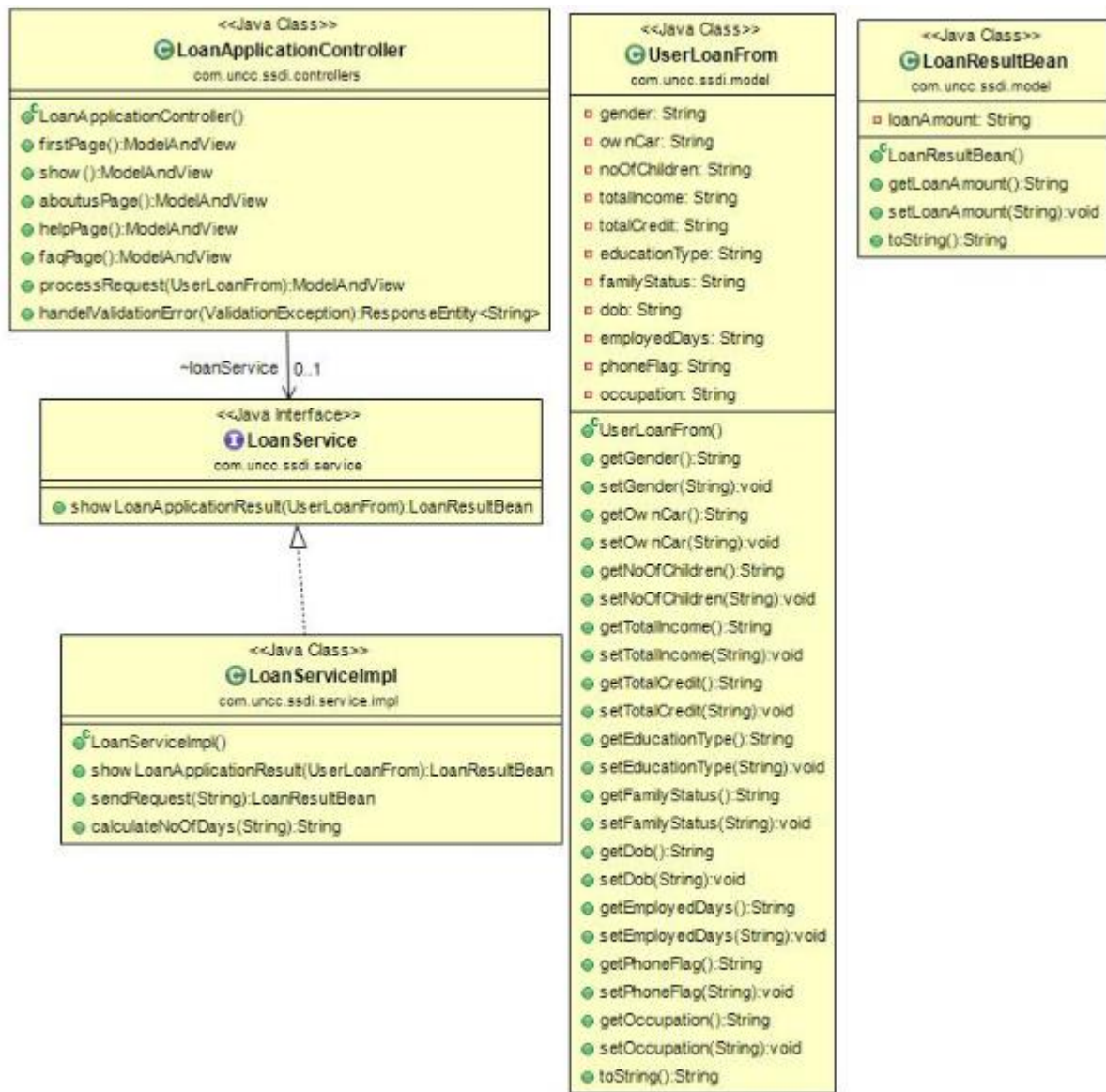
Class Diagrams

Domain Model Class Diagram

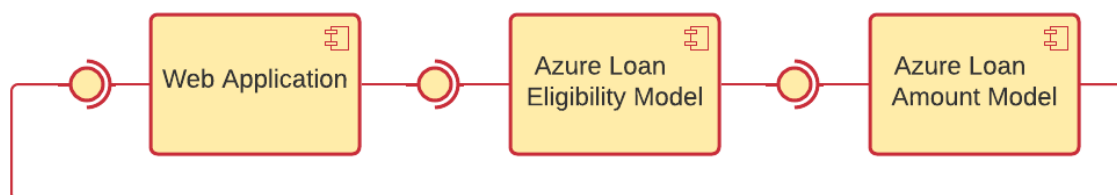


Design Class Diagrams

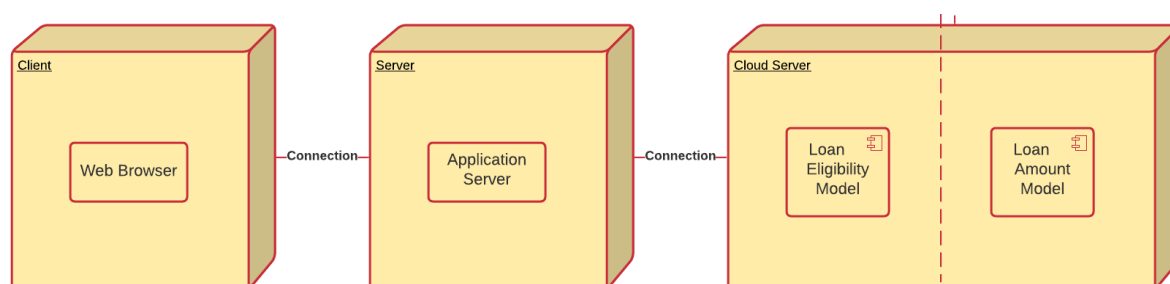




Component Diagram



Deployment Diagram



Loan Eligibility Form Outline

Loan Eligibility Form

Annual Income	<input style="width: 90%;" type="text" value="100,000"/>	Previous Loan Amount	<input style="width: 90%;" type="text" value="20,000"/>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> N.A.	Car Ownership	<input checked="" type="radio"/> Yes <input type="radio"/> No
No. of children	<input style="width: 30px;" type="text" value="4"/>	No. of Credit Bureau Enquiries	<input style="width: 30px;" type="text" value="4"/>
Marital Status	<input style="width: 100%;" type="text" value="Single"/>		
Occupation	<input style="width: 100%;" type="text" value="Accountant"/>		
Education	<input style="width: 100%;" type="text" value="Academic Degree"/>		
Birth date	<input style="width: 100px;" type="text" value="1/1/10"/>	Employment date	<input style="width: 100px;" type="text" value="1/1/10"/>
Credit Score 1	<input style="width: 100%;" type="range"/>		
Credit Score 2	<input style="width: 100%;" type="range"/>		

6. Testing

We created real time data mostly through UI

Use Case Descriptions:

1. User should have an option to type in his/her information to be sent to the model

Description of entity: Users will type in their financial and other information which will be used as an input for the Machine Learning model.

Inputs: Income, Gender, Marital status, Birth date, Occupation, Employment date, Education, number of children, Credit score 1, Credit score 2, Car ownership (Yes/No), number of Credit Bureau enquiries, Previous loan amount.

Precondition: None.

2. User should be able to view his/her loan eligibility

Description of entity: After submitting information, Users should be directed to a page which shows the output of the Machine Learning model.

Outputs: Users should be able to see whether they can get a loan and if so, the expected loan amount.

Preconditions: In order to view the outcome, a user should submit his information.

3. User should have an option to seek help

Description of entity: Users may have questions regarding the form. There should be a help page which should contain general instructions, a pre-filled form and Frequently asked questions to aid the user.

Outputs: A help page containing the above information.

Preconditions: None

4. User should have an option to contact us

Description of entity: Users should be able to contact us regarding any bug, queries or other matters.

Inputs: Name, email address, subject and message to be sent to us.

Preconditions: None

5. User should be able to obtain more information about us

Description of entity: There should be an about us page so users can obtain more information about the developers.

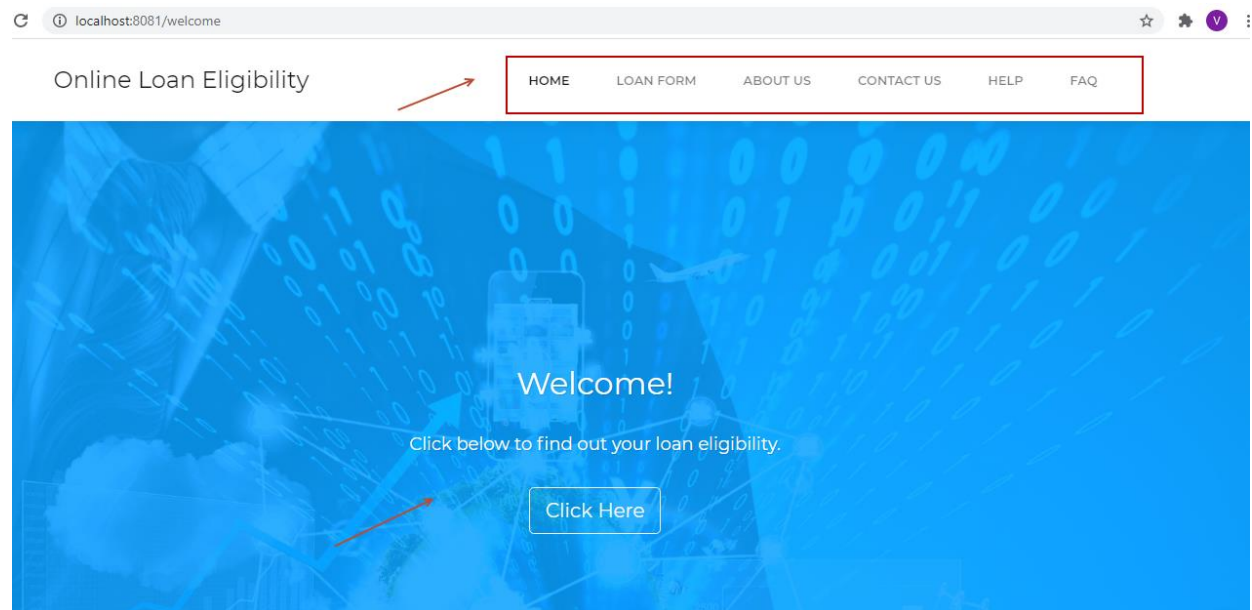
Output: Name and other details about the developers.

Preconditions: None

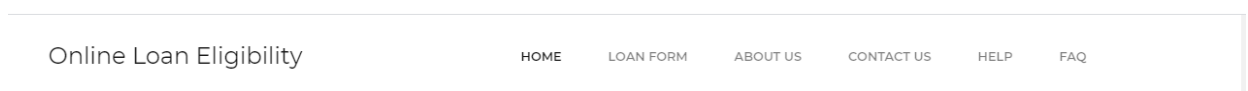
7. Step by Step Illustration of Functionality (With Screenshots)

Web Application View

Home Page



Loan Eligibility Form



localhost:8081/submitLoanApplication

Online Loan Eligibility

HOME LOAN FORM ABOUT US CONTACT US HELP FAQ

Loan Eligibility Form

Annual Income *

Previous Loan Amount (if any) *

Gender *

☒ Male ☐ Female

Car Ownership *

☒ Yes ☐ No

Having Landline Phone *

☒ Yes ☐ No

All Fields are mandatory.

Validation Added so that user can only enter Numeric values into numeric fields.

Online Loan Eligibility

HOME LOAN FORM ABOUT US CONTACT US HELP FAQ

Having Landline Phone *

☒ Yes ☐ No

Number of Kids in Household *

Marital Status *

Single / Unmarried

Education *

Academic Degree

Occupation *

Accountant

Birth Date * mm/dd/yyyy

Employment Date * mm/dd/yyyy

Submit

For Dates Standard Bootstrap date picker been used. Still need to implement validation check that user should not be able to pick future date for DOB and Employment, that one is pending.

localhost:8081/submitLoanApplication

Online Loan Eligibility

HOME LOAN FORM ABOUT US CONTACT US HELP

☒ Yes ☐ No

Number of Kids in Household *

2

Marital Status *

Single / Unmarried

Education *

Academic Degree

Occupation *

Accountant

Birth Date * 02/06/

Employment Date * mm/dd/yyyy

Submit

December 2020

Su	Mo	Tu	We	Th	Fr	Sa
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Today

Loan Eligibility Form- Result Page

localhost:8081/submitLoanApplication

Online Loan Eligibility

HOME LOAN FORM ABOUT US CONTACT US HELP FAQ

Result

You are eligible for a loan of up to

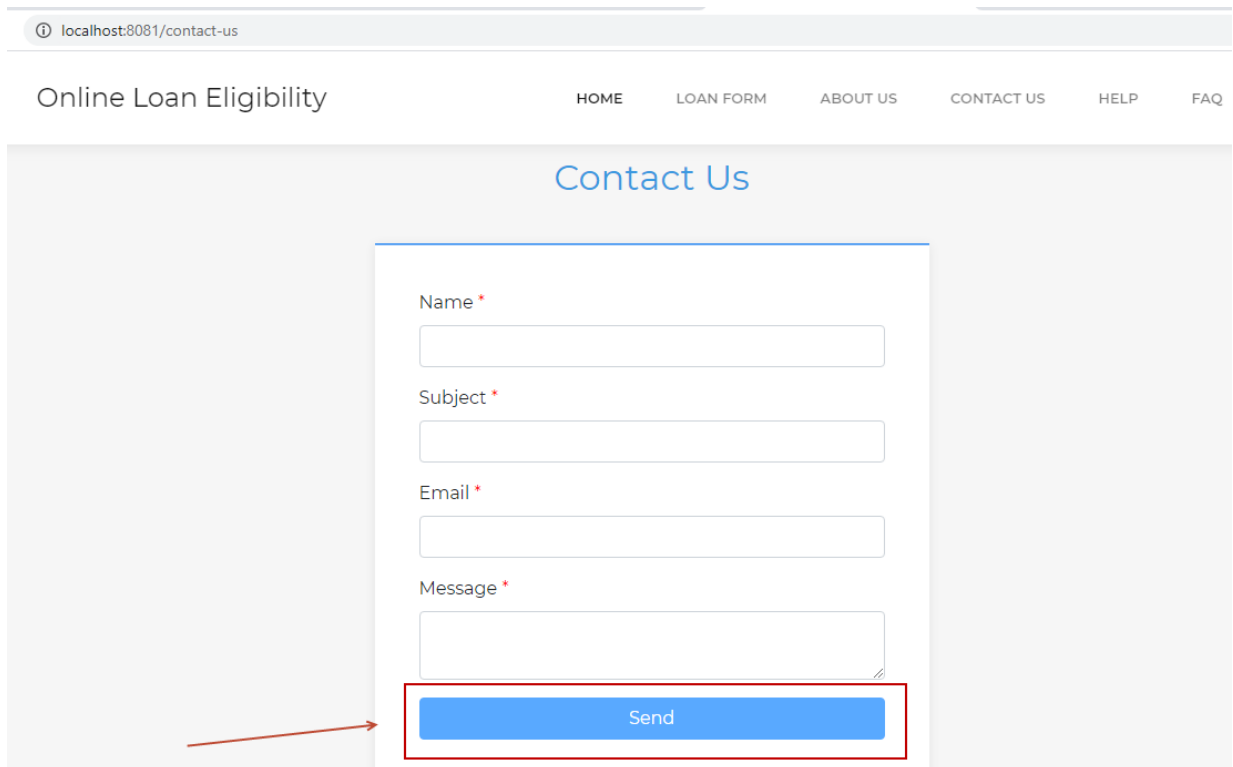
095658.05307717285

Disclaimer: The above result is based on our estimate. The actual outcome of your loan application may be different from the one shown above.

Â© 2020 Copyright Text

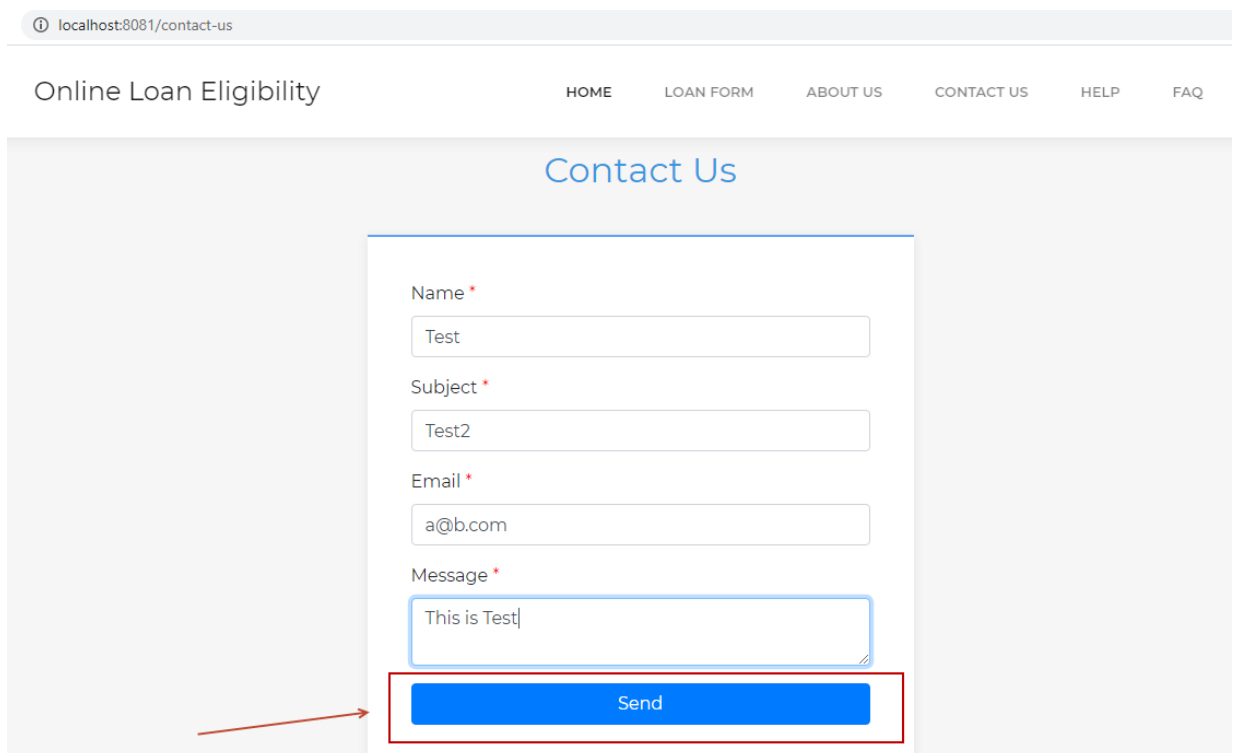
Contact Us

Mandatory Field Check – Send Button - Disabled when fields are blank



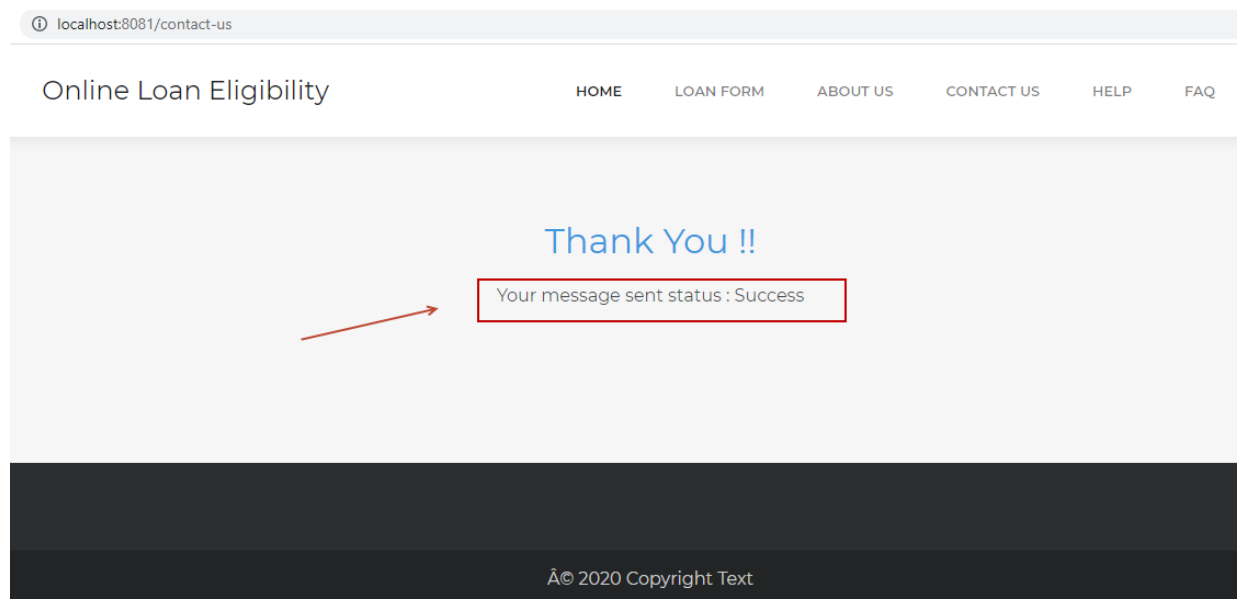
The screenshot shows a web browser at `localhost:8081/contact-us`. The page title is "Online Loan Eligibility". The navigation menu includes HOME, LOAN FORM, ABOUT US, CONTACT US, HELP, and FAQ. The main heading is "Contact Us". The form contains four fields: Name *, Subject *, Email *, and Message *. The Send button is disabled (greyed out) and is highlighted with a red box. A red arrow points to the button.

When filled with required fields - Button enabled.

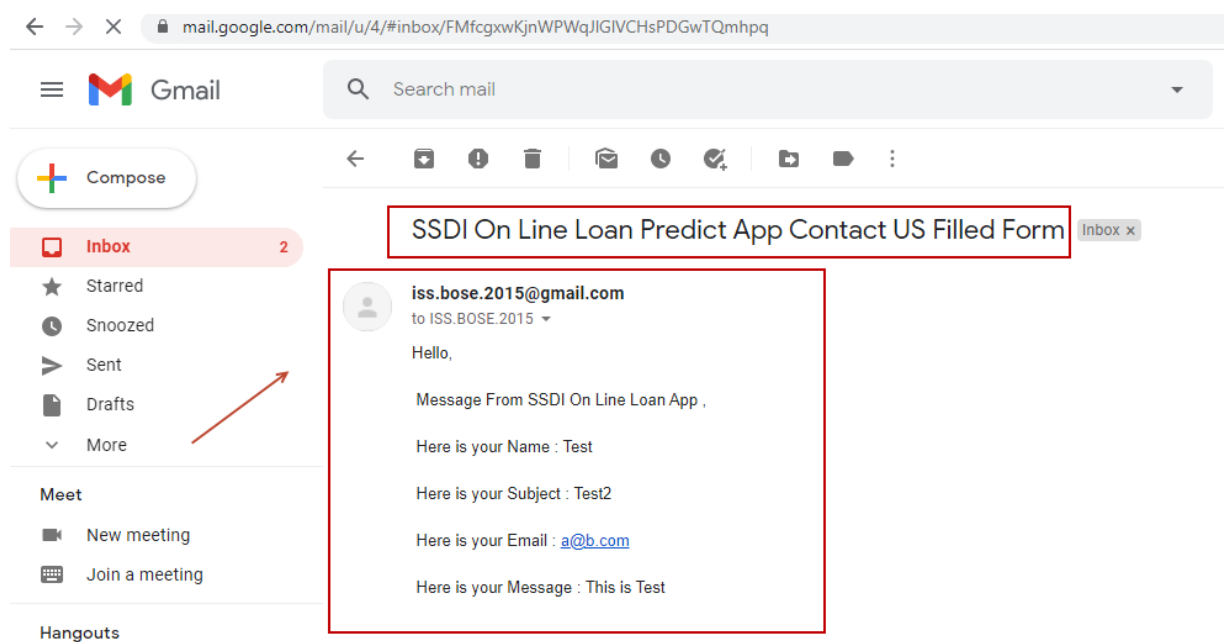


The screenshot shows the same web browser at `localhost:8081/contact-us`. The form fields are now filled: Name * (Test), Subject * (Test2), Email * (a@b.com), and Message * (This is Test). The Send button is now enabled (blue) and is highlighted with a red box. A red arrow points to the button.

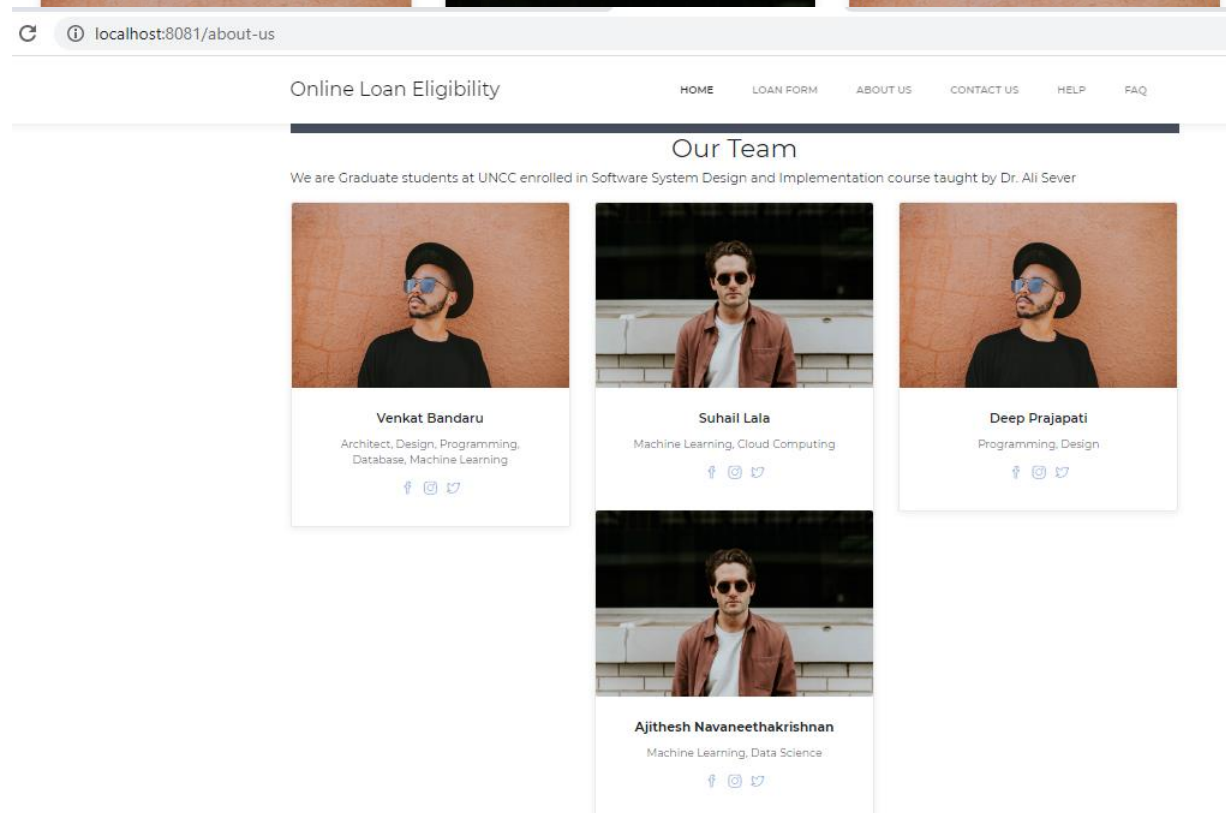
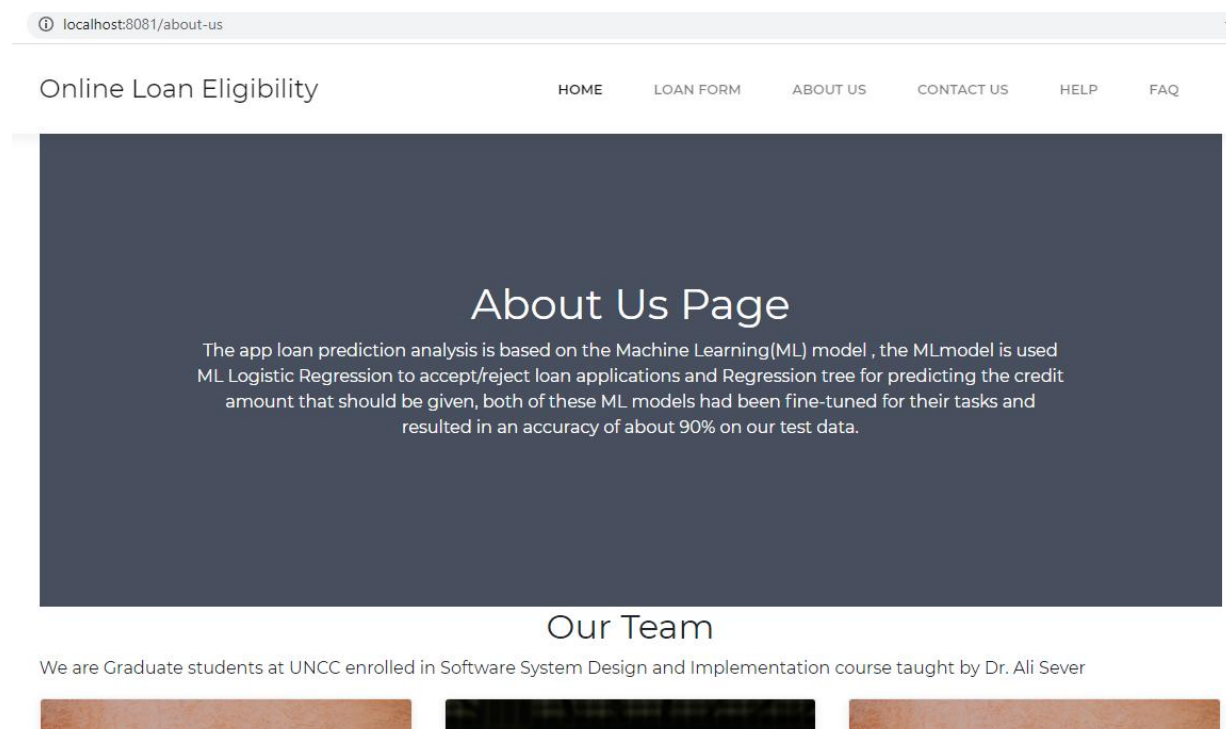
Contact Us – Success Page (mail sent)

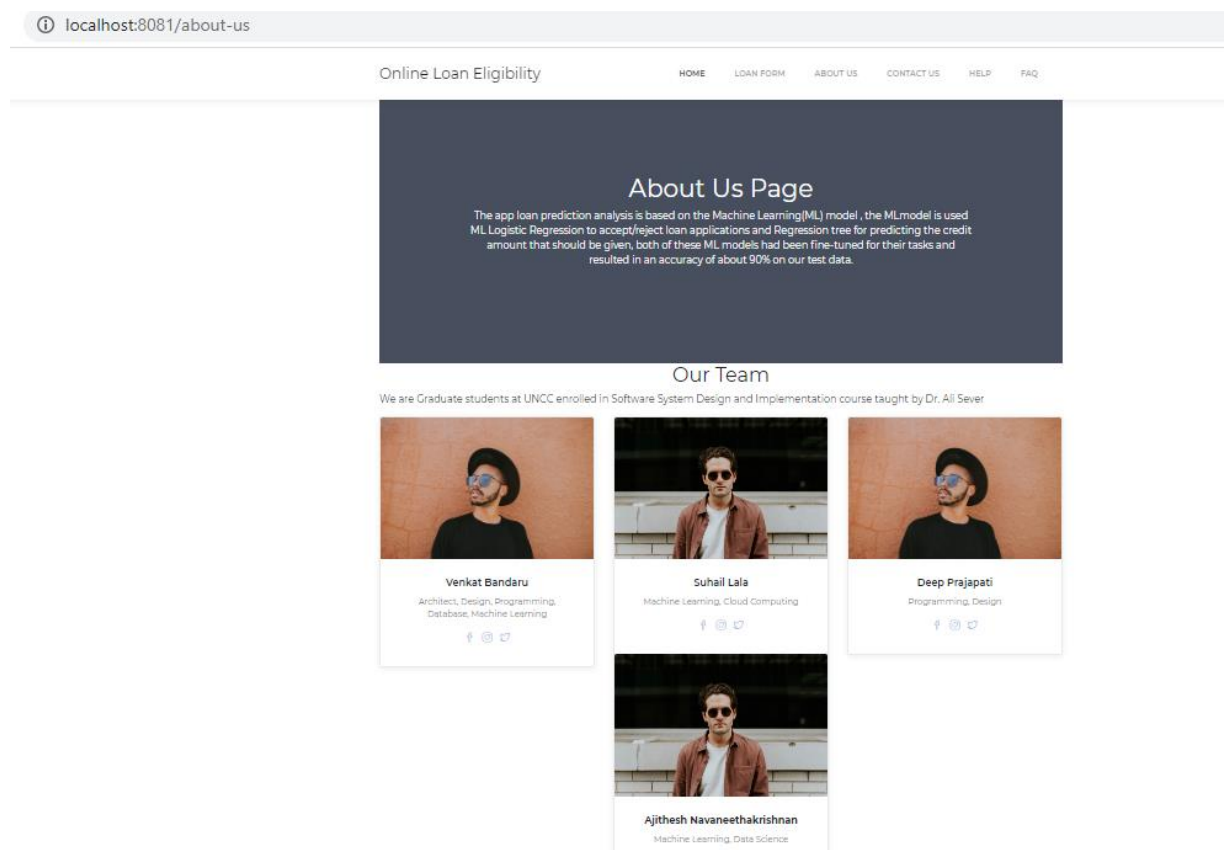


Contact Us – Success Page (mail received)



About Us Page





Faq Page

Online Loan Eligibility
HOME
LOAN FORM
ABOUT US
CONTACT US
HELP
FAQ

FAQ

Approved Loan Page

If your loan is approved, A similar page will be generated with the information about how much loan is approved based on information provided by you in the previous Loan Application Form

Frequently asked Questions!

Will my loan get approved? - We cannot guarantee you on whether you will get approved or not, its purely based on the information you provide

When will I receive the result? - It is as instant you can get.

Who should I contact regarding more information about the process? - You can directly contact us using the contact information provided below.

What to do if i provide information Incorrectly? - Send a quick email to us and we will follow up.

Application irresponsive or showing error? - Send a quick email to us with screenshot and our tech team will follow up.

Require more information? - you can either try to fill the form by yourself or send us an email and we will follow up with step by step instructions.

Other errors? - Try Refreshing the page and you are good to go.

Â© 2020 Copyright Text UNCC

Help Page

localhost:8081/help

Online Loan Eligibility

HOMELOAN FORMABOUT USCONTACTUSHELPFQAQ

Help

Stuck? All the information available here!

Guide to fill the Loan Application Form

A typical Loan Application form requires that that all the fields to be properly filled with the information as depicted above.

Our system then takes all the necessary information provided by you for processing and provides result on whether you are approved or not.

Sample Loan Eligibility Form

Annual Income Average dollar earnings per annum - Must be integer

100000

Previous Loan Amount (if any) Dollar amount of previous loan, 0 for first-time loan seekers - Must be integer

15000

Gender

☒ Male

☐ Female

Car Ownership Current car ownership. Joint-owners should select 'Yes'.

☒ Yes ☐ No

Number of Kids in Household Children below 18 residing in the house - Must be Integer

2

Marital Status

Single / Unmarried

localhost:8081/help

Online Loan Eligibility

HOMELOAN FORMABOUT USCONTACTUSHELPFQAQ

Car Ownership Current car ownership. Joint-owners should select 'Yes'.

☒ Yes ☐ No

Number of Kids in Household Children below 18 residing in the house - Must be Integer

2

Marital Status

Single / Unmarried

EducationChoose 'Academic Degree' if you are holding Bachelor's or higher. For Associate's, choose 'Higher Education'.

Academic Degree

Occupation Choose 'Others' if your occupation is not listed

Accountant

Birth Date Date of Birth as per government records

mm/dd/yyyy

Employment Date Date of joining current employment as per Tax returns. Students should select current date.

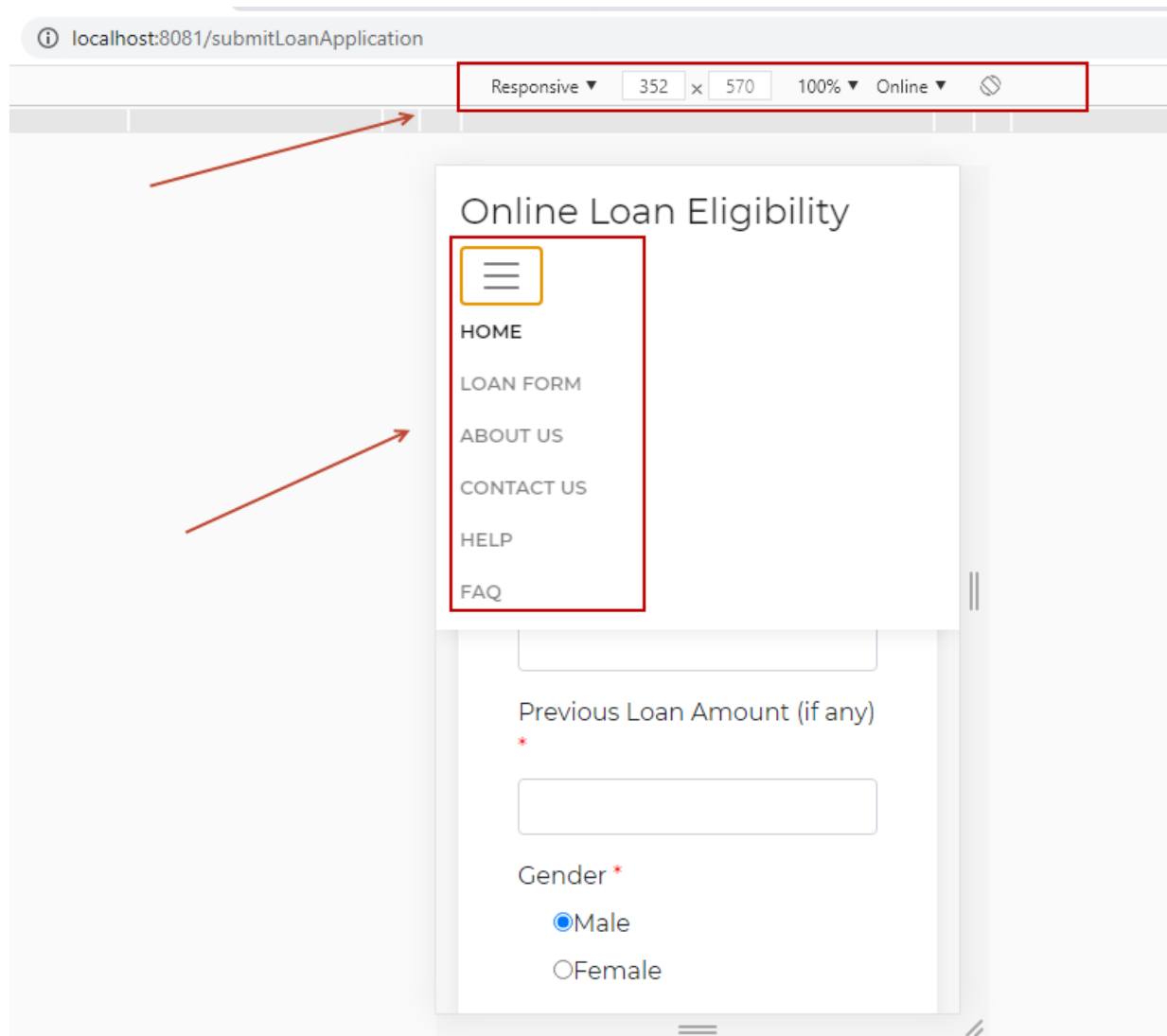
mm/dd/yyyy

Press 'Submit' once you are done with the form

Submit

© 2020 Copyright Text UNCC

Responsive Mobile Browser Application View



localhost:8081/contact-us

Responsive ▾ 352 x 570 100% ▾ Online ▾

Online Loan Eligibility

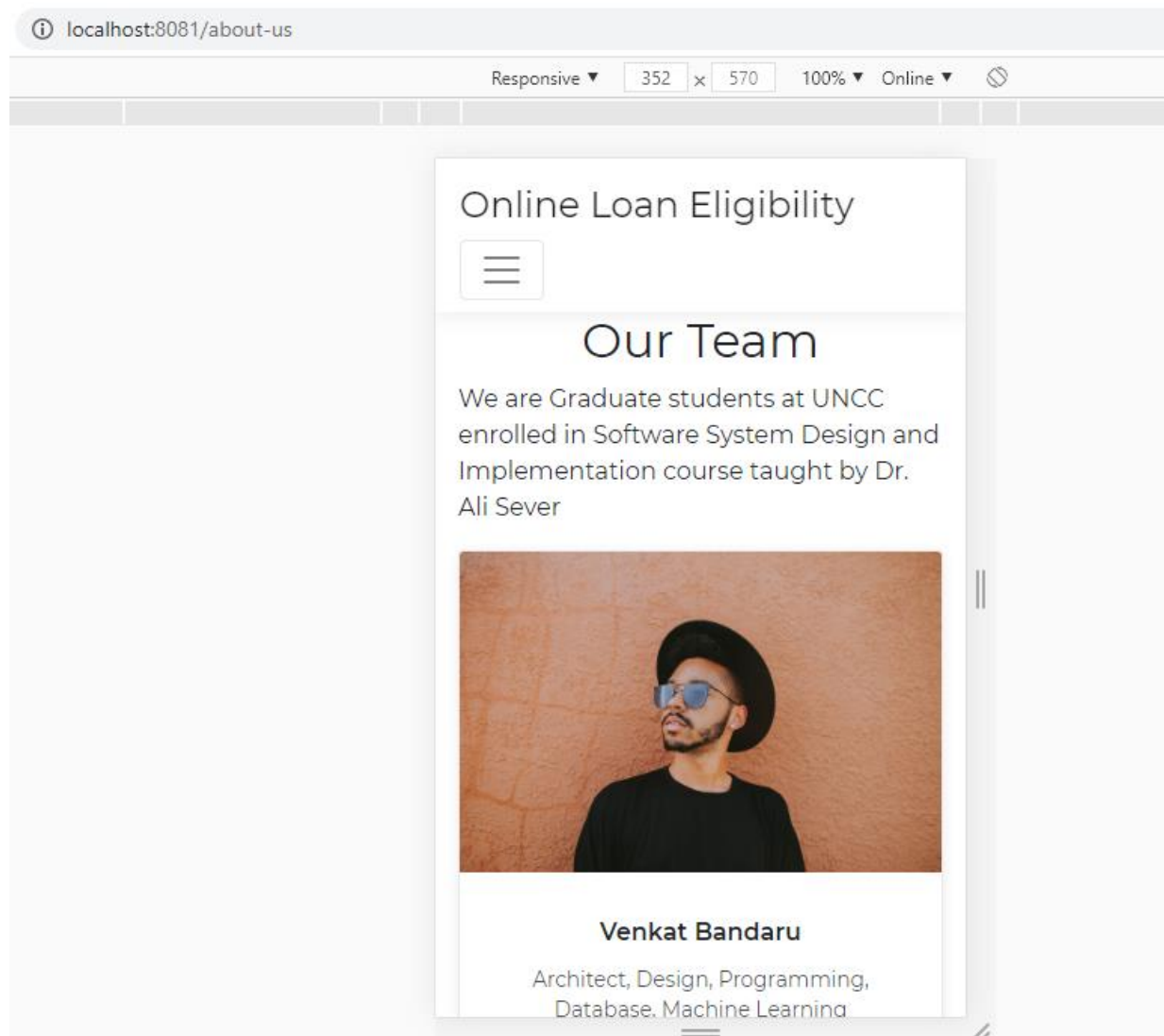
Contact Us

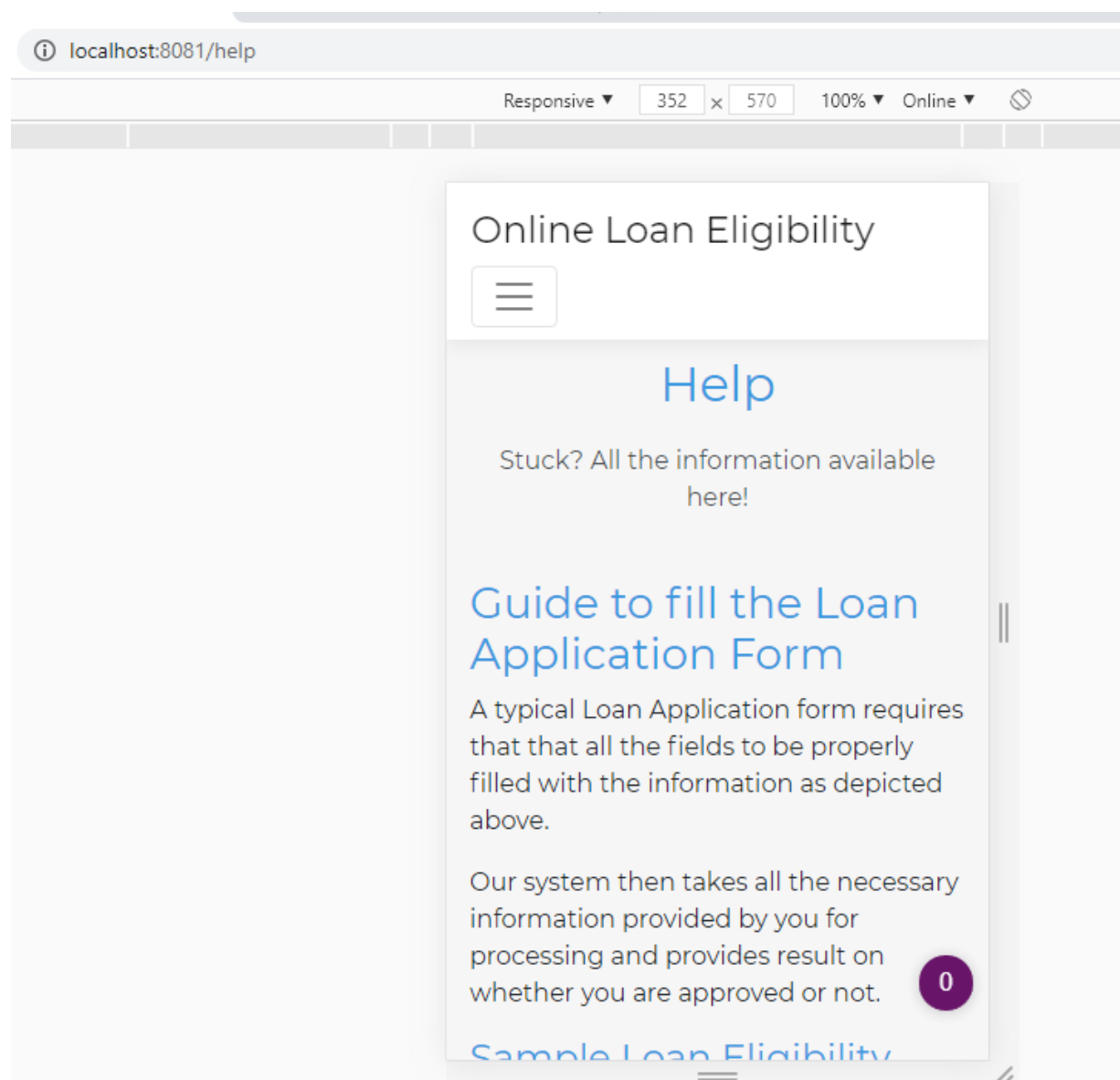
Name *

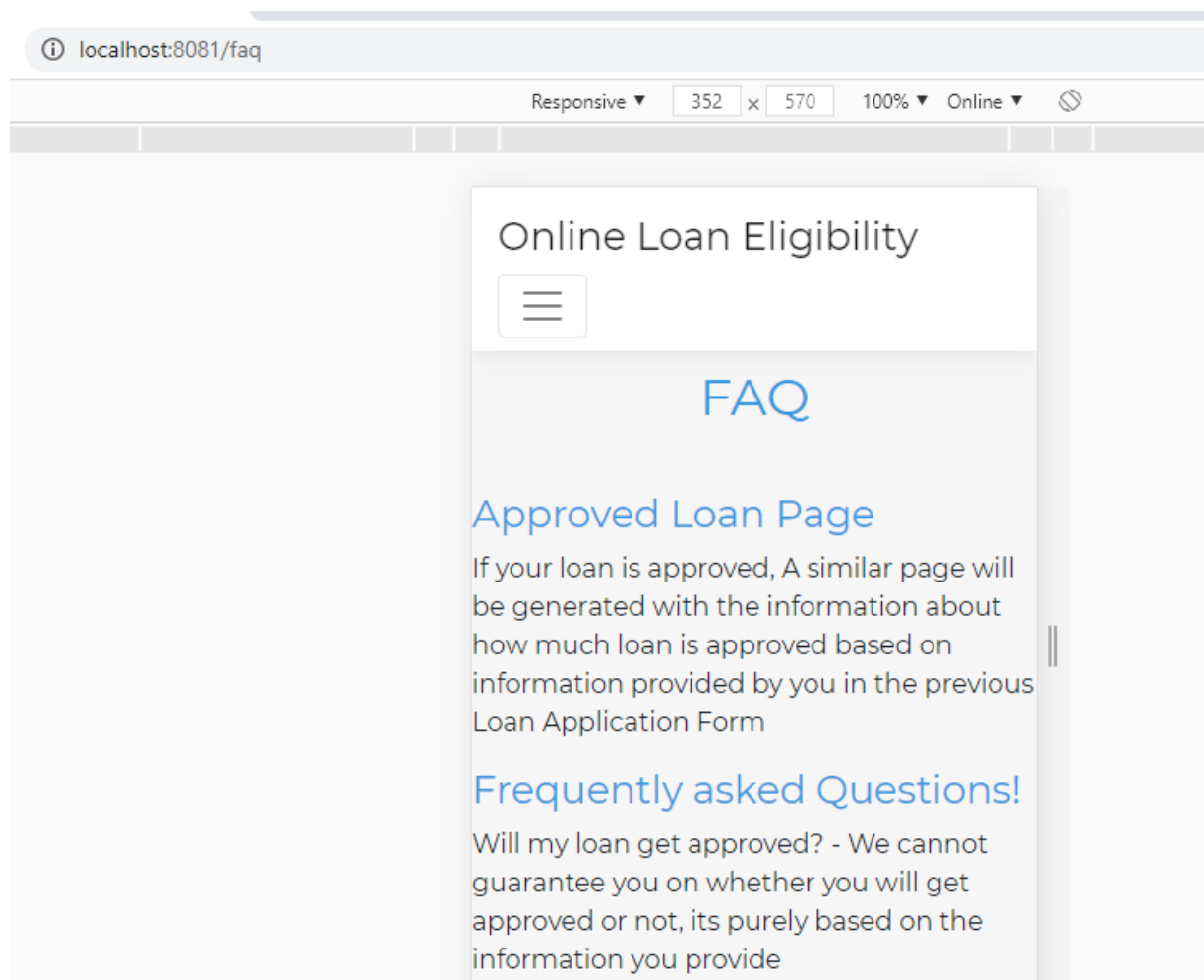
Subject *

Email *

Message *







8. Results and Project Conclusion

We successfully developed the web application. However, along the way we did suffer some setbacks. For instance, our original plan was to deploy the models on AWS. However, due to the restrictions on our AWS account, we had to switch to Azure. Secondly, due to the paucity of time, we had to push the less-essential features to a future release to focus more on the crucial features.

In future releases, we may implement a registration and login system where the user can register an account if he so wishes. For a registered user, the system can remember his form details. We may also implement a page where an Administrator can create, read, update, and delete users from the system.

9. GIT Hub URL – Package Explorer

<https://github.com/ssdiloan/online-loan-eligible>

This project is based on Spring Boot web application.

Tools used:

1. Spring Boot 1.5.16.RELEASE
2. Spring 4.3.7.RELEASE
3. Thymeleaf
4. Maven
5. Embedded Tomcat 8.5.6

How to run this repo :

Right click on project and run as Spring Boot App.

open your browser :

`http://localhost:8081`

