Dr. N.G. P. INSTITUTE OF TECHNOLOGY

An Autonomous Institution

(Approved by AICTE, New Delhi &Affiliated to Anna University of Technology-Chennai)

Recognized by UGC & Accredited by NACC with A+ and NBA (CSE, BME, ECE, EEE and Mech)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Project Submitted for

NEW LEAF

under the title of

HOPE HAVEN USING HTML, CSS, Java Script, MySQL

Submitted to

Mr. PUGAZH MARAN

Submitted by

THARUNVETRIVELAN S

SHUHAIB S

SARAN S B B

HOPE HAVEN – PROJECT REPORT

Introduction

Our project, titled "Hope Haven," is a web application designed to facilitate interactions between donors and an orphanage. The primary goal of the project is to create a platform that streamlines the process of making genuine donations, handling adoption appointments, and providing an all-in-one account for users interested in contributing to various aspects of the orphanage's welfare.

In this report, we will delve into the development and integration of both frontend and backend components. The frontend development involves creating an interactive and user-friendly interface using HTML, CSS, and JavaScript. The dynamic nature of the frontend is complemented by the backend, which utilizes Java Servlets to handle HTTP requests and interacts with a MySQL database for secure and efficient data storage. The project is hosted on a local XAMPP server, providing a robust environment for testing and development.

Our focus on database connectivity ensures the secure storage of user data, allowing for efficient retrieval and management. The report will also cover the challenges faced during development, the testing and debugging methodologies employed, and insights into the integration and deployment processes.

This project not only serves as a practical application of web development skills but also addresses the real-world need for effective online platforms connecting donors with orphanages. As we explore each aspect of the project in this report, we aim to provide a comprehensive understanding of the development process and the functionalities implemented.

HTML

1. About/Home Page

The About/Home Page serves as the entry point for users, providing crucial information about the project. The structure of this page is built with HTML5, ensuring compatibility with modern browsers. Key features include:

- **Navigation Bar:** Utilizing Bootstrap components, the navigation bar ensures a responsive and user-friendly interface.
- **Slider Section:** The carousel slider is implemented using the Owl Carousel library, enhancing the visual appeal of the page. The slider incorporates multiple slides with distinct background images and content.
- **Objective Section:** Highlighting the project's objectives, this section uses a grid layout to showcase key features, such as genuine necessity donation, adoption processes, healthy food donation, and secure data management.
- **Objective Section:** Articulating the project's objectives, this section emphasizes the commitment to creating compassionate sanctuaries for vulnerable children. The content is structured with well-defined HTML elements.
- **Developers Section:** Featuring a list of developers involved in the project, the Developers Section incorporates images, names, and social media links for each developer. The structure is responsive and adheres to modern design principles.
- **Contact Section:** The Contact Section incorporates a call-to-action button for user engagement. The design encourages users to navigate to the dedicated contact page.

- **Info Section:** Providing essential information such as location, phone number, and email, the Info Section enhances user accessibility.
- **Footer Section:** The footer includes copyright information and acknowledges the designer, Tharun Velan.

2. CONTACT US

Overview:

The "**Contact Us**" page serves as a vital bridge between our organization and the users, facilitating seamless communication and interaction. Through this page, visitors can reach out to us, share their inquiries, and connect with our team. The design of the page is crafted to be user-friendly, ensuring a smooth experience for anyone seeking to engage with Hope Haven.

Features:

1. **Contact Form:**
    - The page features a well-structured contact form that allows users to provide essential details such as their name, email address, phone number, and a message.
    - Input validation ensures that users submit accurate and complete information, streamlining the communication process.
2. **Visual Appeal:**
    - A visually engaging layout, coupled with relevant images, creates an inviting atmosphere for visitors.
    - The carefully selected images aim to evoke a sense of warmth and trust, aligning with the compassionate nature of our organization.
3. **Responsive Design:**
    - The page is designed to be responsive across various devices, ensuring a consistent and enjoyable experience for users accessing it from desktops, tablets, or mobile phones.

Functionality:

1. **Email Integration:**
    - The contact form is integrated with email services, allowing users' messages to be promptly delivered to our designated email address.
    - Email.js and SMTP.js are utilized to facilitate the seamless transmission of messages from the user to our communication platform.
2. **Confirmation and Alerts:**
    - Users receive confirmation messages upon successful submission of the contact form.
    - In case of any issues or errors during submission, users are alerted with informative messages for a better user experience.

Call to Action:

The "**Contact Us**" page encourages users to initiate communication, fostering an open channel for inquiries, partnerships, or any other form of engagement. The commitment to prompt and meaningful communication reflects our dedication to transparency and user-centric interaction.

For any queries or assistance, feel free to use the contact form, and our team will respond at the earliest convenience.

Great! Let's generate a write-up for the "Join" and "Signup" pages:

3. JOIN - REDIRECTING PAGE

Overview:

The "**Join**" page acts as a pivotal step for individuals ready to become a part of our Hope Haven family. This page is strategically positioned to seamlessly redirect users after expressing their interest through the "About" page. The dynamic visual elements and user-friendly interface enhance the overall onboarding experience.

Features:

1. **Background Video:**
   - The page features a captivating background video, providing an immersive visual experience.
   - The video content sets an inviting tone, fostering a sense of anticipation for those considering joining Hope Haven.
2. **Call to Action Buttons:**
   - Two prominent buttons, "LOGIN" and "SIGN UP," are strategically placed, offering users clear and distinct pathways based on their status—whether returning members or new enthusiasts.
   - The button design ensures visibility and encourages user interaction.

Visual Appeal:

1. **Font and Color Scheme:**
   - Consistent with the overall theme, the font size and color scheme maintain continuity with other pages, ensuring a harmonious visual experience.
   - The choice of white text against a dark background enhances readability and complements the immersive video backdrop.
2. **Button Animation:**
   - The buttons exhibit a subtle animation on hover, providing visual feedback to users and making the interface more engaging.

Functionality:

1. **Responsive Design:**
   - The page is designed to be responsive across various devices, guaranteeing a seamless experience for users on desktops, tablets, or mobile phones.

  - Responsiveness ensures that the visual elements and interactive features adapt appropriately to different screen sizes.

2. **Redirect Links:**

   - The "LOGIN" and "SIGN UP" buttons are linked to corresponding pages, guiding users to the next step in their journey with Hope Haven.

Login Page

The **Login Page** serves as the gateway for existing users to access their accounts within the Hope Haven community. The design focuses on creating a seamless and visually engaging experience for users. The page incorporates a modern glassmorphism style, providing a sleek and transparent appearance.

Key Features:

1. **Background Video:** The page features a background video that adds a dynamic and immersive element. The video is set to autoplay, creating a visually appealing backdrop.
2. **Form Design:** The login form is designed for simplicity and functionality. It includes fields for the user's username (email or phone) and password. The use of the Poppins font enhances readability.
3. **Social Media Integration:** Users can log in using their Google or Facebook accounts through dedicated buttons. This integration simplifies the login process for users who prefer using their existing social media credentials.
4. **Responsive Design:** The design is responsive, ensuring a seamless experience across various devices, including desktops, tablets, and mobile phones.

Aesthetic Elements: The color scheme, primarily consisting of dark backgrounds and vibrant accent colors, creates a visually appealing contrast. The glassmorphism effect on the form elements adds a modern and stylish touch to the overall design.

Sign Up Page

The **Sign Up Page** is designed for new users who wish to join the Hope Haven community. The page offers a user-friendly and secure registration process, emphasizing a clean and inviting design.

Key Features:

1. **Background Image:** A background image with a warm and welcoming atmosphere sets the tone for the Sign Up page. The image reflects the spirit of Hope Haven and the community's commitment to providing a nurturing environment.
2. **Form Fields:** The registration form includes fields for the user's full name, email, phone number, date of birth, username, password, and address. These fields capture essential information for creating a personalized and secure user account.
3. **Additional Information:** To enhance user engagement, the form includes a field for users to express their interest in adoption, allowing Hope Haven to tailor its services to individual preferences.
4. **Call-to-Action Button:** The "Send" button serves as the call-to-action, prompting users to submit their registration details. The button's design is consistent with the overall aesthetic, maintaining a cohesive and visually pleasing layout.

Responsive Design: Similar to the Login Page, the Sign Up Page is designed to be responsive across different devices. This ensures that users can easily complete the registration process on their preferred platform.

Aesthetic Elements: The background image, combined with a thoughtfully chosen color palette, creates a harmonious and inviting visual experience. The design elements align with Hope Haven's brand identity, conveying a sense of care and community.

These pages collectively contribute to an inclusive and welcoming online environment for users interacting with Hope Haven's platform. The thoughtful integration of design elements and functionality aims to provide a positive and user-centric experience.

CSS

1. Bootstrap CSS Explanation:

1. Linking Bootstrap:

- Both the login and signup pages link to the Bootstrap CSS framework through **<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />**.
- This inclusion provides a foundation for responsive design and styling elements.

2. Grid System:

- Bootstrap's grid system is employed to structure the layout of the pages, ensuring responsiveness.
- Classes like **container** and **col-md-8** help organize content effectively across different screen sizes.

3. Button Styling:

- Bootstrap styles are applied to buttons using the **btn** class, providing a consistent and visually appealing look.
- The framework's predefined classes contribute to responsive button behavior.

4. Navbar Component:

- The navigation bar (**navbar**) is styled with Bootstrap classes, offering a clean and responsive design.
- The integration of the **navbar-toggler** ensures proper navigation on smaller screens.

5. Form Styling:

- Form elements, including inputs and buttons, benefit from Bootstrap's styling for a polished appearance.
- The framework enhances form components for better user experience and visual consistency.

2. Font CSS Explanation:

1. Linking Google Fonts:

- Both pages incorporate Google Fonts with **<link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700|Poppins:400,500,700&display=swap" rel="stylesheet" />**.

- This link introduces the Open Sans and Poppins font families for improved typography.

2. Font Usage in Headings and Text:

- The specified Google Fonts are applied to headings (**h1**, **h2**, **h3**) and other textual elements for a modern and readable display.

3. Font Weights:

- Different font weights, such as **400**, **500**, and **700**, are employed to create text hierarchy and emphasis.
- Varied weights contribute to a structured and visually appealing content layout.

4. Font Color:

- Customized font colors are implemented to enhance readability and align with the overall design theme.
- Consistent color choices ensure a unified and harmonious visual identity.

5. Responsive Typography:

- Google Fonts, in combination with Bootstrap, facilitates responsive typography that adapts well to diverse screen sizes.
- This ensures an optimal reading experience across various devices.

3. Styles Explanation:

1. General Styling:

- The CSS styles begin with general settings for the body, setting the font family, text color, and background color. The body has a white background, ensuring a clean and readable layout.

2. Button Styling:

- Buttons with the class **.button** are styled to have a font size of 30px, substantial padding, and a light slate grey background. Hover effects include a smooth transition of background color and box-shadow, enhancing user interaction.

3. Layout Padding:

- Classes like **.layout_padding**, **.layout_padding2**, **.layout_padding2-top**, **.layout_padding2-bottom**, **.layout_padding-top**, and **.layout_padding-bottom** provide consistent padding for different sections, maintaining spacing and visual balance.

4. Heading Container:

- The heading container includes styles for centering text and creating an underline effect below the heading using a pseudo-element **::before** with a distinctive color.

5. Hero Area and Navbar Styling:

- The hero area utilizes a background image, and the navbar is designed with a transparent background. Navbar links have a unique styling with an underline effect on hover, adding a visual cue for navigation.

6. Slider Section Styling:

- The slider section features a responsive carousel with indicators. The text within the slider is styled with uppercase, bold fonts, and the CTA button has a transparent background with a border, transitioning to a solid background on hover.

7. Section Background Styling:

- Sections like **.us_section**, **.heathy_section**, **.trainer_section**, and **.contact_section** have background images, creating visually appealing backgrounds. Text and button styles are adjusted for readability and consistency.

8. Form and Input Styling:

- The contact form elements have a clean design with white background inputs, placeholders, and a contrasting submit button with a smooth transition on hover.

9. Info Items Styling:

- The info section displays items with icons and text. Each item has a distinct icon, and the text is centered below. The icons have a circular background, enhancing their visual appeal.

10. Footer Styling:

- The footer has a light background with centered text. Links have a darker color, ensuring visibility. Overall, the styles create a cohesive and visually pleasing website layout.

DEMOSERVLET

1. Class Definition:
    - The class extends HttpServlet.
    - It overrides the doGet and doPost methods to handle GET and POST requests, respectively.
2. Constructor:
    - The default constructor (DemoServlet()) is present.
3. doGet Method:
    - Handles HTTP GET requests.
    - Writes a response indicating that the servlet has been served at the specified context path.
4. doPost Method:
    - Handles HTTP POST requests.
    - Retrieves parameters (name, phone, email, message) from the request.
    - Creates a DataBase object with the provided parameters.
    - Calls the store method of the DataBase object to store the data.
    - Invokes the doGet method to send the response.
5. Data Storage:

- It seems that you are using a DataBase class to store the data. However, the DataBase class and its methods are not provided in the code snippet. I assume it's a separate class responsible for handling data storage.

6. SerialVersionUID:

   - serialVersionUID is a version number used during deserialization to verify that the sender and receiver of a serialized object have loaded classes for that object that are compatible with respect to serialization.

7. Comments:

   - Some comments are present in the code, providing a bit of context.

DATABASE.JAVA

1. Database Connection Information:

   - dbUrl: The URL of the MySQL database, specifying the host, port, and database name.
   - dbUname: The database username.
   - dbPassword: The database password.
   - dbDriver: The JDBC driver for MySQL.

2. Instance Variables:

   - name, phone, email, message: Instance variables to store the data that will be inserted into the database.

3. Constructor:

   - The constructor initializes the instance variables with the provided values.

4. store Method:

   - Connects to the MySQL database using the specified URL, username, and password.
   - Uses a prepared statement to execute an SQL INSERT query, inserting the provided data into the contact_sec_table.
   - The ? placeholders in the SQL query are replaced with the values of the instance variables using setString.
   - The executeUpdate method is called to execute the SQL query.

5. Exception Handling:

   - The method includes a try block to catch any exceptions that might occur during the database operation.
   - If an exception occurs, the stack trace is printed.

6. JDBC Driver Loading:

   - The Class.forName(dbDriver) statement is used to dynamically load the MySQL JDBC driver. However, note that in modern JDBC versions, this is not strictly necessary, as the driver can be loaded automatically.

Please ensure that you have the MySQL JDBC driver in your classpath. Also, make sure your MySQL server is running and accessible at the specified URL.

Remember to handle database connections and operations carefully in production code, including proper closing of resources (Connection, PreparedStatement) in a finally block or using try-with-resources to ensure resource cleanup.

Integration of Frontend and Backend Components:

In bringing my web application to life, I meticulously wove together the frontend and backend components. Crafting an engaging user interface using HTML, CSS, and JavaScript, I orchestrated seamless interactions with the server-side logic implemented in Java Servlets. Leveraging the power of AJAX requests, user inputs seamlessly traveled to the backend, creating a harmonious symphony between the user interface and server-side functionalities.

Deployment Process (XAMPP Server):

For the heartbeat of my application, I chose XAMPP—a reliable companion that bundled Apache, MySQL, PHP, and Perl. The deployment dance involved placing my web files in the **htdocs** directory, configuring Apache, and orchestrating the perfect setup through the **httpd.conf** file. With a few clicks in the XAMPP control panel, the Apache server and databases gracefully came to life, making my application accessible at the elegant **http://localhost/your-app**.

Overview of Hosting Solutions:

As my creation outgrew the confines of local development, I cast an eye toward broader horizons. Cloud hosting platforms like AWS, Azure, and Google Cloud beckoned, offering scalable solutions. The allure of Platform as a Service (PaaS) providers such as Heroku, Netlify, and Vercel simplified the deployment narrative, freeing me from the intricacies of server management. Securing a domain name and embracing SSL certificates added a touch of professionalism to my online presence.

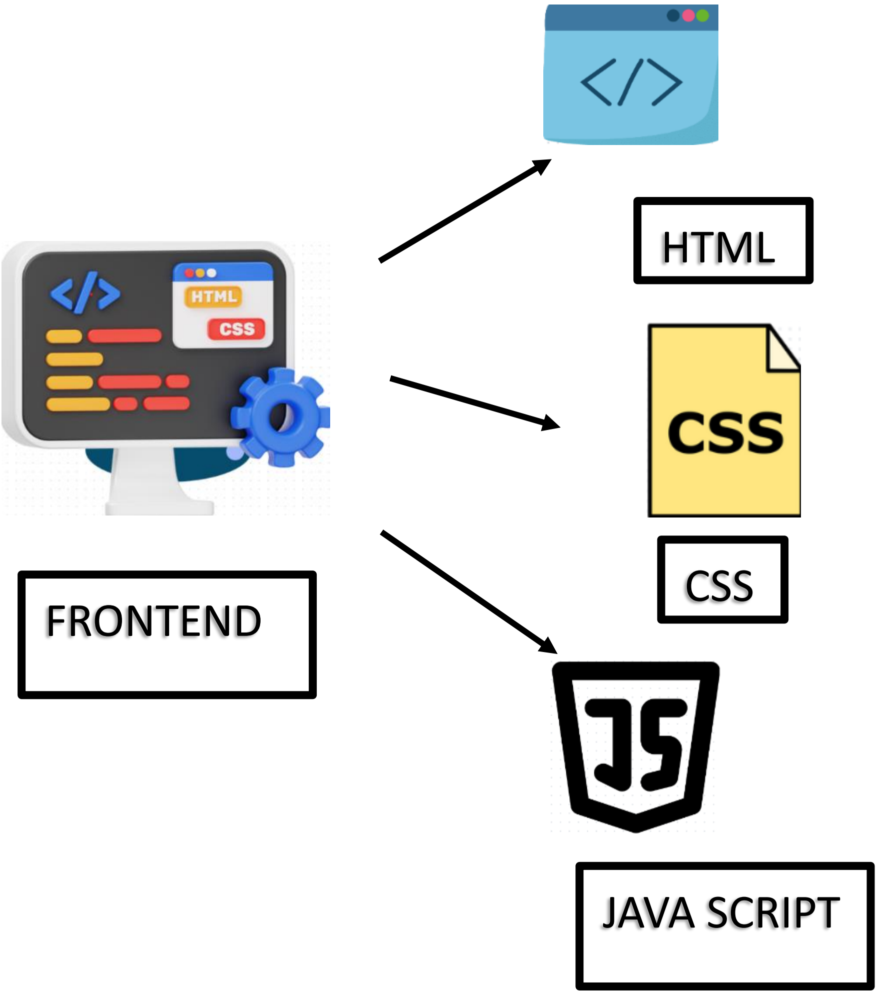
Challenges Faced:

In this journey of creation, challenges emerged like characters in a compelling story. Configuring servers and databases became a puzzle to solve, each piece contributing to the larger narrative. Navigating the intricacies of Cross-Origin Resource Sharing (CORS), I ensured that my frontend and backend spoke the same language, overcoming the hurdles of disparate domains. Vigilance against security vulnerabilities added a layer of suspense to the plot, with each input meticulously validated to thwart potential threats. Scaling, debugging, and monitoring became integral subplots, evolving as my application grew and flourished.

In this narrative of development and deployment, I embraced challenges as plot twists, each adding depth to the story of my web application's journey.

DESIGING THE WEBSITE:

FRONTEND



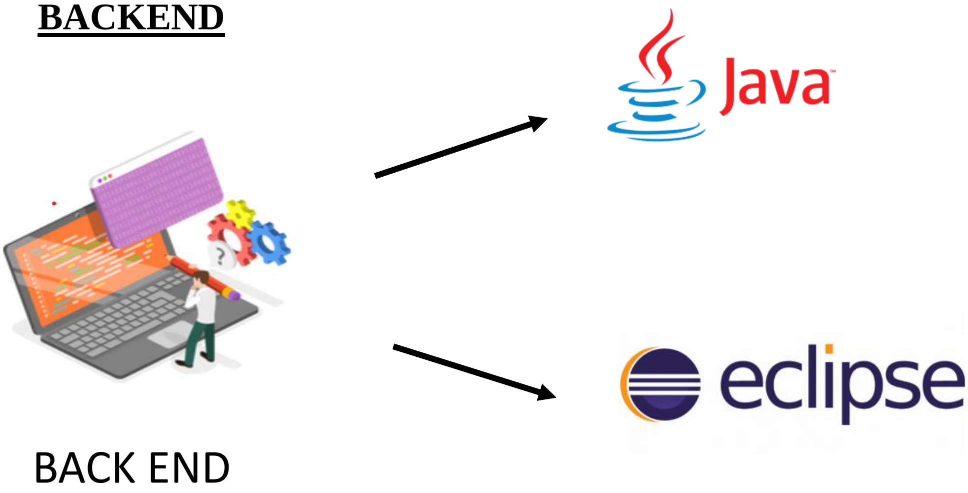
FRONTEND

HTML

CSS

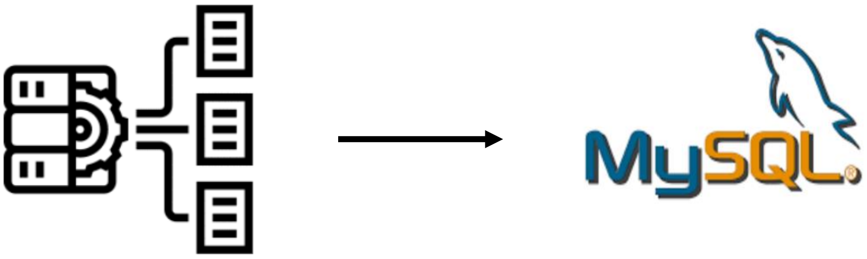
JAVA SCRIPT

BACKEND



BACK END

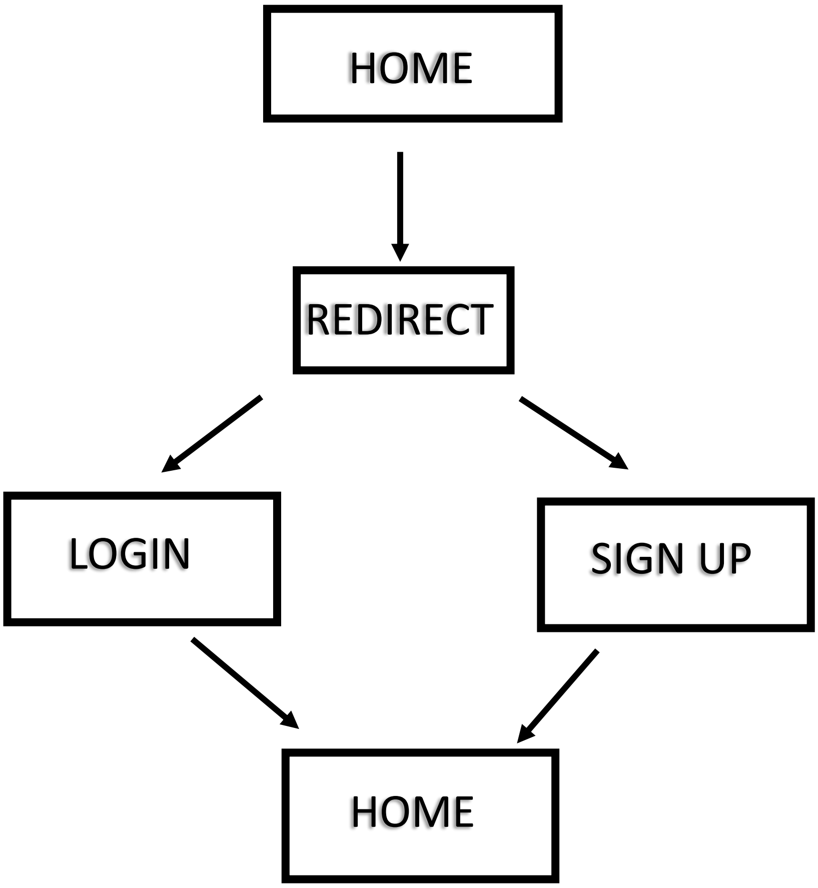
DATABASE



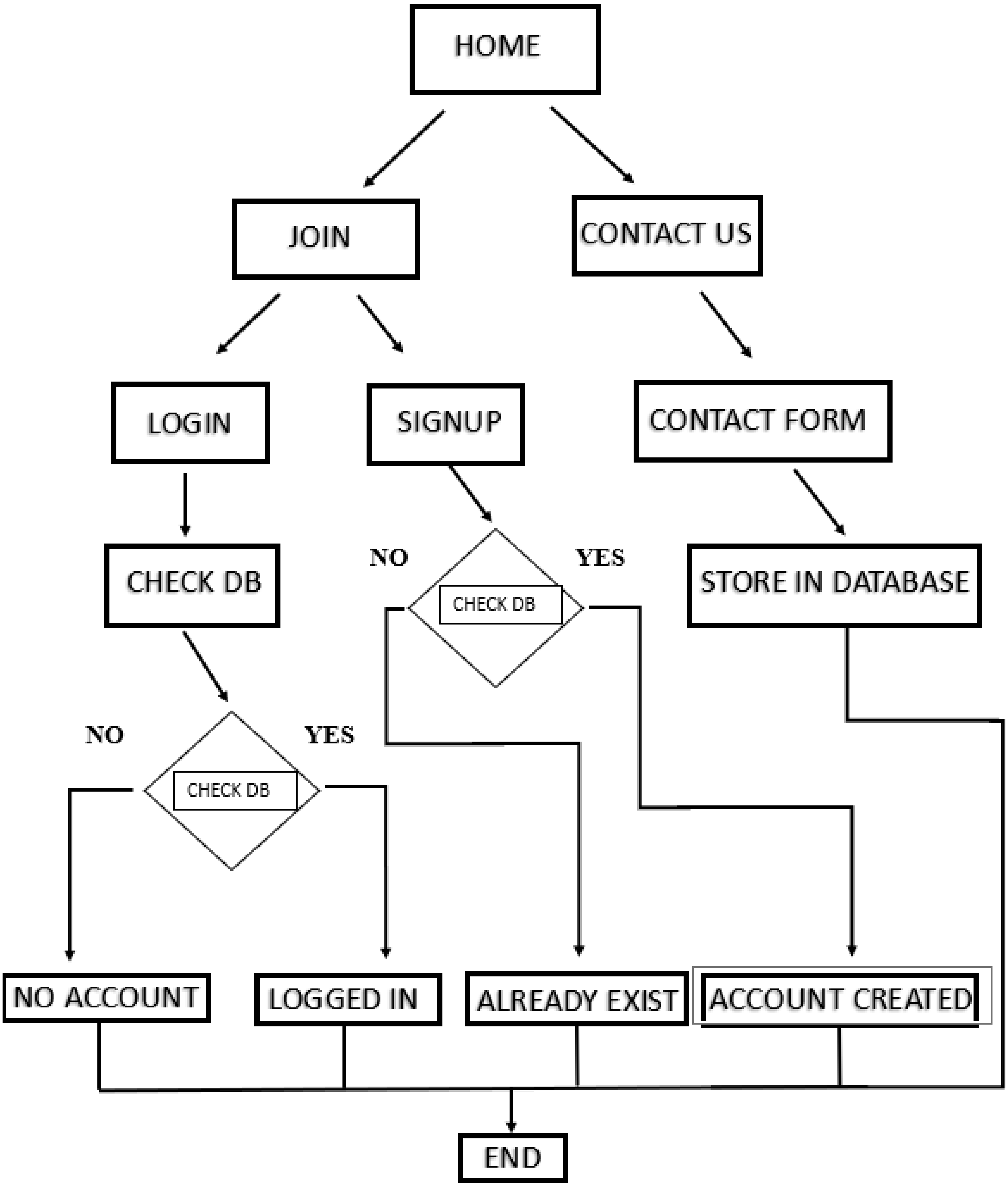
MODULES:

1. Sign-up
2. Log-in
3. WordPuzzl

FLOW DIAGRAM:



DIAGRAMMATIC REPRESENTATION:


HOME
JOIN
CONTACT US
LOGIN
SIGNUP
CONTACT FORM
CHECK DB
NO
CHECK DB
YES
STORE IN DATABASE
NO
CHECK DB
YES
NO ACCOUNT
LOGGED IN
ALREADY EXIST
ACCOUNT CREATED
END