

Institution Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Province** | Sindh | **City** | Karachi |
| **Institution** | National University of Computer and Emerging Sciences (FAST-NU) | **Campus** | Karachi |
| **Department** | Computer Science | **Degree Level** | BS |
| **Degree Program** | Software Engineer | **Telephone** |  |
| **Fax** |  | | |

Supervisor Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Dr Farrukh Shahid | **Gender** | Male |
| **Mobile** | +923331287466 | **Office No** |  |
| **Email** | Mfarrukh.shahid@nu.edu.pk | **Designation** | Lecturer |
| **Qualification** | PHD | | |

Co-Supervisor Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Shahbaz Siddiqui | **Gender** | Male |
| **Mobile** | +923002617916 | **Office No** |  |
| **Email** | Shahbaz.siddiqui@nu.edu.pk | **Designation** | Lecturer |
| **Qualification** | Masters | | |

Head of Department Details



|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Dr. Zulfiqar Memon | **Mobile No.** | - |
| **Email** | zulfiqar.memon@nu.edu.pk | **Gender** | Male |

Project Details



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Title** | Federated Learning using Smart Contracts in Healthcare sector | | | |  | |  |
| **Group Details** | **Member 1 Name: Sarmad Jamal**    **Member 1 Roll#: 19k-1116** | | | **Member 2 Name: Khizer Jilani**    **Member 2 Roll#: 19k-1057** | | **Member 3 Name: Mansoor Butt**    **Member 3 Roll#: 19k-1114** |  |
|  |  | |  | |  | |  |
| **Project Area of** | Blockchain/Federated Learning/Web Development | | | | | |  |
| **Specialization** |  | |  | |  | |  |
|  |  | |  | |  | |  |
| **Project Start** | (As per FYP Calendar) | | **Project End Date** | | (As per FYP Calendar) | |  |
| **Date** |  | |  | |  | |  |
|  |  | |  | |  | |  |
| **Project** |  |
| **Summary (less** |  | | | | | |  |
| **than 2500** |  | | | | | |  |
| **characters)** | We are designing a medical health care system which will give patients a comprehensive, immutable log and easy access to their medical information across providers and treatment sites. Leveraging unique block chain properties, Our System manages authentication, confidentiality, accountability and data sharing—crucial considerations when handling sensitive information. A modular design integrates with providers' existing, local data storage solutions, facilitating interoperability and making our system convenient and adaptable. The block content represents data ownership and viewership permissions shared by members of a private, peer-to-peer network. Blockchain technology supports the use of “smart contracts,” which allow us to automate and track certain state transitions (such as a change in viewership rights, or the birth of a new record in the system). Via smart contracts on an Ethereum blockchain, we log patient-provider relationships that associate a medical record with viewing permissions and data retrieval instructions (essentially data pointers) for execution on external databases. We include on the blockchain a cryptographic hash of the record to ensure against tampering, thus guaranteeing data integrity. Providers can add a new record associated with a particular patient, and patients can authorize sharing of records between providers. In both cases, the party receiving new information receives an automated notification and can verify the proposed record before accepting or rejecting the data. This keeps participants informed and engaged in the evolution of their records.  Our System prioritizes usability by also offering a designated contract which aggregates references to all of a user's patient-provider relationships, thus providing a single point of reference to check for any updates to medical history. We handle identity confirmation via public key cryptography and employ a DNS-like implementation that maps an already existing and widely accepted form of ID (e.g., name, or social security number) to the person's Ethereum address. A syncing algorithm handles data exchange “off-chain” between a patient database and a provider database, after referencing the blockchain to confirm permissions via our database authentication server.  . | | | | | |  |
|  |  | | | | | |  |
|  |  | |  | |  | |  |
| **Project** | Our system implementation addresses the three major issues:   1. slow access to medical data 2. system interoperability 3. patient agency   Recent advances in deep learning have shown many successful stories in smart healthcare applications with data-driven insight into improving clinical institutions’ quality of care. Excellent deep learning models are heavily data-driven. The more data trained, the more robust and more generalizable the performance of the deep learning model. However, pooling the medical data into centralized storage to train a robust deep learning model faces privacy, ownership, and strict regulation challenges. **Federated learning** resolves the previous challenges with a shared global deep learning model using a central aggregator server. At the same time, patient data remain with the local party, maintaining data anonymity and security.  This method provides decentralized machine learning model training with-out transmitting medical data through a coordinated central aggregate server. Medical institutions, working as client nodes, train their deep learning models locally and then periodically forward them to the aggregate server. The central server coordinates and aggregates the local models from each node to create a global model, then distributes the global model to all the other nodes. It is worth noting that the training data are kept private to each node and never transmitted during the training process. Only the model’s weight and parameters are transmitted, ensuring that medical data remain conﬁdential. For these reasons, FL mitigates many security concerns because it retains sensitive and private data while enabling multiple medical institutions to work together. FL holds an excellent promise in healthcare applications to improve medical services for both institutions and patients—for instance, predict autism spectrum disorder, mortality and intensive care unit (ICU) stay-time prediction. | | | | | | it |
|  |  | | | | | |  |
| **than 2500** |  | | | | | |  |
| **characters)** |  | | | | | |  |
|  |  | |  | |  | |  |
| **Literature Review / Background Study** | Blockchain is also used in the healthcare management system for effective maintenance of electronic health and medical records. The technology ensures security, privacy, and immutability. This work proposes a framework by integrating the blockchain and Federated Deep Learning in order to provide a tailored recommendation system.  Electronic medical records contain personal and confidential information that traditional storage methods must protect against cyberattacks and third-party authentication. To overcome this challenge, a method of distributed storage [5] was proposed in this work. The focus of this work is also on making treatment recommendations to patients by comparing their medical records with historical data. This work is motivated by the limitations of existing work. The current study supports his EHR preservation, but the recommendation system is neither discussed nor incorporated, making it difficult to create a treatment recommendation system. Federated learning may therefore lead to more accurate treatment recommendations. | | | | | |  |
| **Project Implementation Method (less than 2500 characters)** | Our approach is based on the Agile Scrum development methodology, which can be thought of as a series of short, reflexive sprints, this seemed to be the most suitable method of development because of its flexible and non-rigid characteristics as opposed to the traditional waterfall approach.  For the purpose of systematically and timely creating this complex and lengthy project we started making weekly goals for ourselves in which we would define a task for ourselves as a goal every week in order to make sure the successful completion of our project within the designated time. This project will be following MVC architecture, where our model will be our test network that we choose, it can be either ropsten, rinkeby. View is where our end user will be communicating with the data, it will be on React or can be on NEXT, since it optimizes the SEO and our controller will be ether.js which will help our contracts getting deployed on test networks and communicating with the React js on the frontend  Through Federated learning, multiple organizations or institutions work together to solve a machine-learning problem under the coordination of a central server or service provider. Thus, a deep-learning model is maintained and improved upon within a central server. The model is trained by distributing itself to hospitals which allows these sites to keep their data localized. Data from each collaborator is never exchanged or transferred during training. Instead of bringing the data to the central server, as in conventional deep learning, the central server maintains a global shared model, which is disseminated to all institutions. Each entity subsequently maintains a separate model based on its own patients’ data. Thereafter, each center provides feedback to the server based on its individually trained model—either by its weight or the error gradient of the model. The central server aggregates the feedback from all participants, and based on predefined criteria, updates the global model. The predefined criteria allow the model to evaluate the quality of the feedback and therefore to only incorporate that which is value-adding. The feedback from centers with adverse or strange results can thus be ignored. This process forms one round of federated learning, and it is iterated until the global model is trained. | | | | | |  |
|  |  | | | | | |  |
|  |  | | | | | |  |
|  |  | | | | | |  |
|  |  | | | | | |  |
| **Benefits of the** |  | | | | | |  |
| **Project (less** |  | | | |  | |  |
| **than 2500 characters)** |  | | | | | |  |
|  | The benefits of the project are listed below:   1. Easy to use GUI dark/light mode. 2. The application will be able to put a patient data on a Block chain Network. 3. The user data will be secure and can be retrieve easily by our web App. 4. For each user there will be a separate node containing user data making it more secure. 5. Our system supplements pointers with on-chain per missioning and data integrity logic, empowering individuals with record authenticity, auditability and data sharing. 6. Our System will be built robust, modular APIs to integrate with existing provider databases for interoperability 7. By Using **Federated Learning** Hospitals will be better equipped on predictive analysis of diseases. 8. Data Sharing will improve the redundant processes a patient has to go through. 9. Decentralization of Patients record will provide more autonomy and efficiency in the healthcare industry | | | | | |  |
|  |  |
|  |  | |  | |  | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technical** |  | | |  |
| **Details of Final Deliverable (less than 2500 characters)** | The project will be on a Web Application. Where we will use React/Next as a frontend framework and will be using MUI and tailwind CSS to make it a bit more professional and responsive, smart contracts will be written on solidity and deployed on ropsten or rinkeby (test networks). Mocha and Chai will help us testing our smart contracts and ether.js will help us integrating our contracts with React. For our federating learning part, there will be implementation of our Federated Learning model. Python will offer us a great feasibility with training and testing the model, each of the hospitals registered on the blockchain network act as a node and will have its own model trained on the concept of Federated Learning | | |  |
|  |  |  |  |  |
| **Final Deliverable of the Project** | The modules which will be used in this project are as follow:   * User Friendly GUI * A User Portal * Subscription Packages for different Users * Add a medical record to a system * Export PDF of any medical Report * Automation notification/alert to user for appointment * Previous medical Records * Secure and fast data retrieving | | |  |
|  |  |  |  |  |
| **Core Industry (Optional)** | As suggested by Supervisor |  |  |  |
|  |  |  |  |  |
| **Other** |  |  |  |  |
| **Industries**  **(Optional)** |  |  |  |  |
|  |  |  |  |  |
| **Core** | Blockchain, Federated Learning, Web3, React, Solidity, hardhat |  |  |  |
| **Technology** |  |  |  |  |
|  |  |  |  |  |
| **Other** |  |  |  |  |
| **Technologies (Optional)** |  |  |  |  |
|  |  |  |  |  |
| **Sustainable** |  |  |  |  |
| **Development** |  |  |  |  |
| **Goals**  **(Optional)** |  |  |  |  |
|  |  |  |  |  |
| References     |  |  |  | | --- | --- | --- | | 1. List and number all bibliographical references here like this. 2. A.B. Smith, C.D. Jones, and E.F. Roberts, “Article Title”, Journal, Publisher, Location, Date, pp. 1-10. 3. Jones, C.D., A.B. Smith, and E.F. Roberts, Book Title, Publisher, Location, Date |  |  | | |  |  |  |
| Project Key Milestones | |  |  |  |
|  | |  |  |  |
|  | | – |  |  |
| **Elapsed time in (days or weeks or month or quarter) since start of the project** | | **Milestone** | **Deliverable** |  |
|  |  |  |  |  |
| Month 1 |  | Project study and Research | Project Proposal, Presentation |  |
|  |  |  |  |  |
| Month 2 |  | Data Gathering | Initial Prototype |  |
|  |  |  |  |  |
| Month 3 |  | Frontend Development | Web Apps |  |
|  |  |  |  |  |
| Month 4 |  | Smart Contracts Creation | Solidity Codebase |  |
|  |  |  |  |  |
| Month 5 |  | Work on Federated Learning | ML Model |  |
|  |  |  |  |  |
| Month 6 |  | Integration of Smart Contracts and FL Model | FL Model with Smart Contract |  |
|  |  |  |  |  |
| Month 7 |  | Integration with Frontend | Fully Integrated Frontend and Backend |  |
|  |  |  |  |  |
| Month 8 |  | Testing and Deployment | Passed Test-Cases and Deployed WebApp |  |
|  |  |  |  |  |

Project Equipment Details



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item(s) Name** | **Type** | **No. of Units** | **Per Unit Cost (in Rs)** | **Total (in Rs)** |
|  |  |  |  |  |
| No need of equipment |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  | **Total in (Rs)** |  |
|  |  |  |  |  |

